### The Combinatorics of Generalised Cumulative Arrays[1]

Keith Martin, Siaw-Lynn Ng,
Information Security Group, Royal Holloway, University of London.

**Abstract**

In this paper we present a combinatorial analysis of generalised cumulative arrays. These are structures that are associated with a monotone collections of subsets of a base set and have properties that find application in areas of information security. We propose a number of basic measures of efficiency of a generalised cumulative array and then study fundamental bounds on their parameters. We then look at a number of construction techniques and show that the problem of finding good generalised cumulative arrays is closely related to the problem of finding boolean expressions with special properties.

# 1 Introduction

A *generalised cumulative array* (*GCA*) is a collection of functions that satisfy a specific property with regard to a monotone collection of subsets of a set. Generalised cumulative arrays are natural generalisations of structures known as cumulative arrays and, like cumulative arrays, have applications with respect to certain problems arising in the theory of information security. In this paper we present a combinatorial analysis of GCAs.

We begin in Section 2 with formal definitions and a look at simple examples. We will also discuss prior work and the motivation for the study of GCAs. In Section 3 we look at measures of efficiency of GCAs and discuss some fundamental bounds on the parameters of a GCA. In Section 4 we discuss the relationship between GCAs and boolean expressions and look at some construction techniques.

# 2 Generalised cumulative arrays

We begin this section with basic definitions and notation.

## 2.1 Definitions and notation

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a finite set and let $\Gamma$ be a collection of subsets of $\mathcal{P}$. We say that $\Gamma$ is *monotone* if for any pair of subsets $A, B$ of $\mathcal{P}$, if $A \in \Gamma$ and $A \subseteq B$ then $B \in \Gamma$. If $\Gamma$ is monotone then it can be described uniquely by the collection $\Gamma^-$ of *minimal sets* of $\Gamma$, that is, the subsets $A$ such that $A \in \Gamma$ but $A \backslash P_i \notin \Gamma$ for all $P_i \in A$. We say that $\Gamma$ is *connected* if every member of $\mathcal{P}$ belongs to some minimal set. In this paper we will refer to a pair $(\mathcal{P}, \Gamma)$ as a *defining structure* and will only consider defining structures that are monotone and connected. When convenient we will often simply denote a defining structure $(\mathcal{P}, \Gamma)$ by $\Gamma$.

A *generalised cumulative array* $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ *for* $(\mathcal{P}, \Gamma)$ is a set of $l > 0$ functions $f_1, \ldots, f_l$ and $l$ disjoint sets $\mathcal{K}_1, \ldots, \mathcal{K}_l$ with

$$f_i : \mathcal{P} \rightarrow 2^{\mathcal{K}_i}, \text{ where } \mathcal{K}_i = \{k_1^i, \ldots, k_{v_i}^i\}, \;\; v_i > 0,$$

such that, for all $A \subseteq \mathcal{P}$,

[1] This is the peer-reviewed version of the paper "The combinatorics of generalised cumulative arrays". K. Martin and S.-L. Ng. Journal of Mathematical Cryptology, Vol 1, 13-32, 2007, DOI: 10.1515/JMC.2007.002. The final publication is available at www.degruyter.com.

(D1) if $A \in \Gamma$ then for some $i_A \in \{1,\dots,l\}$,

$$\bigcup_{Q\in A} f_{i_A}(Q) = \mathcal{K}_{i_A};$$

(D2) if $A \notin \Gamma$ then for all $i \in \{1,\dots,l\}$,

$$\bigcup_{Q\in A} f_i(Q) \neq \mathcal{K}_i.$$

In other words, if $A$ is a member of the defining structure $\Gamma$ of $\mathcal{G}$ then there exists at least one function $f_{i_A}$ such that the image of $A$ under $f_{i_A}$ is $\mathcal{K}_{i_A}$. On the other hand, if $A$ is not a member of the defining structure $\Gamma$ of $\mathcal{G}$ then there is no function $f_i$ such that the image of $A$ under $f_i$ is $\mathcal{K}_i$.

For convenience of notation we will say that a generalised cumulative array as defined above is an example of a $GCA(\mathcal{P},\Gamma;l,\{v_1,\dots,v_l\})$. We will also, when appropriate, refer to it is an example of a $GCA(\mathcal{P},\Gamma)$ (and implicitly associate $GCA(\mathcal{P},\Gamma)$ with the set of all GCAs for $(\mathcal{P},\Gamma)$). If $v_1 = \cdots = v_l = v$ then we say that $\mathcal{G}$ is *$v$-uniform* and denote this more simply by $GCA(\mathcal{P},\Gamma;l,v)$.

One special class of defining structures are those for which $\Gamma$ is all the subsets of $\mathcal{P}$ of at least some fixed size $t$. We refer to such defining structures as *$(n,t)$-threshold structures* and use the simpler notation $GCA(n,t;l,\{v_1,\dots,v_l\})$ in this case.

Generalised cumulative arrays are referred to as *arrays* because they are often most conveniently represented as a set of arrays. More precisely, a $GCA(\mathcal{P},\Gamma;l,\{v_1,\dots,v_l\})$ can be represented as a collection $\{M_1,\dots,M_l\}$ of $n \times v_i$ arrays, where for each $1 \le h \le l$:

- $M_h$ has rows indexed by members of $\mathcal{P}$ and columns indexed by members of $\mathcal{K}_h$;
- Entry $(i,j)$ of $M_h$ is 1 if $k_j^h \in f_h(P_i)$, and 0 otherwise.

We illustrate the notation and representation in the following example.

**Example 2.1** Let $\mathcal{P} = \{a,b,c,d,e\}$ and the minimal sets of a defining structure $\Gamma$ be:

$$\begin{aligned}\Gamma^- \;=\; & \{\{a,b,c\},\{a,b,d\},\{a,b,e\},\{a,c,d\},\{a,c,e\},\\ & \{a,d,e\},\{b,c,d\},\{b,c,e\},\{b,d,e\},\{c,d,e\}\}\,.\end{aligned}$$

In other words, $\Gamma$ is a $(5,3)$-threshold structure. We give five examples of a $GCA(5,3)$ as follows:

(a) $\mathcal{G}_1$ is the following $GCA(5,3;10,3)$:

$$M_1 \quad \begin{array}{c|c|c|c|} & k_1^1 & k_2^1 & k_3^1 \\ \hline a & 1 & 0 & 0 \\ \hline b & 0 & 1 & 0 \\ \hline c & 0 & 0 & 1 \\ \hline d & 0 & 0 & 0 \\ \hline e & 0 & 0 & 0 \\ \hline \end{array} \qquad M_2 \quad \begin{array}{|c|c|c|} k_1^2 & k_2^2 & k_3^2 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad M_3 \quad \begin{array}{|c|c|c|} k_1^3 & k_2^3 & k_3^3 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \qquad M_4 \quad \begin{array}{|c|c|c|} k_1^4 & k_2^4 & k_3^4 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad M_5 \quad \begin{array}{|c|c|c|} k_1^5 & k_2^5 & k_3^5 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

$$M_6 \quad \begin{array}{c|c|c|c|} & k_1^6 & k_2^6 & k_3^6 \\ \hline a & 1 & 0 & 0 \\ \hline b & 0 & 0 & 0 \\ \hline c & 0 & 0 & 0 \\ \hline d & 0 & 1 & 0 \\ \hline e & 0 & 0 & 1 \\ \hline \end{array} \qquad M_7 \quad \begin{array}{|c|c|c|} k_1^7 & k_2^7 & k_3^7 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad M_8 \quad \begin{array}{|c|c|c|} k_1^8 & k_2^8 & k_3^8 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \qquad M_9 \quad \begin{array}{|c|c|c|} k_1^9 & k_2^9 & k_3^9 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \qquad M_{10} \quad \begin{array}{|c|c|c|} k_1^{10} & k_2^{10} & k_3^{10} \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

For clarity we will omit further labelling of the arrays.

