# On separating codes


Gérard D. Cohen
ENST Dept. INFRES
46, rue Barrault
F-75634 Paris Cedex 13
France
cohen@enst.fr

Sylvia B. Encheva
HSH
Bjørnsonsgt. 45
N-5528 Haugesund
Norway
sbe@hsh.no

Hans Georg Schaathun
UiB Dept. of informatics
Høyteknologisenteret
N-5020 Bergen
Norway
georg@ii.uib.no



**Abstract**

Let $\Gamma$ be a code of length $n$, and $(T, U)$ a pair of disjoint subsets of $\Gamma$. We say that $(T, U)$ is separated if there exists a coordinate $i$, such that for any codeword $(c_1, \ldots, c_n) \in T$ and any codeword $(c'_1, \ldots, c'_n) \in U$, $c_i \neq c'_i$. The code $\Gamma$ is $(t, u)$-separating if all pairs $(T, U)$ with $\#T = t$ and $\#U = u$ are separated.

Separating codes (or systems) are known from combinatorics, and they have also been applied, under various terminology, for water-marking.

We present some new bounds, generalisations, and constructions for separating codes.



**Keywords**

separating system, intersecting code


## Codes séparants


**Résumé**

Soit $\Gamma$ un code de longeur $n$, et $(T, U)$ un couple de sous-ensembles disjoints de $\Gamma$. On dit que $(T, U)$ est séparé s'il existe une position $i$ telle que pour tout mot $(c_1, \ldots, c_n) \in T$ et tout mot $(c'_1, \ldots, c'_n) \in U$, $c_i \neq c'_i$. Le code $\Gamma$ est dit $(t, u)$-séparant si tout tel couple où $\#T = t$ et $\#U = u$ est séparé.

Les codes, ou systèmes, séparants sont connus en Combinatoire ; ils ont été utilisés, sous des vocables divers, dans des problèmes de marquage numérique.

Nous présentons de nouvelles bornes, des généralisations et des constructions de codes séparants.



**Mots-clefs**

système séparant, code intersectant

Bergen, October 23rd 2001
This report was originally published as internal report no. 2001D003 at Ecole Nationale Supérieure des Télécommunications, Paris, France.

# Skiljande kodar

**Samandrag**

Lat $\Gamma$ vera ein kode med leng $n$, og lat $(T, U)$ vera eit par av disjunkte delmengder av $\Gamma$. Me seier at $(T, U)$ er skilt om det er ein plass $i$ slik at for alle ord $(c_1, \ldots, c_n) \in T$ og alle ord $(c'_1, \ldots, c'_n) \in U$, har me $c_i \neq c'_i$. Koden $\Gamma$ er skiljande om alle slike par med $\#T = t$ og $\#U = u$ er skilde.

Skiljande kodar, eller system, er kjende frå kombinatorikken, og dei har vorte nytta, under ymse namn, til vannmerking og kopivern.

Me skal finna nokre nye grenser, generaliseringar og konstruksjoner for skiljande kodar.

**Stikkord**
skiljande system, snittande kode

# Chapter 1

# Introduction

The theory of separating systems has been applied in different areas of science and technology such as automata synthesis, technical diagnosis, constructions of hash functions, and authenticating ownership claims.

The following property and generalisations have been studied e.g. by Körner and Simonyi under the set-theoretic terminology of $(i, j)$-*separation* [KS88]. Associate in a natural way to every row $r$ of an $n \times M$ array $T$ a bi-partition of the set of coordinates $E = \{1, 2, \ldots, M\}$, i.e. a pair $\{A_r, B_r\}$ of disjoint subsets of $E$ such that $A_r \cup B_r = E$. Then for any ordered $t$-tuple $(j_1, j_2, \ldots, j_t)$ of $E$, there is a bipartition which separates $\{j_1, j_2, \ldots, j_w\}$ from $\{j_{w+1}, \ldots, j_t\}$. The problem of finding the minimum size of such a separating family of partitions for arbitrary $|E|$ remains open. The case of $(2, 2)$-separation is introduced by Sagalovich in the context of automata: two such systems transiting simultaneously from state $a$ to $a'$ and from $b$ to $b'$ respectively should be forbidden to pass through a common intermediate state. He has written a long series of papers since the sixties, e.g. [Sag65, Sag75]; a fairly recent survey can be found in [Sag94]. States are simply binary $n$-tuples and only shortest paths are allowed during transitions; in other words, the only 'moves' permitted while transiting from $a$ to $a'$ are complementing the $d(a, a')$ bits where $a$ and $a'$ differ (one at a time). Clearly if the separation property holds, no two such minimal-length paths between $a$ and $a'$, and $b$ and $b'$ will intersect.

The design of self-checking asynchronous networks has been a challenging problem. Friedmann et al. [FGU69] have shown that the unicode single-transition-time asynchronous state assignment correspond to $(2, 2)$- and $(2, 1)$-separating systems.

Digital watermark is a perceptually invisible pattern embedded in a digital image. The watermark can carry information about the owner of the image or the recipient: watermarking for copyright protection, fingerprinting [BS98], or traitor tracing [SW98]. Codes were introduced in [BS98] (see also [DSW00]) as

a method of 'digital fingerprinting' which prevents a coalition of a given size from forging a copy with no member of the coalition being caught, or from framing an innocent user.

The rest of this chapter will be spent on defining basic terminology and notation, whereas in the next section we will discover some bounds on the length and distance of separating codes, as well as some constructions.

In Chapter 3, we will introduce generalised separation and hashing, and give some more general results. Chapter 4 will be dealing with intersecting codes and demonstrate the relation between separation and intersection.

We will try to summarise the known results in form of tables of rates in Chapter 5. Finally we will give some results on maximum weights in Chapter 6.

## 1.1 Separating Codes

Let $G$ be an additive group and $\mathbb{V}$ the set of $n$-tuples over $G$. An $(n, M)$ code $\Gamma$ is an $M$-subset $\Gamma \subseteq \mathbb{V}$. If $G$ is a field of $q$ elements and $C$ is a $k$-dimensional subspace $C \leqq \mathbb{V}$, then we say that $C$ is a $[n, k]_q$ (linear) code.

The feasible set $F(T)$ of some vector set $T \subseteq \mathbb{V}$ is

$$F(T) := \{(v_1, \ldots, v_n) \in \mathbb{V} \mid \forall i, 1 \leq i \leq n, \exists (a_1, \ldots, a_n) \in T, a_i = v_i\}.$$

In the case of fingerprinting, each user holds one codeword, and a coalition of users make a set $T$ of codewords; then $F(T)$ is the set of (false) fingerprints they can produce.

**Definition 1**
We say that a code $C$ is $(t, u)$-separating if, for any pair $(T, U)$ of disjoint subsets of $C$ where $\#T = t$ and $\#U = u$, the feasible sets are disjoint, i.e. $F(T) \cap F(U) = \emptyset$.

In earlier works on watermarking, $(t, t)$-separating codes have been called $t$-PIC (partially identifying codes) [CE00b] or $t$-SFP (secure frameproof) [SW98, SvTW00, SSW00]. The current terminology appears to be older though [Sag94]. Different special cases have also appeared in litterature; the $t$-frameproof codes from [SSW00] are just $(t, 1)$-separating codes. The $(2, 1)$-separating, binary codes has also been studied in [Kör95].

**Definition 2**
A $(t, u)$-configuration is a pair $(T, U)$ of disjoint vector sets of sizes $t$ and $u$ respectively. We say that $(T, U)$ is separated if $F(T) \cap F(U) = \emptyset$, and otherwise it is non-separated. A $(t, u)$-NSC is a non-separated $(t, u)$-configuration.

A code is $(t,u)$-separating if and only if it contains no $(t,u)$-NSC. Obviously, if $C$ is $(t,u)$-separating, then it is also $(u,t)$-separating, and $(t',u')$-separating for all $t' \leq t$ and all $u' \leq u$.

**Remark 1.1**
If $\pi : \mathbb{V} \mapsto \mathbb{V}$ is an automorphism, then $(T,U)$ is a $(t,u)$-NSC if and only if $(\pi(T),\pi(U))$ is a $(t,u)$-NSC.

**Remark 1.2**
If $(T,U)$ is a $(t,u)$-NSC, then so is $(T+\mathbf{c},U+\mathbf{c})$ for any $\mathbf{c} \in \mathbb{V}$. If $T+\mathbf{c},U+\mathbf{c} \subset C$, then $T,U \subset C'$ for some code $C'$ equivalent to $C$. If $C$ is linear and $T,U \subset C$, then $T+\mathbf{c},U+\mathbf{c} \subset C$.

Remark 1.1 tells us that if $C$ is a linear $(t,u)$-separating code, then so is any equivalent code. Also, if $\Gamma$ is a non-linear $(t,u)$-separating code, then so is any equivalent code, by Remark 1.2.

## 1.2 Minimum and Maximum Weights

For any vector $\mathbf{c} = (c_1,\ldots,c_n) \in \mathbb{V}$ we define the support to be

$$\chi(\mathbf{c}) := \{i \mid c_i \neq 0\}.$$

For any subset $S \subseteq \mathbb{V}$, the support is

$$\chi(S) := \bigcup_{\mathbf{c}\in S} \chi(\mathbf{c}).$$

We define the weight of subsets and codewords to be the size of their support, and denote it $w(\mathbf{c}) := \#\chi(\mathbf{c})$ or $w(S) := \#\chi(S)$.

Let $C$ be a linear code. The $r$-th minimum support weight $d_r$ of $C$ is the least weight of an $r$-dimensional subcode of $C$. The $r$-th maximum support weight $m_r$ is the largest weight of an $r$-dimensional subcode of $C$. Both these numbers were first studied in [HKM77], and the minimum support weight has received quite some attention following [Wei91], where it was called the $r$-th generalised Hamming weight.

It is clear that $d_1$ is the minimum distance of the code, and likewise $m_1$ is the maximum distance of the code; so these two numbers are defined also for non-linear codes. Several general definitions of $d_r$ exist for non-linear codes, but we will not need any of them here.

# Chapter 2

# Separation

The first section of this chapter is devoted to necessary conditions for $(t, t')$-separation. We will start with lower bounds on $q$ in terms of $t$ and $t'$. The second half of the section will present bounds on the minimum and maximum distances.