(b) $\mathcal{G}_2$ is the following $GCA(5,3;1,10)$:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|}
\cline{2-11}
a & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \cline{2-11}
b & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \cline{2-11}
c & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \cline{2-11}
d & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ \cline{2-11}
e & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ \cline{2-11}
\end{array}$$

(c) $\mathcal{G}_3$ is the following $GCA(5,3;3,3)$:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\cline{2-4}\cline{6-8}\cline{10-12}
a & 1 & 0 & 0 & \quad & 1 & 0 & 0 & \quad & 1 & 0 & 0 \\ \cline{2-4}\cline{6-8}\cline{10-12}
b & 0 & 1 & 0 & & 1 & 0 & 0 & & 1 & 0 & 0 \\ \cline{2-4}\cline{6-8}\cline{10-12}
c & 0 & 0 & 1 & & 1 & 0 & 0 & & 0 & 1 & 0 \\ \cline{2-4}\cline{6-8}\cline{10-12}
d & 0 & 0 & 1 & & 0 & 1 & 0 & & 0 & 0 & 1 \\ \cline{2-4}\cline{6-8}\cline{10-12}
e & 0 & 0 & 1 & & 0 & 0 & 1 & & 0 & 0 & 1 \\ \cline{2-4}\cline{6-8}\cline{10-12}
\end{array}$$

(d) $\mathcal{G}_4$ is the following $GCA(5,3;2,\{3,6\})$:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|}
\cline{2-4}\cline{6-11}
a & 1 & 0 & 0 & \quad & 1 & 1 & 1 & 0 & 0 & 0 \\ \cline{2-4}\cline{6-11}
b & 1 & 0 & 0 & & 1 & 0 & 0 & 1 & 1 & 0 \\ \cline{2-4}\cline{6-11}
c & 0 & 1 & 0 & & 0 & 1 & 0 & 1 & 0 & 1 \\ \cline{2-4}\cline{6-11}
d & 0 & 1 & 0 & & 0 & 0 & 1 & 0 & 1 & 1 \\ \cline{2-4}\cline{6-11}
e & 0 & 0 & 1 & & 1 & 0 & 0 & 0 & 0 & 1 \\ \cline{2-4}\cline{6-11}
\end{array}$$

(e) $\mathcal{G}_5$ is the following $GCA(5,3;2,\{4,6\})$:

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|}
\cline{2-5}\cline{7-12}
a & 1 & 0 & 0 & 0 & \quad & 0 & 0 & 0 & 0 & 0 & 1 \\ \cline{2-5}\cline{7-12}
b & 0 & 0 & 1 & 1 & & 0 & 0 & 0 & 1 & 1 & 1 \\ \cline{2-5}\cline{7-12}
c & 0 & 1 & 0 & 1 & & 0 & 1 & 1 & 0 & 0 & 1 \\ \cline{2-5}\cline{7-12}
d & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & 0 & 1 & 0 \\ \cline{2-5}\cline{7-12}
e & 0 & 1 & 1 & 0 & & 1 & 1 & 0 & 1 & 0 & 0 \\ \cline{2-5}\cline{7-12}
\end{array}$$

□

Example 2.1 illustrates that for a particular defining structure $\Gamma$ there are many different types of GCA that can be found for $\Gamma$. In Section 3 we will propose a number of measures of efficiency that will help us to recognise when a GCA has properties that might be desirable.

## 2.2 Some constructions

We show first three simple constructions. We assume in each case that $\mathcal{P} = \{P_1, \ldots, P_n\}$.

**Construction 2.2** (a) Suppose $\Gamma$ is an $(n,1)$-threshold structure on $\mathcal{P}$. Let $\mathcal{K} = \{k\}$, and $f(P_i) = \{k\}$ for $i = 1, \ldots, n$. It is then straightforward to verify that $\mathcal{G}_{(n,1)} = (f;\mathcal{K})$ is a $GCA(n,1;1,1)$.

(b) Suppose $\Gamma$ is an $(n,n)$-threshold structure on $\mathcal{P}$. Let $\mathcal{K} = \{k_1, \ldots, k_n\}$ and $f(P_i) = \{k_i\}$ for $i = 1, \ldots, n$. It is then straightforward to verify that $\mathcal{G}_{(n,n)} = (f;\mathcal{K})$ is a $GCA(n,n;1,n)$.

(c) Suppose $\Gamma$ is an $(n,2)$-threshold structure on $\mathcal{P}$, and let $(b[i]_1 b[i]_2 \ldots b[i]_l)$ be the binary representation of $i$, $1 \leq i \leq n$, and $l = \lceil \log_2 n \rceil$.

Let $\mathcal{K}_j = \{k_0^j, k_1^j\}$, $1 \leq j \leq l$. Define $f_j : \mathcal{P} \rightarrow \mathcal{K}_j$, $1 \leq j \leq l$ as $f_j(P_i) = \{k^j_{b[i]_j}\}$.

Now, for any pair of participants $P_i$, $P_j$, since $i \neq j$, the binary representations of $i$ and $j$ must differ in some position, say the $h$th position. Then $f_h(P_i) \cup f_h(P_j) = \mathcal{K}_h$. For any single participant $P_i$, however, $f_j(P_i) \neq \mathcal{K}_j$ for all $1 \leq j \leq l$, by definition. Hence $\mathcal{G}_{(n,2)} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ is a $GCA(n, 2; \lceil \log_2 n \rceil, 2)$.

□

It is clear that both $\mathcal{G}_{(n,1)}$ and $\mathcal{G}_{(n,n)}$ are the smallest possible in terms of the number of arrays, the size of the arrays, and the number of $k^i_j$ assigned to each participant. We will see in Section 3 that $\mathcal{G}_{(n,2)}$ also meets the lower bounds for the number and size of arrays.

We now mention two known constructions for GCAs.

**Construction 2.3** [[12]]
Let $\Gamma^+ = \{B_1, \ldots, B_v\}$ be the set of all maximal subsets that are not members of a defining structure $(\mathcal{P}, \Gamma)$. Then $\mathcal{G} = (f, \mathcal{K})$ with $f : \mathcal{P} \rightarrow 2^{\mathcal{K}}$, $\mathcal{K} = \{k_1, \ldots, k_v\}$ and

$$f(P_i) = \{k_j \mid P_i \notin B_j,\ 1 \leq j \leq v\},$$

is a $GCA(\mathcal{P}, \Gamma; 1, v)$. □

A GCA constructed using the method in Construction 2.3 is more commonly referred to as a *cumulative array*. Cumulative arrays were first defined and studied in [12] and, as Construction 2.3 indicates, they exist for all defining structures. Example 2.1(b) is a GCA constructed using this method, and is thus a cumulative array for $(\mathcal{P}, \Gamma)$.

The second construction is based on perfect hash families. A *perfect hash family* $PHF(l; n, t, t)$, is a set of $l$ functions $H = \{h_1, \ldots, h_l\}$ with

$$h_i \ : \ \mathcal{P} = \{P_1, \ldots, P_n\} \ \rightarrow \ \{1, \ldots, t\}$$

such that for all $X \subseteq \mathcal{P}$, $|X| = t$, there is a $h_i \in H$ which is bijective when restricted to $X$.

**Construction 2.4** [[8]]
Let $H = \{h_1, \ldots, h_l\}$ be a $PHF(l; n, t, t)$. Let $\mathcal{K}_1, \ldots, \mathcal{K}_l$ be $l$ disjoint sets of size $t$, $\mathcal{K}_i = \{k_1^i, \ldots, k_t^i\}$, $i = 1, \ldots, l$. Let $f_i : \mathcal{P} \rightarrow 2^{\mathcal{K}_i}$, $i = 1, \ldots, l$, be defined as follows:

$$f_i(P_j) = \left\{k^i_{h_i(P_j)}\right\}.$$

Then, for any set $A \subseteq \mathcal{P}$, $|A| = t$, there exists an $i_A \in \{1, \ldots, l\}$ such that $h_{i_A}$ is bijective on $A$, that is,

$$\bigcup_{P \in A} h_{i_A}(P) = \{1, \ldots t\}.$$

Hence we have

$$\bigcup_{Q \in A} f_{i_A}(Q) = \bigcup_{Q \in A} \left\{k^{i_A}_{h_{i_A}(Q)}\right\} = \{k_1^{i_A}, \ldots, k_t^{i_A}\} = \mathcal{K}_{i_A}.$$

On the other hand, if $A < t$ then $\cup_{P \in A} h_i(P) \neq \{1, \ldots, t\}$ for all $i$, so we have $\cup_{P \in A} f_i(P) \neq \mathcal{K}_i$ for all $i$. Hence $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ is a $GCA(n, t; l, t)$. □

Note that Construction 2.2(c) is an example of Construction 2.4.

As we will see in Section 3, Construction 2.4 provides efficient GCAs for threshold structures. This construction does not, however, apply to general defining structures.