In Section 2.2, we give sufficient conditions for a code to be $(2, 2)$-separating based on minimum and maximum weights. Finally, in Section 2.3, we present some constructions.

## 2.1 Bounds for linear codes

**Proposition 1**
Let $\mathbf{a}$ and $\mathbf{b}$ be two linearly independent codewords, and write $T = \{\mathbf{a}, \mathbf{b} + \alpha\mathbf{a} \mid \alpha \in \mathsf{GF}(q)\}$. Then $(\mathbf{0}, T)$ is a $(q+1, 1)$-NSC.

**Proof:** We shall prove that in every position $i$, at least one codeword in $T$ has a 0. If $b_i = 0$, this holds, so assume $b_i \neq 0$. Then $\mathbf{b} + (-a_i^{-1})b_i\mathbf{a}$ has 0 in position $i$, as required. □

**Corollary 1**
If $C$ is $q$-ary, linear $(t, t')$-separating, then $\max\{t, t'\} \leq q$.

This bound is tight in the binary case, since $(2, 2)$-separating, binary, linear codes are known to exist (e.g. [Sag94]).

**Theorem 1**
If $C$ is a non-binary, linear $(t, t')$-separating, then $t + t' \leq q + 1$.

**Proof:** We have already proved that $t, t' \leq q$. It only remains to prove that we can construct a $(t, q + 2 - t)$-NSC for all $t$ such that $2 \leq t \leq q$. Let $\alpha_0, \alpha_1, \ldots, \alpha_{q-1}$ be

all the fields elements, where $\alpha_0 = 0$ and $\alpha_1 = 1$. Let $\mathbf{a}$ and $\mathbf{b}$ be two independent codewords. A $(t, q+2-t)$-NSC is given by

$$(\{\alpha_0\mathbf{a}, \ldots, \alpha_{t-1}\mathbf{a}\}, \{\alpha_t\mathbf{a}, \mathbf{a}+\alpha_1\mathbf{b}, \ldots, \mathbf{a}+\alpha_{q+1-t}\mathbf{b}\}).$$

First note that $\alpha_t\mathbf{a}$ matches $\mathbf{0}$ on every position not in $\chi(\mathbf{a})$, and $\mathbf{a}+\mathbf{b}$ match $\mathbf{a}$ on every position not in $\chi(\mathbf{b})$. In every position in $\chi(\mathbf{a}) \cap \chi(\mathbf{b})$, we get $t$ different field values in the first set, and $q+1-t$ different field values from the $\mathbf{a}+\alpha_i\mathbf{b}$. Since there are only $q$ elements in the field, they cannot be separated. □

**Proposition 2**
If $C$ is a linear, $(q, 1)$-separating code, then $m_1 < n-k+2$.

**Proof:** We shall prove that if $n-m_1 \leq k-1$, then $C$ cannot be $(q, 1)$-separating. Consider a codeword $\mathbf{c}$ of maximum weight and let $T = \{\alpha\mathbf{c} \mid \alpha \in \mathsf{GF}(q)\}$. Since the code is linear, for every set of $k-2$ coordinate positions, there exist at least $q-1$ non-zero codewords which are zero on these positions. In particular, there is a non-zero codeword $\mathbf{a}$ which is zero on every position not in $\chi(\mathbf{c})$. Thus $(T; \mathbf{a})$ is a $(q, 1)$-NSC. □

**Proposition 3**
If $C$ is a linear, binary $(2, 2)$-separating code, then $m_1 < n-2(k-2)$.

**Proof:** If $k \leq 1$, the result is trivial. For $k = 2$, it only says that the all-one codeword $\mathbf{1}$ cannot be in the code $C$, lest $(\mathbf{0}, \mathbf{1}; \mathbf{c}, \mathbf{c}+\mathbf{1})$ form a $(2, 2)$-NSC for $\mathbf{c} \in C \setminus \{\mathbf{0}, \mathbf{1}\}$.

We then turn to the case $k \geq 3$. We shall prove that if $n-m_1 \leq 2(k-2)$, then $C$ cannot be $(2, 2)$-separating. Consider a codeword $\mathbf{c}$ of maximum weight. Since the code is linear, for every set of $k-2$ coordinate positions, there exist at least three non-zero codewords which are zero on these positions, and thus at least one which is not $\mathbf{c}$. In particular, there is a non-zero codeword $\mathbf{a}$ which is zero on half the positions not in $\chi(\mathbf{c})$, and one $\mathbf{b}$ which is zero on the other half. Thus $(\mathbf{0}, \mathbf{c}; \mathbf{a}, \mathbf{b})$ is a $(2, 2)$-NSC. □