### 2.3 Motivation and prior work

Generalised cumulative arrays are a generalisation of cumulative arrays, a class of objects studied in relation to secret sharing schemes in [12]. A *secret sharing scheme* is a way of protecting a secret value amongst a set of *participants* by generating *shares* of the secret in such a way that only certain qualified groups (identified by the *access structure*) of participants can reconstruct the secret jointly from the shares that are allocated to them.

Cumulative arrays were defined in [12] as a combinatorial representation of a construction technique for secret sharing schemes for general access structures that was first proposed in [6]. As shares need to be kept secure in such a scheme, it is important to keep shares as small as possible. Translated into requirements for a cumulative array, this means that it is important to keep both $|f(P_i)|$ and $v$ as small as possible.

In [12] it was shown that the cumulative array generated by Construction 2.3 is essentially the unique *minimal* cumulative array, in the sense that no cumulative array exists for $\Gamma$ in which any of the values $|f(P_i)|$ or $v$ are smaller than those of Construction 2.3. Some other applications of cumulative arrays have subsequently been studied in [9, 15].

Generalised cumulative arrays arose from the problem of distributing computation for symmetric cipher systems. As symmetric encryption mechanisms explicitly avoid having underlying algebraic structure, in [3] a combinatorial technique of distributing the computation was proposed. In [8] it was pointed out that this approach was another application of cumulative arrays. It was also observed that significantly beneficial tradeoffs between the amount of information that participants needed to securely store and the amount of communication data (reducing the former at the expense of increasing the later) could be achieved through a generalisation of the techniques in [3]. The underlying combinatorial structure behind this generalisation was isolated in [8] and defined to be a generalised cumulative array.

We note that a GCA gives rise to a special type of secret sharing scheme. Let

$$\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$$

be a $GCA(\mathcal{P}, \Gamma; l, \{v_1, \ldots, v_l\})$, where $\mathcal{K}_i = \{k_1^i, \ldots, k_{v_i}^i\}$.

1. Let $s$ denote a secret that is chosen from $Z_m$ (for some suitably large $m$).
2. For each $i = 1, \ldots, l$ associate each value $k_j^i$, $j = 1, \ldots, v_i - 1$, with a random element $r(k_j^i) \in Z_m$. Let $r(k_{v_i}^i) = s - \sum_{j=1}^{v_i-1} r(k_j^i)$.
3. Let $P$ be issued with the share $\{r(k_j^i) \ : \ k_j^i \in f_i(P)\}$.

This gives rise to a secret sharing scheme with participant set $\mathcal{P}$ and access structure $\Gamma$, since any set $A \in \Gamma$ can compute $s = \sum_{j=1}^{v_i} r(k_j^i)$ for at least one array $i$, but any set $A \notin \Gamma$ is missing at least one value $r(k_j^i)$ amongst its collective shares.

## 3 Efficiency of GCAs

From Example 2.1 it is clear that many different GCAs can be found for a given defining structure. In this section we consider efficiency of GCAs. We will propose a number of appropriate efficiency measures and then establish some fundamental bounds on these measures.

### 3.1 Measures of efficiency

Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be a *GCA*$(\mathcal{P}, \Gamma)$. The discussion in Section 2.3 motivates the following three measures of efficiency of a GCA.

(a) The *storage* $\sigma(\mathcal{G})$ is defined as:

$$\sigma(\mathcal{G}) = \begin{matrix} \max \\ P \in \mathcal{P} \end{matrix} \left\{ \sum_{j=1}^{l} |f_j(P)| \right\}.$$

In terms of the application mentioned in Section 2.3, $\sigma(\mathcal{G})$ corresponds to the largest number of values that any participant would need to securely store. We further define the *optimal storage for* $(\mathcal{P}, \Gamma)$ to be:

$$\sigma^*(\mathcal{P}, \Gamma) = \begin{matrix} \min \\ \mathcal{G} \in GCA(\mathcal{P}, \Gamma) \end{matrix} \{\sigma(\mathcal{G})\}.$$

Thus $\sigma^*(\mathcal{P}, \Gamma)$ denotes the smallest storage of any *GCA*$(\mathcal{P}, \Gamma)$.

(b) The *weight* $\tau(\mathcal{G})$ is defined as:

$$\tau(\mathcal{G}) = \sum_{i=1}^{l} v_i.$$

In terms of the application mentioned in Section 2.3, $\tau(\mathcal{G})$ corresponds to the total number of values that appear in $\mathcal{G}$ (and is thus related to the amount of information it takes to represent any one of these values). We further define the *optimal weight for* $(\mathcal{P}, \Gamma)$ to be:

$$\tau^*(\mathcal{P}, \Gamma) = \begin{matrix} \min \\ \mathcal{G} \in GCA(\mathcal{P}, \Gamma) \end{matrix} \{\tau(\mathcal{G})\}.$$

Thus $\tau^*(\mathcal{P}, \Gamma)$ denotes the smallest weight of any *GCA*$(\mathcal{P}, \Gamma)$.

(c) The *dimension* $l(\mathcal{G})$ of $\mathcal{G}$ is defined to be $l$. In terms of the application mentioned in Section 2.3, $l(\mathcal{G})$ is a measure of communication bandwidth cost. We do not formally define a notion of optimal dimension, since the existence of a cumulative array for any defining structure indicates that such a measure would always be 1.

When the defining structure is an $(n, t)$-threshold structure then we will represent the optimal storage and optimal weight by $\sigma^*(n, t)$ and $\tau^*(n, t)$ respectively.

**Example 3.1** We apply the above efficiency measures to the various GCAs in Example 2.1:

(a) For $\mathcal{G}_1$: $\sigma(\mathcal{G}_1) = 6$, $\tau(\mathcal{G}_1) = 30$ and $l(\mathcal{G}_1) = 10$.

(b) For $\mathcal{G}_2$: $\sigma(\mathcal{G}_2) = 6$, $\tau(\mathcal{G}_2) = 10$ and $l(\mathcal{G}_2) = 1$.

(c) For $\mathcal{G}_3$: $\sigma(\mathcal{G}_3) = 3$, $\tau(\mathcal{G}_3) = 9$ and $l(\mathcal{G}_3) = 3$.

(d) For $\mathcal{G}_4$: $\sigma(\mathcal{G}_4) = 4$, $\tau(\mathcal{G}_4) = 9$ and $l(\mathcal{G}_4) = 2$.

(e) For $\mathcal{G}_5$: $\sigma(\mathcal{G}_5) = 5$, $\tau(\mathcal{G}_4) = 10$ and $l(\mathcal{G}_5) = 2$.

From these values we can see that $\sigma^*(5, 3) \leq 3$, and $\tau^*(5, 3) \leq 9$. □

**Example 3.2** Recall the known constructions from Section 2.2.

(a) The GCAs $\mathcal{G}$ defined by Construction 2.3 have $\tau(\mathcal{G}) = |\Gamma^+|$, $1 \leq \sigma(\mathcal{G}) \leq |\Gamma^+|$ and $l(\mathcal{G}) = 1$. In the case where $\Gamma$ is an $(n,t)$-threshold structure, Construction 2.3 yields GCAs with $\sigma(\mathcal{G}) = \binom{n-1}{t-1}$, $\tau(\mathcal{G}) = \binom{n}{t-1}$ and $l(\mathcal{G}) = 1$.

(b) The GCAs $\mathcal{G}$ defined by Construction 2.4, based on a $PHF(l;n,t,t)$, have $\sigma(\mathcal{G}) = l$, $\tau(\mathcal{G}) = tl$ and $l(\mathcal{G}) = l$. From [16] explicit constructions for $PHF(l;n,t,t)$s are provided with $l = C\log n$, where $C$ is a constant dependent on $t$ but independent of $n$. Hence we have $\sigma^*(n,t) \leq C\log n$ and $\tau^*(n,t) \leq Ct\log n$.

□

It should be evident from the examples looked at so far that it is not possible to find GCAs that have minimum storage, weight and dimension. While storage and weight are normally closely related, they generally need to be traded off against dimension. The cumulative arrays of Construction 2.3 have large storage and weight but minimum dimension. On the other hand Construction 2.4 provides a significant reduction in both storage and weight at the expense of an increase in dimension. The main challenge is thus to find GCAs that have "low" (as opposed to minimal) values for all of these parameters. This can be broken down into either:

1. seeking constructions with small dimension for some fixed storage and/or weight;
2. seeking constructions with small storage and/or weight for some fixed dimension.

We return to these challenges in Section 4.

## 3.2 Reducing and tightening a GCA

Since we wish to construct GCAs that have low weight and dimension, we observe that it is possible to ensure that a GCA does not contain any "redundancy". Here we introduce the notion of *irreducibility* and *tightness*:

Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be a $GCA(\mathcal{P}, \Gamma)$. We say that $\mathcal{G}$ is *irreducible* if we require all the $\mathcal{K}_i$; more precisely, if for each $1 \leq i \leq l$ there exists $A_i \in \Gamma$ such that $\mathcal{K}_i = \cup_{Q \in A_i} f_i(Q)$ and for any $j \neq i$, $\mathcal{K}_j \neq \cup_{Q \in A_i} f_i(Q)$. If $\mathcal{G}$ is not irreducible then there exists a set $\mathcal{K}_i$ that we can discard and still have a $GCA(\mathcal{P}, \Gamma)$.

We say that a GCA $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ is *tight* if for every $k^i_j \in \mathcal{K}_i$ there exists a $P \in \mathcal{P}$ and an $A \in \Gamma$ containing $P$ such that

1. $k^i_j \in f_i(P)$,
2. $\bigcup_{Q \in A} f_i(Q) = K_i$ but $\bigcup_{Q \in A} f_i(Q) \setminus \{k^i_j\} \neq K_i$, and
3. $\bigcup_{Q \in A} f_h(Q) \neq K_h$ for any $h \neq i$.

In other words, in a tight GCA it is not possible to reduce the weight by deleting some element $k^j_i$ without rendering $\mathcal{G}$ invalid. Note that tightness necessarily implies irreducibility.

The next two theorems indicate when certain types of redundancy can be removed from a GCA.

**Theorem 3.3** *Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be a $GCA(\mathcal{P}, \Gamma; l, \{v_1, \ldots, v_l\})$ such that either:*

*(a) Some array $\mathcal{K}_i$ with $v_i \geq 2$ has a column $k^i_r$ of 1s;*

*(b) Some array $\mathcal{K}_i$ has two identical columns $k_r^i$, $k_s^i$.*

*Then there exists $\mathcal{G}'$ a $GCA(\mathcal{P},\Gamma;l,\{v_1,\dots,v_i-1,\dots,v_l\})$ with $\sigma(\mathcal{G}')\leq\sigma(\mathcal{G})$ and $\tau(\mathcal{G}')=\tau(\mathcal{G})-1$.*

**Proof:** It is straightforward to verify that in either case $\mathcal{G}$ is not tight and that the GCA

$$\mathcal{G}'=(f_1,\dots,f_i',\dots,f_l;\mathcal{K}_1,\dots,\mathcal{K}_i',\dots,\mathcal{K}_l),$$

where $\mathcal{K}_i'=\mathcal{K}_i\setminus\{k_r^i\}$ and $f_i':\mathcal{P}\to\mathcal{K}_i'$ with $f_i'(P)=f_i(P)\setminus\{k_r^i\}$ is a $GCA(\mathcal{P},\Gamma;l,\{v_1,\dots,v_i-1,\dots,v_l\})$. □

**Theorem 3.4** *Let $\mathcal{G}=(f_1,\dots,f_l;\mathcal{K}_1,\dots,\mathcal{K}_l)$ be a $GCA(\mathcal{P},\Gamma;l,\{v_1,\dots,v_l\})$ such that either:*

*(a) Some array $\mathcal{K}_i$ has a column $k_r^i$ of 0s.*

*(b) Some array $\mathcal{K}_i$ such that for any $A\in\Gamma$ with $\cup_{Q\in A}f_i(Q)=\mathcal{K}_i$, we have $\cup_{Q\in A}f_i(Q)=\mathcal{K}_r$ for some $r\neq i$.*

*Then there exists $\mathcal{G}'$ a $GCA(\mathcal{P},\Gamma;l-1,\{v_1,\dots,v_{i-1},v_{i+1},\dots,v_l\})$ with $\sigma(\mathcal{G}')\leq\sigma(\mathcal{G})$ and $\tau(\mathcal{G}')=\tau(\mathcal{G})-v_i$.*

**Proof:** It is straightforward to verify that in either case $\mathcal{G}$ is not irreducible and that the GCA

$$\mathcal{G}'=(f_1,\dots,f_{i-1},f_{i+1},\dots,f_l;\mathcal{K}_1,\dots,\mathcal{K}_{i-1},\mathcal{K}_{i+1},\dots,\mathcal{K}_l)$$

is a $GCA(\mathcal{P},\Gamma;l-1,\{v_1,\dots,v_{i-1},v_{i+1},\dots,v_l\})$. □

We say that $\mathcal{G}'$ is a *tightening* of $\mathcal{G}$ if $\mathcal{G}'$ is obtained from $\mathcal{G}$ by applying Theorem 3.3 and we say that $\mathcal{G}'$ is a *reduction* of $\mathcal{G}$ if $\mathcal{G}'$ is obtained from $\mathcal{G}$ by applying Theorem 3.4.

## 3.3 Bounds on efficiency measures

In this section we consider some fundamental bounds on the efficiency measures identified in Section 3.1. We treat each of these in turn.

### 3.3.1 Bounds on weight

In this section we will show some lower bounds on the weight of a GCA. We begin with a useful set of observations.

**Lemma 3.5** *Let $\mathcal{G}=(f_1,\dots,f_l;\mathcal{K}_1,\dots,\mathcal{K}_l)$ be a $GCA(\mathcal{P},\Gamma)$ and let $A\in\Gamma^-$. If $\cup_{P\in A}f_i(P)=\mathcal{K}_i$ for some $1\leq i\leq l$ then:*

*(a) For each $Q\in A$, it follows that $f_i(Q)\cap\mathcal{K}_i\neq\emptyset$.*

*(b) For any distinct $Q,R\in A$, it follows that $f_i(Q)\not\subseteq f_i(R)$.*

*(c) $v_i\geq|A|$.*

**Proof:**

(a) If $f_i(Q)\cap\mathcal{K}_i=\emptyset$ then $\cup_{P\in(A\setminus\{Q\})}f_i(P)=\mathcal{K}_i$. Thus $(A\setminus\{Q\})\in\Gamma$, contradicting the minimality of $A$.

(b) If $f_i(Q)\subseteq f_i(R)$ then $\cup_{P\in(A\setminus\{Q\})}f_i(P)=\mathcal{K}_i$, contradicting the minimality of $A$ as in part (a).

(c) If $v_i < |A|$ then by (a) and (b) there must exist $B \subset A$ such that $\cup_{P \in B} f_i(P) = \mathcal{K}_i$, again contradicting the minimality of $A$.

□

The following lower bound on the weight of a GCA follows immediately from Lemma 3.5(c).

**Theorem 3.6** *Let $\mathcal{G}$ be a GCA$(\mathcal{P}, \Gamma)$ with dimension $l$ and let $a_\Gamma = \min\{|A| \; : \; A \in \Gamma^-\}$. Then $\tau(\mathcal{G}) \geq l a_\Gamma$.*

As an immediate corollary to Theorem 3.6 we see that Construction 2.4 generates GCAs for threshold structures of optimal weight.

**Corollary 3.7** *Let $\mathcal{G}$ be a GCA$(n, t)$ with dimension $l$. Then $\tau(\mathcal{G}) \geq lt$.*

The next result that we show is not strictly a bound on the weight of a GCA. Instead it establishes a bound on the product of the values $v_i$ (the weight is the sum).

**Theorem 3.8** *Let $\Gamma^+ = \{B_1, \ldots, B_v\}$ be the set of all maximal subsets that are not members of a defining structure $(\mathcal{P}, \Gamma)$. Then for any GCA$(\mathcal{P}, \Gamma; l, \{v_1, \ldots, v_l\})$ we have*

$$\prod_{i=1}^{l} v_i \geq |\Gamma^+|.$$

**Proof:** We count the size of following set in two different ways:

$$F = \left\{ (B, (k^1, \ldots, k^l) \mid B \in \Gamma^+, \; k^i \in \mathcal{K}_i, \; k^i \notin \bigcup_{P \in B} f_i(P) \text{ for all } i \right\}.$$

1. There are $|\Gamma^+|$ subsets $B \in \Gamma^+$. Since $B$ is not a member of $\Gamma$, there must be at least one $l$-tuple $(k^1, \ldots, k^l)$ such that $k^i \notin \bigcup_{P \in B} f_i(P)$, $1 \leq i \leq l$. Hence $|F| \geq |\Gamma^+|$.