**Proposition 4**
If $C$ is non-binary, linear, $(t, 2)$-separating, then $d_1 > (t-1)k$.

**Proof:** Assume for a contradiction that $d_1 \leq (t-1)k$. We shall construct a $(t, 2)$-NSC. Let $\mathbf{c}$ be a codeword of minimum weight. By Remark 1.1, we can assume that $\mathbf{c}$ is one on every non-zero coordinate. Since the code is linear, for every set of $k-1$ coordinates there exist at least $(q-1)$ non-zero codewords which are zero on these coordinates. For every set of $k$ coordinates there exists at least one non-zero codeword which are either one or zero on these coordinates. Hence there exist $t-1$ codewords $\mathbf{a}_1, \ldots, \mathbf{a}_{t-1}$ such that at least one of them is either one or zero

on every position in $\chi(\mathbf{c})$. Hence $(\mathbf{0}, \mathbf{c}; \alpha\mathbf{c}, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1})$ is a $(2,t)$-NSC for every $\alpha \neq 0, 1$. □

## 2.2 $(2,2)$-separating codes

Let $\mathbf{c}', \mathbf{c}, \mathbf{a}, \mathbf{b}$ be distinct vectors such that $(\{\mathbf{c}', \mathbf{c}\}, \{\mathbf{a}, \mathbf{b}\})$ is a $(2,2)$-NSC. From this assumption, we will derive some statements on the minimum and maximum weights of any code which is not $(2,2)$-separating. This will give a sufficient condition for a code to be $(2,2)$-separating in Theorem 2.

By Remark 1.2, we can assume that $\mathbf{c}' = \mathbf{0}$; and by Remark 1.1, we can assume that $\mathbf{c} = (1, \ldots, 1, 0, \ldots, 0)$. We write

$$\begin{aligned} \mathbf{c} &= (c_1, c_2, \ldots, c_n), \\ \mathbf{a} &= (a_1, a_2, \ldots, a_n), \\ \mathbf{b} &= (b_1, b_2, \ldots, b_n). \end{aligned}$$

Let $r$ be such that $c_i = 1$ for $i \leq r$ and $c_i = 0$ for $i > r$. Since $(\mathbf{0}, \mathbf{c}; \mathbf{a}, \mathbf{b})$ is a $(2,2)$-NSC, there is no coordinate $i$ such that both $a_i \notin \{0, c_i\}$ and $b_i \notin \{0, c_i\}$.

We consider the sum

$$\begin{aligned} \Sigma &:= d(\mathbf{0}, \mathbf{a}) + d(\mathbf{0}, \mathbf{b}) + d(\mathbf{c}, \mathbf{a}) + d(\mathbf{c}, \mathbf{b}) \\ &= w(\mathbf{a}) + w(\mathbf{b}) + w(\mathbf{a} - \mathbf{c}) + w(\mathbf{b} - \mathbf{c}). \end{aligned}$$

We have trivially that

$$4d_1 \leq \Sigma \leq 4m_1. \tag{2.1}$$

Consider now the matrix with rows $\mathbf{0}, \mathbf{c}, \mathbf{a}, \mathbf{b}$. Let $\mathbf{x}_i$ be the $i$-th column in this matrix. We have four main types of columns:

$$\begin{aligned} &\text{Type 0}: && \mathbf{x}_i = (0,0,0,0), && \\ &\text{Type I}: && \mathbf{x}_i \in \{(0,0,0,\alpha), (0,0,\alpha,0)\}, && \alpha \neq 0, \\ &\text{Type IIa}: && \mathbf{x}_i \in \{(0,1,0,0), (0,1,1,1)\}, && \\ &\text{Type IIb}: && \mathbf{x}_i \in \{(0,1,0,1), (0,1,1,0)\}, && \\ &\text{Type III}: && \mathbf{x}_i \in \{(0,1,0,\beta), (0,1,\beta,0), (0,1,1,\beta), (0,1,\beta,1)\}, && \beta \notin \{0,1\}. \end{aligned}$$

No other possibility exists because the rows form a $(2,2)$-NSC. We have now that

$$\Sigma = \sum_{i=1}^{n} \sigma(\mathbf{x}_i), \tag{2.2}$$

where $\sigma(\mathbf{x}_i)$ is 0 for Type 0, 2 for Types I and II, and 3 for Type III. Let $v_{\mathrm{X}}$ denote the number of columns of Type X. Then we get

$$n = v_0 + v_{\mathrm{I}} + v_{\mathrm{II}} + v_{\mathrm{III}}, \tag{2.3}$$
$$\Sigma = 2v_{\mathrm{I}} + 2v_{\mathrm{II}} + 3v_{\mathrm{III}}. \tag{2.4}$$

**Proposition 5**
If $(\mathbf{0}, \mathbf{c}; \mathbf{a}, \mathbf{b})$ is a $(2,2)$-NSC, then

$$\Sigma = w(\mathbf{c}) + w(\mathbf{a}-\mathbf{b}) + w(\mathbf{a}+\mathbf{b}-\mathbf{c}).$$

**Proof:** We have trivially that

$$n - w(\mathbf{c}) = v_0 + v_{\mathrm{I}}. \tag{2.5}$$

Define two vectors

$$\mathbf{y} = (y_1, y_2, \ldots, y_n) := \mathbf{a}+\mathbf{b}-\mathbf{c},$$
$$\mathbf{z} = (z_1, z_2, \ldots, z_n) := \mathbf{a}-\mathbf{b}.$$

We have that

$$\begin{aligned}
&\mathbf{x}_i \text{ of Type 0} && \Rightarrow y_i = 0 \quad \wedge \quad z_i = 0,\\
&\mathbf{x}_i \text{ of Type I} && \Rightarrow y_i = 1 \quad \wedge \quad z_i = \pm 1,\\
&\mathbf{x}_i \text{ of Type IIa} && \Rightarrow y_i = \pm 1 \quad \wedge \quad z_i = 0,\\
&\mathbf{x}_i \text{ of Type IIb} && \Rightarrow y_i = 0 \quad \wedge \quad z_i = \pm 1,\\
&\mathbf{x}_i \text{ of Type III} && \Rightarrow y_i \in \{\beta, \beta-1\} = \{\alpha \neq 0\}\\
& && \quad \wedge z_i \in \{\pm(\beta-1), \pm\beta\} = \{\alpha \neq 0\}.
\end{aligned}$$

This gives

$$n - w(\mathbf{a}+\mathbf{b}-\mathbf{c}) = n - w(\mathbf{y}) = v_0 + v_{\mathrm{IIb}},$$
$$n - w(\mathbf{a}-\mathbf{b}) = n - w(\mathbf{z}) = v_0 + v_{\mathrm{IIa}}.$$

By adding together the two equations above as well as (2.5), we get

$$3n - (w(\mathbf{c}) + w(\mathbf{a}-\mathbf{b}) + w(\mathbf{a}+\mathbf{b}-\mathbf{c})) = 3v_0 + v_{\mathrm{IIa}} + v_{\mathrm{IIb}} + v_{\mathrm{I}}.$$

From (2.4) and (2.3) we get that

$$\Sigma = 3n - (3v_0 + v_{\mathrm{IIa}} + v_{\mathrm{IIb}} + v_{\mathrm{I}}) = w(\mathbf{c}) + w(\mathbf{a}-\mathbf{b}) + w(\mathbf{a}+\mathbf{b}-\mathbf{c}),$$

as required. □

We observe that $d(\mathbf{a}, \mathbf{b}) = w(\mathbf{a}-\mathbf{b})$ and $d(\mathbf{0}, \mathbf{c}) = w(\mathbf{c})$ are distances in the code; hence they are bounded by $m_1$. If $C$ is linear, $w(\mathbf{a}+\mathbf{b}-\mathbf{c})$ is also a distance in the code, and thus bounded by $m_1$. If $C$ is non-linear, we still have $w(\mathbf{a}+\mathbf{b}-\mathbf{c}) \leq n$. This gives directly the following theorem.

**Theorem 2**
If a code satisfies $4d_1 > 2m_1 + n$, or if $4d_1 > 3m_1$ and it is linear, then it is $(2,2)$-separating.

**Corollary 2**
All linear, equidistant codes are $(2,2)$-separating. A non-linear, equidistant code is $(2,2)$-separating if $2d_1 > n$.

The binary linear case of Theorem 2 has previously been proved by Sagalovich [Sag75] (see also [Sag94]). The non-linear case of the corollary was proved in [CE01].

The non-linear case of the corollary is tight, in fact

$$C = \{(1000), (0100), (0010), (0001)\}$$

is an equidistant $(4,4,2)$ code, but it is not separating. The linear case of the theorem is also tight, as the following example shows.

**Example 2.1** *From the proposition we get that if* $(\mathbf{0}, \mathbf{c}; \mathbf{a}, \mathbf{b})$ *is a binary* $(2,2)$*-NSC and* $4d_1 = 3m_1$*, then*

$$w(\mathbf{c}) = w(\mathbf{a}-\mathbf{b}) = w(\mathbf{a}+\mathbf{b}-\mathbf{c}) = m_1 = 4l,$$
$$w(\mathbf{a}) = w(\mathbf{b}) = w(\mathbf{a}-\mathbf{c}) = w(\mathbf{b}-\mathbf{c}) = d_1 = 3l.$$

*It turns out that the only possible* $(2,2)$*-NSC is the following, or replications thereof:*

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{c} \\ \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} 000000 \\ 111100 \\ 110010 \\ 101001 \end{bmatrix}.$$

*Note that the linear code* $\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle$ *has also* $d_1 = 3$ *and* $m_1 = 4$*.*

**Proposition 6**
If $C$ is binary, linear and $2d_1 > m_2$, then it is $(2,2)$-separating.

**Proof:** Let $(\mathbf{0}, \mathbf{c}; \mathbf{a}, \mathbf{b})$ be a $(2,2)$-NSC. We consider first the case where $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are linearly independent. Then $\mathbf{a}+\mathbf{b}$, $\mathbf{a}+\mathbf{b}+\mathbf{c}$, and $\mathbf{c}$ are the three non-zero codewords in some 2-dimensional subcode $D$. Thus we get that

$$w(\mathbf{c}) + w(\mathbf{a}-\mathbf{b}) + w(\mathbf{a}+\mathbf{b}-\mathbf{c}) = 2w(\langle \mathbf{a}+\mathbf{b}, \mathbf{c} \rangle) \leq 2m_2, \tag{2.6}$$

and by Proposition 5 that

$$4d_1 \leq \Sigma = w(\mathbf{c}) + w(\mathbf{a}-\mathbf{b}) + w(\mathbf{a}+\mathbf{b}-\mathbf{c}) \leq 2m_2.$$

If **a**, **b**, and **c** are linearily dependent, then $\mathbf{a}+\mathbf{b}+\mathbf{c}=\mathbf{0}$, and (2.6) becomes

$$w(\mathbf{c})+w(\mathbf{a}-\mathbf{b})+w(\mathbf{a}+\mathbf{b}-\mathbf{c}) \leq 2m_1,$$

which is stronger. □

It is easy to show that $m_2 \leq \lfloor 3m_1/2 \rfloor$, which is a maximum support weight analogue of the Griesmer bound. If this bound is not met with equality, then the above result is stronger than that of Theorem 2.