2. Consider an $l$-tuple $(k^1, \ldots, k^l)$, $k^i \in \mathcal{K}_i$. Suppose there are distinct $B_1$, $B_2 \in \Gamma^+$ such that $k^i \notin \bigcup_{P \in B_1} f_i(P)$ and $k^i \notin \bigcup_{P \in B_2} f_i(P)$ for all $i$. Then $k^i \notin \bigcup_{P \in B_1 \cup B_2} f_i(P)$ for all $i$. Hence $B_1 \cup B_2$ contains $B_1$, $B_2$ and is not a member of $\Gamma$, contradicting the maximality of $B_1$, $B_2$. Hence for each $l$-tuple $(k^1, \ldots, k^l)$, there is at most one $B \in \Gamma^+$ such that $k^i \notin \bigcup_{P \in B} f_i(P)$ for all $i$, and there are $\prod_{i=1}^{l} v_i$ $l$-tuples. Hence we $\prod_{i=1}^{l} v_i \geq |F|$.

□

Note that the case $l = 1$ in Theorem 3.8 corresponds to a known result ([12]) about the weight of cumulative arrays:

**Corollary 3.9** *Let $\Gamma^+ = \{B_1, \ldots, B_v\}$ be the set of all maximal subsets that are not members of a defining structure $(\mathcal{P}, \Gamma)$. If $\mathcal{G}$ is a GCA$(\mathcal{P}, \Gamma; 1, v)$ then we have*

$$v = \tau(\mathcal{G}) \geq |\Gamma^+|.$$

### 3.3.2 Bounds on storage

Here we will prove some upper bounds on the storage of a GCA. We first establish a useful lemma.

**Lemma 3.10** *Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be a GCA$(\mathcal{P}, \Gamma)$ and let $A \in \Gamma^-$. If $\cup_{P \in A} f_i(P) = \mathcal{K}_i$ then $|f_i(P)| \leq v_i - |A| + 1$ for all $P \in A$.*

**Proof:** By Lemma 3.5(a) it follows that for every $Q \in A$ we have $f_i(Q) \cap \mathcal{K}_i \neq \emptyset$. Further by Lemma 3.5(b) it follows that for any pair $Q$, $R \in A$ we have $f_i(Q) \not\subseteq f_i(R)$. If $f_i(Q) > v_i - |A| + 1$ for some $Q \in A$ then there are at most $|A| - 2$ values of $\mathcal{K}_i$ left to be distributed amongst the members of $A \setminus \{Q\}$. Hence some $R \in A$ must be such that $f_i(R) \subseteq \cup_{P \in A \setminus \{R\}} f_i(P)$. Thus $(A \setminus \{R\}) \in \Gamma$, contradicting the minimality of $A$. □

The following upper bound on the storage of a GCA now follows.

**Theorem 3.11** *Let $\mathcal{G}$ be a GCA$(\mathcal{P}, \Gamma; l, \{v_1, \ldots, v_l\})$ and let $a_\Gamma = \min\{|A| \ : \ A \in \Gamma^-\}$. Then:*

$$\sigma(\mathcal{G}) \leq \sum_{i=1}^{l} v_i - l(a_\Gamma - 1).$$

**Proof:** By Lemma 3.10 it follows that $|f_i(P)| \leq v_i - a_\Gamma + 1$ for all $P \in \mathcal{P}$. Thus for any $P \in \mathcal{P}$, $\sum_{i=1}^{l} |f_i(P)| \leq \sum_{i=1}^{l} v_i - l a_\Gamma + l$. The result follows. □

The following immediate corollary to Theorem 3.11 sums up the case when $\Gamma$ is a threshold structure.

**Corollary 3.12** *Let $\mathcal{G}$ be a GCA$(n, t; l, \{v_1, \ldots, v_l\})$. Then:*

*(a)* $\sigma(\mathcal{G}) \leq \sum_{i=1}^{l} v_i - l(t-1)$*;*

*(b) if $\mathcal{G}$ is also $v$-uniform then $\sigma(\mathcal{G}) \leq l(v - t + 1)$.*

We can now use Corollary 3.12 to show that the existence of certain GCAs corresponds precisely to the existence of certain perfect hash functions.

**Theorem 3.13** *There exists a GCA$(n, t; l, t)$ if and only if there is a PHF$(l; n, t, t)$.*

**Proof:** The existence of a $GCA(n, t; l, t)$ follows the existence of a $PHF(l; n, t, t)$ by Construction 2.4. Suppose now that $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ is a $GCA(n, t; l, t)$, with $\mathcal{K}_i = \{k_1^i, \ldots, k_t^i\}$. From the proof of Theorem 3.11 we have that $|f_i(P_j)| \leq 1$. For each $i$, if $f_i(P_j) = \emptyset$ for any $P_j$ then let $f_i(P_j) = \{k_1^i\}$, say, without loss of generality. Now define $h_i$ as follows: if $f_i(P_j) = \{k_\alpha^i\}$ then $h_i(j) = \alpha$. It follows that $\{h_1, \ldots, h_l\}$ is a $PHF(l; n, t, t)$, since for any $A = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$, $\cup_{j=1}^{t} f_{i_A}(P_{i_j}) = \{k_1^{i_A}, \ldots, k_t^{i_A}\}$ for some $i_A$. Hence $\cup_{j=1}^{t} h_{i_A}(i_j) = \{1, \ldots, t\}$, so $h_{i_A}$ must be a bijection. □

### 3.3.3 Bounds on the dimension

We now establish some bounds relating to the dimension of a GCA. We first observe an upper bound on the dimension of an irreducible GCA.

**Theorem 3.14** *If $\mathcal{G}$ is an irreducible GCA$(\mathcal{P}, \Gamma)$ then $l(\mathcal{G}) \leq |\Gamma^-|$.*

**Proof:** Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be an irreducible $GCA(\mathcal{P}, \Gamma)$, where $\Gamma^- = \{A_1, \ldots, A_m\}$. Consider the $m \times l$ matrix $(a_{ij})$, with

$$a_{ij} = \begin{cases} 1 & \text{if } \cup_{Q \in A_i} f_j(Q) = \mathcal{K}_j, \\ 0 & \text{otherwise.} \end{cases}$$

As $\mathcal{G}$ is irreducible, for every column $j$ there must be at least one row $i$ with a 1 in column $j$ and 0 everywhere else. Hence there must be at least as many rows as columns, so $m = |\Gamma^-| \geq l$. □

The next bound on the dimension applies to $v$-uniform GCAs for threshold defining structures.

**Theorem 3.15** *If $\mathcal{G}$ is a $GCA(n, t; l, v)$ then $l \geq \left\lceil \log \binom{n}{t-1} / \log v \right\rceil$.*

**Proof:** By Theorem 3.8 it follows that

$$v^l \geq \binom{n}{t-1},$$

and hence $l \geq \log \binom{n}{t-1} / \log v$ . □

The bound of Theorem 3.15 can be met when $l = 1$ and when $t = 2$. When $l = 1$ this is the bound of cumulative arrays (see Example 3.2). When $t = 2$, Construction 2.2(c) provides a $GCA(n, 2; \lceil \log_2 n \rceil, 2)$ that meets the bound, with storage $\sigma = \lceil \log_2 n \rceil$ and weight $\tau = 2\lceil \log_2 n \rceil$. We now use the proof of Theorem 3.8 to construct a further example of a GCA meeting the bound of Theorem 3.15. This is a generalisation of Construction 2.2(c).

**Construction 3.16** Let $\Gamma$ be an $(n, 2)$-threshold structure on $\mathcal{P} = \{P_0, \ldots, P_{n-1}\}$. Suppose $v$ and $l$ are positive integers such that $l = \lceil \log_v n \rceil$. Let $\mathcal{K}_i = \{k^i_0, k^i_1, \ldots, k^i_{v-1}\}$, $i = 1, \ldots, l$. Let $(v[j]_1 v[j]_2 \ldots v[j]_l)$ be the base-$v$ representation of $j$. Let $f_i$, $i = 1, \ldots l$ be defined as

$$f_i(P_j) = \{k^i_h \mid h \neq v[j]_i\}.$$

Now, for any pair of participants $P_i$, $P_j$, their base-$v$ representation must differ in some position, say the $h$th position. Then $k^h_{v[i]_h} \in f_h(P_j)$ and $k^h_{v[j]_h} \in f_h(P_i)$, so $f_h(P_i) \cup f_h(P_j) = \mathcal{K}_h$. For any single participant $P_i$, $f_h(P_i) \neq \mathcal{K}_h$ for all $1 \leq h \leq l$, by definition. Hence $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ is a $GCA(n, 2; \lceil \log_v n \rceil, v)$ with storage $\sigma(\mathcal{G}) = (v-1)\lceil \log_v n \rceil$ and weight $\tau(\mathcal{G}) = v\lceil \log_v n \rceil$.

For example, a $GCA(9, 2; 2, 3)$ constructed using this method is as follows:

| | | | | | | |
|---|---|---|---|---|---|---|
| $P_1$ | 0 | 1 | 1 | 0 | 1 | 1 |
| $P_2$ | 0 | 1 | 1 | 1 | 0 | 1 |
| $P_3$ | 0 | 1 | 1 | 1 | 1 | 0 |
| $P_4$ | 1 | 0 | 1 | 0 | 1 | 1 |
| $P_5$ | 1 | 0 | 1 | 1 | 0 | 1 |
| $P_6$ | 1 | 0 | 1 | 1 | 1 | 0 |
| $P_7$ | 1 | 1 | 0 | 0 | 1 | 1 |
| $P_8$ | 1 | 1 | 0 | 1 | 0 | 1 |
| $P_9$ | 1 | 1 | 0 | 1 | 1 | 0 |

In general, this construction gives GCAs of higher weight and storage but lower dimension than those of Construction 2.2(c). □

# 4 Finding efficient GCAs

In this section we consider the problem of contructing efficient GCAs. We will begin by considering the close relationship between GCAs and boolean functions. Using this relationship we then look at some techniques for constructing GCAs recursively.

## 4.1 GCAs from boolean expressions

We now observe an important relationship between GCAs and boolean expressions. Recall that a boolean expression in $n$ variables $X_1, \ldots, X_{i_r}$ is said to be an *elementary disjunct* if it is a sum of boolean variables of the form $X_{i_1} + \cdots + X_{i_r}$ and is said to be a *elementary conjunct* if it is a product of boolean variables of the form $X_{j_1} \ldots X_{j_s}$. Further, a boolean expression is said to be in *disjunctive normal form (DNF)* if it is written as a disjunct (sum) of elementary conjuncts and is said to be in *conjunctive normal form (CNF)* if it is written as a conjunct (product) of elementary disjuncts. Any boolean expression has a unique disjunctive and conjunctive normal form.

We also recall that a convenient way of representing a defining structure $(\mathcal{P}, \Gamma)$ is to consider it as a (monotone) boolean function where the elements of $\mathcal{P}$ are boolean variables and the function $\Gamma$ is true for $A = \{P_{i_1}, \ldots, P_{i_r}\}$ if and only if $A \in \Gamma$. The following observation was made in [12]:

1. If $\Gamma$ is expressed in disjunctive normal form then the elementary conjuncts correspond to the minimal sets in $\Gamma$;

2. If $\Gamma$ is expressed in conjunctive normal form then the elementary disjuncts correspond to the maximal sets not in $\Gamma$ (and hence by Construction 2.3 the number of such elementary disjuncts is equal to the weight of the corresponding cumulative array).

We will now show that GCAs correspond precisely to boolean expressions written in a particular form. We say that a boolean expression is in *DCD form* if it is written as a disjunct of conjuncts of elementary disjuncts. For example, the boolean function $f(a, b, c, d) = (a + bc)d$ is not in DCD form. However, it can be written as $(a + b)(a + c)d$, which is in DCD form. Both disjunctive and conjunctive normal forms are also examples of DCD form.

**Theorem 4.1** *Let $\mathcal{G} = (f_1, \ldots, f_l; \mathcal{K}_1, \ldots, \mathcal{K}_l)$ be a $GCA(\mathcal{P}, \Gamma; l, \{v_1, \ldots, v_l\})$. Then $\mathcal{G}$ corresponds precisely to a boolean expression of the boolean function $\Gamma$ written in DCD form.*

**Proof:** We recall that $\mathcal{G}$ can be represented as a collection $\{M_1, \ldots, M_l\}$ of $n \times v_i$ arrays, where for each $1 \leq h \leq l$:

- $M_h$ has rows indexed by members of $\mathcal{P}$ and columns indexed by members of $\mathcal{K}_h$;

- Entry $(i, j)$ of $M_h$ is 1 if $k^h_j \in f_j(P_i)$, and 0 otherwise.

We associate each array $M_h$ with a conjunct of our boolean expression and each column $k_j$ of $M_h$ with an elementary disjunct. More precisely, we write $M_h = Z^1_h \ldots Z^{v_h}_h$, where $Z^j_h$ is the elementary disjunct consisting of all boolean variables $P_i$, where entry $(i, j)$ of $M_h$ is 1. We then represent $\mathcal{G}$ by the sum of the conjuncts $M_h$. It is easy to see that the resulting boolean expression is in DCD form. Further, this expression is true if and only if one of the conjuncts $M_h$ is true. By definition, this happens if and only if a set of variables $A \in \Gamma$ and hence this boolean expression represents the boolean function $\Gamma$.

It should be clear that the reverse process can be applied to any boolean expression for the boolean function $\Gamma$ in DCD form to produce a set of arrays that in turn corresponds to a $GCA(\mathcal{P}, \Gamma)$. □

We illustrate Theorem 4.1 by returning to Example 2.1 and providing the equivalent DCD boolean expressions.

**Example 4.2** Let $(\mathcal{P},\Gamma)$ be the $(5,3)$-threshold structure defined on $\mathcal{P}=\{a,b,c,d,e\}$. The five examples of a $GCA(5,3)$ provided in Example 2.1 correspond precisely to the following boolean expressions for $\Gamma$ in DCD form:

(a) $\mathcal{G}_1 = abc+abd+abe+acd+ace+ade+bcd+bce+bde+cde$, which is the disjunctive normal form for $\Gamma$;

(b) $\mathcal{G}_2 = (a+b+c)(a+b+d)(a+b+e)(a+c+d)(a+c+e)(a+d+e)(b+c+d)(b+c+e)(b+d+e)(c+d+e)$, which is the conjunctive normal form for $\Gamma$;

(c) $\mathcal{G}_3 = ab(c+d+e)+(a+b+c)de+(a+b)c(d+e)$;

(d) $\mathcal{G}_4 = (a+b)(c+d)e+(a+b+e)(a+c)(a+d)(b+c)(b+d)(c+d+e)$;

(e) $\mathcal{G}_5 = a(b+c)(b+d+e)(c+d+e)+(a+b+c)(b+d)(b+e)(c+d)(c+e)(d+e)$.

□

Note that in the correspondence shown in Theorem 4.1 the storage of a GCA is equivalent to the maximum occurrence of a variable in the boolean expression, the weight corresponds to the total number of elementary disjuncts in the boolean expression and the dimension corresponds to the number of conjuncts of elementary disjuncts in the expression. Hence we may attempt to construct a GCA with particular properties by trying to find a boolean expression for the boolean function corresponding to the defining structure that has a particular DCD form.

In general, efficient GCAs will correspond to DCD boolean expressions in which

1. each variable does not appear too frequently;
2. the number of elementary disjuncts is small;
3. the number of conjuncts of elementary disjuncts is small.

Consider for a moment an $n\times v$ array $M$ with rows indexed by $\{1,\ldots,n\}$ and columns by $\{1,\ldots,v\}$. Consider how one might assign 0 or 1 to the entries $m_{ij}$ of $M$, with exactly one 1 in every row, so that the number of subsets $A\subseteq\{1,\ldots,n\}$ satisfying

$$\bigcup_{i\in A}\{j\mid m_{ij}=1\}=\{1,\ldots,v\}$$

is maximal. This is the problem of determining the maximum possible number of sets of size $v$ that can be separated by a fixed partition of $n$ points into $v$ parts. This problem was discussed in [5], and it is clear that the maximal is achieved when the numbers of 1 in each column are as uniform as possible.

Interpreted in terms of GCAs, this indicates that the function $f_i$ should assign each $k_j^i$ to roughly the same number of participants, thereby making sure that as many sets as possible belonging to the defining structure has image $\mathcal{K}_i$ under $f_i$. This in turn will help to reduce the dimension and weight of the GCA. Reinterpreted in terms of the boolean expression for a GCA in DCD form, this indicates that to achieve efficiency, each elementary disjunct in a conjunct should have roughly the same number of variables. Hence the DCD form should be a disjunct of conjuncts of "linear factors" consisting of roughly the same number of variables.