## 2.3 $(2,2)$-separating constructions

The next results provide a way to combine $(2,2)$-separating codes into new ones.

**Theorem 3** [NT78]
Let $M = M_1 M_2$, with $M_2 \geq M_1$ and $M_2$ is not divisible either by 2 or by 3. Suppose $C_1(n_1, M_1)$ and $C_2(n_2, M_2)$ are binary $(2,2)$-separating. Then there is a $(2,2)$-separating $(n, M)$ code with $n = n_1 + 4n_2$.

**Example 2.2** *Suppose we take $C_1 = C_2$ where $M_1 = 11, n_1 = 11$, then $M = 121, n = 55$. Applying Theorem 3 again, with $C_1(11,11)$ and $C_2(55,121)$, leads to a $(2,2)$-separating code with $M = 1331, n = 231$.*

*Take now $M_1 = M_2 = 13, n_1 = n_2 = 13$, then $M = 169, n = 65$.*
*Applying Theorem 3 a second time with $M_1 = 13, n_1 = 13$ and $M_2 = 169, n_1 = 65$ leads to a $(2,2)$-separating code with $M = 1859, n = 273$.*

It is relatively easy to apply Theorem 2 on codes with few weights, such as the following examples with two- and three-weight codes.

**Example 2.3** *Take a linear projective code over $\mathsf{GF}(p^2)$ [Cha90] with length*

$$n = \frac{(p^{k_1} - (-1)^{k_1})(p^{k_1-1} - (-1)^{k_1-1})}{p^2 - 1},$$

*dimension $k_1$ and weights $w_1 = p^{2k_1-3}, w_2 = p^{2k_1-3} + (-p)^{k_1-2}$.*

*In the case $p = 2$ for $k_1 = 4$ it gives a $(45, 4^4)_4$ code, which is $(2,2)$-separating since it satisfies $4d_1 > 3m_1$.*

**Example 2.4** *A three-weight code over $\mathsf{GF}(p)$ is given in [Cha90] with length*

$$n = p + 1 + p^2(p^{k_1-1} - (-1)^{k_1-1})(p^{k_1-2} - (-1)^{k_1-2})/(p-1),$$

*dimension $k = 2k_1$ and weights $w_1 = p^{2k_1-2} - (-p)^{k_1} - (-p)^{k_1-1}, w_2 = p^{2k_1-2}, w_3 = p^{2k_1-2} - (-p)^{k_1}$.*

*In the binary case for* $k = 6$ *it gives a* $(39, 2^6)_2$ *code, which is* $(2,2)$*-separating since it satisfies* $4d_1 > 3m_1$.

# Chapter 3

# Generalised separation and hashing

A $(n, m, \{w_1, w_2\})$-separating hash family, as defined in [SWZ00], is the same as a $(w_1, w_2)$-separating, $m$-ary code of cardinality $n$. The perfect hash families studied in [SWZ00] can also be viewed as a variant of separating codes, if we adopt the following definition.

**Definition 3**
A sequence $(T_1, \ldots, T_z)$ of pairwise disjoint vector sets is called a $(t_1, \ldots, t_z)$-configuration if $\#T_j = t_j$ for all $j$. Such a configuration is separated if there is a position $i$, such that for all $l \neq l'$ every vector of $T_l$ is different from every vector of $T_{l'}$ on position $i$.

A code is $(t_1, \ldots, t_z)$-separating if every $(t_1, \ldots, t_z)$-configuration is separated.

For $z = 2$, this definition coïncides with the previous one. A $(M, q, z)$-perfect hash family from [SWZ00] is a $(1, 1, \ldots, 1)$-separating (with $z$ ones) $(n, M)$ code. For brevity, we will say that such a code is $z$-hashing. The $(t, u)$-partial hashing [BCE+01] is $(1, 1, \ldots, 1, u-t)$-separation (with $t$ ones) in our terminology.

## 3.1 Basic results

Define

$$P(t_1, \ldots, t_z) := \sum_{i=1}^{z-1} \sum_{j=i+1}^{z} t_i t_j. \tag{3.1}$$

Note that if $t_j = 1$ for all $j$, then

$$P(t_1, \ldots, t_z) = \binom{z}{2}, \tag{3.2}$$

and if $z = 2$, then

$$P(t_1, t_2) = t_1 t_2. \tag{3.3}$$

The following proposition generalises the results on separating and perfect hashing families from [SWZ00, Alo86].