On the other hand, once a set belonging to the defining structure has image $\mathcal{K}_i$ under $f_i$, it does not need to have any other image $\mathcal{K}_j$ under $f_j$, $j \neq i$. The elimination of such duplications will go towards reducing the storage as well as the dimension of the GCA. In terms of boolean expressions, this indicates that when the expression is expanded into disjunctive normal form, each disjunct should not be repeated too many times.

We illustrate this with an example:

**Example 4.3** Consider the defining structure

$$\Gamma = P_1P_2 + P_1P_3P_4 + P_2P_3P_4 + P_1P_3P_5P_6 + P_2P_3P_5P_6 + P_1P_4P_5P_6 + P_2P_4P_5P_6 + P_3P_4P_5P_6.$$

This is in disjunctive normal form (hence in DCD form) and corresponds to a GCA with $l(\Gamma) = 8$, $\tau(\Gamma) = 28$, and $\sigma(\Gamma) = 5$. We may express $\Gamma$ in a DCD form that is closer to the description above, that is, as a disjunct of conjuncts of "linear factors" with roughly the same number of variables, as follows:

$$\Gamma_1 = P_1P_2 + (P_1 + P_2)P_3(P_4 + P_5)(P_4 + P_6) + (P_1 + P_2 + P_3)P_4P_5P_6.$$

This gives a more efficient GCA with $l(\Gamma_1) = 3$, $\tau(\Gamma_1) = 10$ and $\sigma(\Gamma_1) = 3$. □

The criteria for a "good" boolean expression of an efficient GCA for general defining structures cannot be formulated exactly, since it is not always clear, as discussed in Section 3.1, what the "best" GCA is. For example, it is not clear which of $\mathcal{G}_3$, $\mathcal{G}_4$ from Example 4.2 is the "better" GCA. It would appear then that any method of producing efficient DCD forms would be heuristic. The difficulty of producing efficient circuits for boolean functions in general ([17]) would indicate that this too is a difficult problem.

In the next section we exploit the relationship between GCAs and boolean expressions in the recursive construction of GCAs.

## 4.2 Some recursive constructions

It is clear that one may construct GCAs for general defining structures by simply splitting the defining structure into subsets and constructing GCAs (by using cumulative arrays or other means) for each subset. Since a defining structure can be written in disjunctive normal form a trivial example of such a construction would be to construct a GCA for each elementary conjunct. (The GCA $\mathcal{G}_1$ in Example 2.1 is such a construction.) This will give a GCA with as many arrays as there are minimal sets. Splitting the defining structure in different ways will give different GCAs with differing efficiency. For instance, splitting the defining structure of Example 2.1 into $\Gamma_1 = abc + acd + ade + bce + cde$ and $\Gamma_2 = abd + abe + ace + bcd + bde$ and constructing cumulative arrays for each of $\Gamma_1$, $\Gamma_2$ using Construction 2.3 would yield a $GCA(5, 3; 2, \{5, 6\})$ with $\sigma = 5$, $\tau = 11$ and $l = 2$, compared with $\mathcal{G}_1$ which has $\sigma = 6$, $\tau = 30$ and $l = 10$.

This observation motivates the following constructions of new GCAs from old GCAs. We first recall some terminology from [4, 7] that we can apply to defining structures of a GCA.

Let $(\mathcal{P}_1, \Gamma_1)$ and $(\mathcal{P}_2, \Gamma_2)$ be two defining structures and let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$.

1. The *sum* of $(\mathcal{P}_1, \Gamma_1)$ and $(\mathcal{P}_2, \Gamma_2)$ is the defining structure $(\mathcal{P}, \Gamma_1 + \Gamma_2)$ such that for all $A \subseteq \mathcal{P}$,

$$A \in \Gamma_1 + \Gamma_2 \iff A \cap \mathcal{P}_1 \in \Gamma_1 \text{ or } A \cap \mathcal{P}_2 \in \Gamma_2.$$

2. The *product* of $(\mathcal{P}_1, \Gamma_1)$ and $(\mathcal{P}_2, \Gamma_2)$ is the defining structure $(\mathcal{P}, \Gamma_1\Gamma_2)$ such that for all $A \subseteq \mathcal{P}$,

$$A \in \Gamma_1\Gamma_2 \iff A \cap \mathcal{P}_1 \in \Gamma_1 \text{ and } A \cap \mathcal{P}_2 \in \Gamma_2.$$

Using the correspondence between GCAs and boolean expressions, it is clear that a GCA for the sum of defining structures $\Gamma_1$, $\Gamma_2$ can be derived by a disjunct of $\Gamma_1$, $\Gamma_2$ expressed as boolean functions. Similarly a GCA for the product of $\Gamma_1$, $\Gamma_2$ can be constructed by a conjunct of $\Gamma_1$, $\Gamma_2$ expressed as boolean functions. Hence the following theorem:

**Theorem 4.4** *Let $\mathcal{G}_1$ be a $GCA(\mathcal{P}_1, \Gamma_1)$ and let $\mathcal{G}_2$ be a $GCA(\mathcal{P}_2, \Gamma_2)$. Then there exists:*

*(a) $\mathcal{G}_{\Gamma_1+\Gamma_2}$, a $GCA(\mathcal{P}, \Gamma_1 + \Gamma_2)$ with $l(\mathcal{G}_{\Gamma_1+\Gamma_2}) \leq l(\mathcal{G}_1) + l(\mathcal{G}_2)$, $\tau(\mathcal{G}_{\Gamma_1+\Gamma_2}) \leq \tau(\mathcal{G}_1) + \tau(\mathcal{G}_2)$ and $\sigma(\mathcal{G}_{\Gamma_1+\Gamma_2}) \leq \sigma(\mathcal{G}_1) + \sigma(\mathcal{G}_2)$ (if $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$ then $\sigma(\mathcal{G}_{\Gamma_1+\Gamma_2}) = \max(\sigma(\mathcal{G}_1), \sigma(\mathcal{G}_2))$).*

*(b) $\mathcal{G}_{\Gamma_1\Gamma_2}$, a $GCA(\mathcal{P}, \Gamma_1\Gamma_2)$ with $l(\mathcal{G}_{\Gamma_1\Gamma_2}) \leq l(\mathcal{G}_1)l(\mathcal{G}_2)$, $\tau(\mathcal{G}_{\Gamma_1\Gamma_2}) \leq l_2\tau(\mathcal{G}_1)+l_1\tau(\mathcal{G}_2)$ and $\sigma(\mathcal{G}_{\Gamma_1\Gamma_2}) \leq l_2\sigma(\mathcal{G}_1) + l_1\sigma(\mathcal{G}_2)$ (if $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$ then $\sigma(\mathcal{G}_{\Gamma_1\Gamma_2}) = \max(l_2\sigma(\mathcal{G}_1), l_1\sigma(\mathcal{G}_2))$).*

We will refer to the GCA $\mathcal{G}_{\Gamma_1+\Gamma_2}$ constructed as in Theorem 4.4 as the *sum* of $\mathcal{G}_1$, $\mathcal{G}_2$ and write $\mathcal{G}_{\Gamma_1+\Gamma_2} = \mathcal{G}_1 + \mathcal{G}_2$. Similarly we will refer to $\mathcal{G}_{\Gamma_1\Gamma_2}$ as the *product* of $\mathcal{G}_1$, $\mathcal{G}_2$ and write $\mathcal{G}_{\Gamma_1\Gamma_2} = \mathcal{G}_1\mathcal{G}_2$.

Now we consider some applications of Theorem 4.4.

**Construction 4.5** As an example of how these constructions may be combined, we consider a recursive construction of GCAs for threshold defining structures using the binomial identity

$$\binom{n}{t} = \binom{n-k}{t} + \sum_{i=1}^{k} \binom{k}{i}\binom{n-k}{t-i}.$$

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$, and let $\Gamma_{(n,t)}$ denote an $(n,t)$-threshold structure defined on $\mathcal{P}$. We may construct a $GCA(n,t)$ from existing $GCA(n-k,t)$s, $GCA(n-k,t-i)$s and $GCA(k,i)$s for any $1 \leq k < t$, $1 \leq i \leq k$.

Let $k$ be any integer such that $1 \leq k < t$. Let $\mathcal{G}_{(n-k,t)}$ be a $GCA(\{P_1, \ldots, P_{n-k}\}, \Gamma_{(n-k,t)})$. For each $1 \leq i \leq k$, let $\mathcal{G}_{(n-k,t-i)}$ be a $GCA(\{P_1, \ldots, P_{n-k}\}, \Gamma_{(n-k,t-i)})$, and let $\mathcal{G}_{(k,i)}$ be a $GCA(\{P_{n-k+1}, \ldots, P_n\}, \Gamma_{(k,i)})$. Then we may construct a $GCA(n,t)$, $\mathcal{G}$, using Theorem 4.4 by observing that:

$$\mathcal{G} = \mathcal{G}_{(n-k,t)} + \sum_{i=1}^{k} \mathcal{G}_{(n-k,t-i)}\mathcal{G}_{k,i}.$$

□

Since there are constructions using perfect hash families which will give $GCA(n,t)$s with both $l$ and $\sigma$ in the order of $\log n$, such a construction will produce GCAs with $l$ and $\sigma$ about

$$C_1 \log(n-k) + C_2 k \log k \log(n-k)$$

for some $C_1$, $C_2$ dependent on $t$. In general, these will not be as efficient as those constructed directly from a perfect hash family. However, in some cases, it may allow the choice of some components that may be more suitable for an application which are not constructed from a perfect hash family (for example, using Construction 3.16 which has lower dimension than Construction 2.2(c)), and it may also allow the reuse of some existing components.

We now show that recursive constructions can be used to build GCAs for defining structures that are not threshold structures. As an example we consider the quasi-threshold multipartite defining structures that were first described in [11].

**Construction 4.6** Let $\mathcal{P}_1, \ldots, \mathcal{P}_N$ be disjoint participant sets with $|\mathcal{P}_i| = n_i$, $1 \le i \le N$. Let $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_N$ and let $n = |\mathcal{P}|$. Let $t_1, \ldots, t_N$ be integers such that $1 \le t_i < n_i$, and let $1 \le T \le N-1$. Then $\Gamma$ is a *quasi-threshold* $((n_1, t_1), \ldots, (n_N, t_N); T)$*-multipartite defining structure* if $\Gamma$ consists of sets $A \subseteq \mathcal{P}$ such that

either $A \subseteq \mathcal{P}_1$ with $|A| \ge t_1$,
or $|A \cap \mathcal{P}_1| \ge t_1 - 1$ and $|A \cap \mathcal{P}_j| \ge t_j$ for all $j \in S$
where $S$ is a $T$-subset of $\{2, \ldots, N\}$.

In other words, sets in the defining structure contains either $t_1$ or more members of $\mathcal{P}_1$, or at least $t_1 - 1$ members of $\mathcal{P}_1$ together with a threshold number $(t_j)$ of members from a minimum number $(T)$ of the remaining $N-1$ classes $\mathcal{P}_2, \ldots, \mathcal{P}_N$.

We can construct a $GCA(\mathcal{P}, \Gamma)$ from existing "smaller" GCAs as follows. Writing in terms of sums and product of defining structures, we have, in fact, that

$$\Gamma = \Gamma_{(n_1,t_1)} + \Gamma_{(n_1,t_1-1)} \left( \sum_{\substack{S \subseteq \{2,\ldots,N\} \\ |S|=T}} \prod_{j \in S} \Gamma_{(n_j,t_j)} \right).$$

Hence a $GCA(\mathcal{P}, \Gamma)$, $\mathcal{G}$, may be constructed correspondingly: Let $\mathcal{G}_i$ be a $GCA(\mathcal{P}_i, \Gamma_{(n_i,t_i)}\ )$, $1 \le i \le N$, and let $\mathcal{G}_1'$ be a $GCA(\mathcal{P}_1, \Gamma_{(n_1,t_1-1)}\ )$. Then

$$\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_1' \left( \sum_{\substack{S \subseteq \{2,\ldots,N\} \\ |S|=T}} \prod_{j \in S} \mathcal{G}_j \right).$$

Since there are $GCA(n, t)$ with dimension and weight in the order of $\log n$, this construction gives GCAs of quasi-threshold multipartite defining structure with dimension and weight in the order of $\log \left( n_1 + \sum_{\substack{S \subseteq \{2,\ldots,N\} \\ |S|=T}} n_i \right)$. □

**Example 4.7** Let $\Gamma$ be the quasi-threshold $((3, 2), (5, 2); 1)$- multipartite threshold defining structure on $\mathcal{P} = \{a, b, c\} \cup \{d, e, f, g, h\}$, so

$$\begin{aligned} \Gamma &= \Gamma_{(3,2)} + \Gamma_{(3,1)}\Gamma_{(5,2)} \\ &= (ab + ac + bc) + (a + b + c)(de + df + dg + dh + ef + eg + eh + fg + fh + gh) \\ &= (a + c)(a + b)(b + c) + (a + b + c)(d + e + f + g)h \\ &\quad +(a + b + c)(f + g)(d + e + h) + (a + b + c)(d + f + h)(e + g). \end{aligned}$$

This gives a GCA with $l = 4$, $\sigma = 5$, $\tau = 12$. A cumulative array would have given $l = 1$, $\sigma = 11$ and $\tau = 16$. □

# 5 Conclusions

We have presented a combinatorial analysis of general cumulative arrays and proposed measures of efficiency corresponding to various properties of GCAs that might be of consideration in applications. We have also provided bounds on these measures. We then showed an important relationship between GCAs and boolean expressions. While there is a construction using perfect hash families which gives efficient GCAs for threshold defining structures, there is no known efficient construction for general defining structures. Using this correspondence with boolean expressions, however, we are able to provide a description of desirable properties of boolean expressions in DCD form corresponding to efficient GCAs. This correspondence also provides a convenient tool for constructing GCAs recursively.

# References

[1] M. Atici, S. S. Magliveras, D. R. Stinson and W. D. Wei. *Some recursive constructions for perfect hash families.* Journal of Combinatorial Designs, 4(353–363), 1996.

[2] S. R. Blackburn, M. Burmester, Y. Desmedt and P. R. Wild. *Efficient multiplicative sharing schemes.* Advances in Cryptology - Eurocrypt '96, LNCS, 1070(107–118), 1996.

[3] E. Brickell, G. Di Crescenzo and Y. Frankel. Sharing Block Ciphers. ACISP'00, 457-470.

[4] E. Brickell and D.R. Stinson. *Some improved bounds on the information rate of perfect secret sharing schemes.* Journal of Cryptology, pp 153–166, Vol 5, 1992.

[5] M. L. Fredman and J. Komlós. *On the size of separating systems and families of perfect hash functions.* SIAM Journal of Algebraic and Discrete Mathematics, pp 61–68, Vol 5, No 1, 1984.

[6] M. Ito, A. Saito and T. Nishizeki. Secret sharing scheme realsing general access structures. Journal of Cryptology, pp 15–20, Vol 6, 1993.

[7] K. Martin. New secret sharing schemes from old. Journal of Comb. Mathematics and Comb. Computing, pp 65–77, vol 14 1993.

[8] K. Martin, R. Safavi-Naini, H Wang and P. R. Wild. Sharing the encryption and decryption of a block cipher. To appear in Designs, Codes and Cryptography.

[9] K. Martin, J. Peiprzyk, R. Safavi-Naini, H Wang and P. R. Wild. Threshold MACs. ICISC02, 5th international conference on information security and cryptography. Lecture Notes in Computer Science, Vol 2587, pp 237–252, 2003.

[10] K. Mehlhorn. *On the program size of perfect and universal hash functions.* Proceedings of the 23rd Annual IEEE Symposium on foundations of Computer Science, 1982.

[11] S. -L. Ng. Ideal secret sharing schemes with multipartite access structures To appear in *IEE Proc. Communications.*

[12] G. J. Simmons, W. A. Jackson and K. M. Martin. The Geometry of shared secret schemes. *Bull. Inst. Combin. Appl. 1: 71–88, 1991.*

[13] D. R. Stinson, R. Wei and L. Zhu. *New constructions for perfect hash families and related structures using combinatorial designs and codes.* To appear in Journal of Combinatorial Designs.

[14] K. Tochikubo, T. Uyematsu and R. Matsumoto. *Efficient secret sharing schemes based on authorised subsets.* IEICE Trans. Fundamentals, Vol E88-A, No. 1. January 2005.

[15] H. Wang and J. Peiprzyk. Shared generation of pseudo-random functions with cumulative maps. CT-RSA'03, Lecture Notes in Computer Science, Vol 2612, pp 281–294, 2003.

[16] H. Wang and C. Xing. *Explicit constructions of perfect hash families from algebraic curves over finite fields.* Journal of Combinatorial Theory A, 93(112–124), 2001.

[17] I. Wegener. The complexity of boolean functions. Wiley-Teubner, Stuttgart, 1987.