**Proposition 7**
An $(n, M, d)_q$ code $\Gamma$ is $(t_1, \ldots, t_z)$-separating if

$$\frac{d}{n} > 1 - \frac{1}{P(t_1, \ldots, t_z)}.$$

**Proof:** Suppose $\Gamma$ is non-$(t_1, \ldots, t_z)$-separating, and consider some non-separated $(t_1, \ldots, t_z)$-configuration $(T_1, \ldots, T_z)$ and the sum

$$\Sigma := \sum_{i=1}^{z-1} \sum_{j=i+1}^{z} \sum_{(x,y) \in T_i \times T_j} d(x, y).$$

This is the sum of $P(t_1, \ldots, t_z)$ distances in the code, so $\Sigma \geq P(t_1, \ldots, t_z)d$. Each coordinate can contribute at most $P(t_1, \ldots, t_z)$ to the sum $\Sigma$, but if any coordinate does contribute that much, then the configuration is separated on this coordinate. Hence each coordinate can contribute at most $P(t_1, \ldots, t_z) - 1$ to the sum $\Sigma$, and we get

$$P(t_1, \ldots, t_z)d \leq \Sigma \leq (P(t_1, \ldots, t_z) - 1)n.$$

Simplifying this, we get that any non-$(t_1, \ldots, t_z)$-separating code must satisfy

$$\frac{d}{n} \leq 1 - \frac{1}{P(t_1, \ldots, t_z)},$$

and the proposition follows. □

It must be noted that, to get infinite families of separating codes with good rate, the alphabet size $q$ grows extremely rapidly in the $t_j$-s, due to the Plotkin bound. However, better separating codes can be built by concatenation. Though this construction is well-known in various special cases from the literature [Alo86], we have not found as general a statement as the one we give below.

**Proposition 8**
If $\Gamma_1$ is a $(t_1, \ldots, t_z)$-separating, $M'$-ary $(n_1, M)$ code and $\Gamma_2$ a $(t_1, \ldots, t_z)$-separating, $q$-ary, $(n_2, M')$ code, then the concatenated code $\Gamma := \Gamma_2 \circ \Gamma_1$ is a $(t_1, \ldots, t_z)$-separating $(n_1 n_2, M)_q$ code.

**Proof:** Consider a $(t_1, \ldots, t_z)$-configuration $(T_1, \ldots, T_z)$ in $\Gamma$. Then there is a corresponding configuration in $\Gamma_1$, $(T''_1, \ldots, T''_z)$ which is separated on some coordinate $i$ by assumption. Considering only the positions of $\Gamma$ corresponding to position $i$ in $\Gamma_2$, we get a $(t'_1, \ldots, t'_z)$-configuration $(T'_1, \ldots, T'_z)$ in $\Gamma_1$ where $1 \leq t'_j \leq t_j$ for all $j$. Since also $\Gamma_1$ is $(t_1, \ldots, t_z)$-separating, $(T'_1, \ldots, T'_z)$ is separated on some position $j$. Hence $(T_1, \ldots, T_z)$ must be separated, as required. □

Note that the only thing we know about the minimum distance of $\Gamma$ is that it is at least equal to that of $\Gamma_1$. In general the concatenated code will not satisfy the requirement of Proposition 7. We will give a thorough example of the technique in Section 3.2.

**Proposition 9**
If $C$ is a linear, $(t_1, \ldots, t_z)$-separating code and $z \geq 3$, then $\sum_{j=1}^{z} t_j \leq q$.

Recall that the case when $z = 2$ was solved in Theorem 1.
**Proof:** First we prove that $t_1 + t_2 < q$, for if

$$T_1 \cup T_2 \supseteq \{\alpha \mathbf{c} \mid \alpha \in \mathsf{GF}(q)\},$$

then no third set $T_3$ will be separated from $T_1$ and $T_2$.

Let $\alpha_0, \alpha_1, \ldots, \alpha_{q-1}$ be all the field elements, where $\alpha_0 = 0$ and $\alpha_1 = 1$. Let $\mathbf{a}$ and $\mathbf{b}$ be two independent codewords. Let

$$T_1 := \{\alpha_0 \mathbf{a}, \ldots, \alpha_{t_1 - 1} \mathbf{a}\},$$
$$T_2 := \{\alpha_{t_1} \mathbf{a}, \ldots, \alpha_{t_2 - 1} \mathbf{a}\},$$

and let $T_3, \ldots, T_z$ be any sequence of pairwise disjoint sets such that

$$T := \bigcup_{j=3}^{z} T_j = \{\mathbf{a} + \alpha_1 \mathbf{b}, \ldots, \mathbf{a} + \alpha_{t'} \mathbf{b}\},$$

where $t' = t_3 + \ldots + t_z$. Clearly, $T_1$ and $T_2$ are only separated on $\chi(\mathbf{a})$. Also $T$ and $T_1$ are only separated on $\chi(\mathbf{b})$. On any coordinate $i \in \chi(\mathbf{a}) \cap \chi(\mathbf{b})$, $t_1 + t_2$ different values occur in $T_1 \cup T_2$ and $t'$ different values occur in $T$. Hence the configuration can only be separated if

$$t' + t_1 + t_2 = t_1 + \ldots + t_z \leq q,$$

as required. □

**Corollary 3**
If $C$ is a linear $q$-ary, $(t, u)$-partially hashing code with $t \geq 2$, then $u \leq q$.

These bounds are tight, since $q$-hashing codes can be constructed for any $q$.

## 3.2 The tetracode and compositions thereof

The ternary constructions will make use of three ingredient codes, and apply twice the concatenation method.

The first seed is the remarkable $[4,2,3]_3$ tetracode $\mathfrak{T}$, defined by the generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix}.$$

This code is self-dual and MDS (on Singleton's bound $d = n - k + 1$). It is both an extended perfect Hamming code and a simplex (all codewords are at distance 3 apart). The next result is a consequence of Theorem 2 and Proposition 7.

**Proposition 10**
The $[4,2,3]_3$ tetracode is both $(2,2)$- and $(3,1)$-separating, and 3-hashing.

This proposition is not new. The tetracode was proved 3-hashing in [KM88], and it was proved to have the IPP property in [HvLLT98]. The codes constructed in the sequel are also 3-hashing, but we do not claim that they are particularly good compared to existing codes. Our main interest is $(2,2)$- and $(3,1)$-separation after all.

Let $\mathfrak{R}_1$ be the $[9,3,7]_{3^2}$ Reed-Solomon code, which is both $(2,2)$- and $(1,3)$-separating, and 3-hashing by Proposition 7. The concatenated code $\mathfrak{T} \circ \mathfrak{R}_1$ has parameters $[36,6]_3$, and by Proposition 8, it is $(2,2)$- and $(1,3)$-separating, and 3-hashing. In order to produce infinite families of separating codes, we need the following constructive result from Tsfasmann [Tsf91].

**Lemma 1**
For any $\alpha > 0$ there is an infinite families of codes $\mathfrak{A}(N)$ with parameters $[N, NR, N\delta]_q$ for $N \geq N_0(\alpha)$ and

$$R + \delta \geq 1 - (\sqrt{q} - 1)^{-1} - \alpha.$$

We should note that the rate of $\mathfrak{A}(N)$ is interesting only for large $q$, but $\mathfrak{T} \circ \mathfrak{R}_1$ allows for concatenation with $\mathfrak{A}(N)$ over $\mathsf{GF}(3^6)$ which may be acceptable. Thus consider the family of $[N, K, D = \lceil 3N/4 \rceil + 1]_{3^6}$ codes $\mathfrak{A}(N)$, which has rate $R' \approx 1/4 - (3^3 - 1)^{-1} = 11/52$. The concatenated code $\mathfrak{T} \circ \mathfrak{R}_1 \circ \mathfrak{A}(N)$ gives an infinite family of linear, ternary $(3,1)$- and $(2,2)$-separating and 3-hashing codes with rate $R'/6 \approx 0.0352$.

If we only want $(3,1)$-separating and 3-hashing codes, we can obtain a better rate by using the Reed-Solomon code $\mathfrak{R}_2$ with parameters $[10,4,7]_{3^2}$, which results in the concatenated code $\mathfrak{T} \circ \mathfrak{R}_2$ with parameters $[40,8]_3$. Then we take the infinite family $\mathfrak{A}(N)$ of codes with parameters $[N, K, D = \lceil 2N/3 \rceil + 1]_{3^8}$ of rate

$1/3-(3^4-1)^{-1}$, and the concatenated code $\mathfrak{T}\circ\mathfrak{R}_2\circ\mathfrak{A}(N)$ is an infinite family of linear ternary $(3,1)$-separating and 3-hashing codes with rate approximately $77/1200\approx 0.0642$.

**Example 3.1** *We sketch a construction with $q=4$ as well. As in the previous example, we concatenate three codes to build the infinite family. Each code has $d/n>3/4$ and thus is $(2,2)$-separating by Proposition 7. The first two are doubly extended Reed-Solomon codes. We take successively:*

1. $C_1[5,2,4]_4$*;*
2. $C_2[17,5,13]_{4^2}$*, getting* $C_1\circ C_2[85,10]_4$*;*
3. *and finally,* $C(N)[N,K,D=\lceil 3N/4\rceil+1]_{4^{10}}$ *with rate* $\approx 1/4-(4^5-1)^{-1}\approx 1/4$.

*The final outcome is an infinite constructive family of linear quaternary $(2,2)$-separating codes with rate approximately $1/34\approx 0.029$.*

# Chapter 4

# Intersecting codes

Intersecting codes is another known concept from the literature. In this chapter we will see how we can construct separating codes from intersecting codes.

**Definition 4**
A linear code $C$ of dimension $k \geq t$ is said to be $t$-wise intersecting if any $t$ linearly independent codewords have intersecting supports. If $t > k$, we say that $C$ is $t$-wise intersecting if and only if it is $k$-wise intersecting.

It is easy to verify that any $t$-wise intersecting code is also $(t-1)$-wise intersecting.

## 4.1 How intersecting codes give separation

**Proposition 11**
For a linear, binary code, the following properties are equivalent:

1. 3-wise intersection;

2. $(2,2)$-separation.

The fact that linear, $(2,2)$-separating codes must be 3-wise intersecting holds not only for binary codes, and it is proved in Proposition 12 below. Unfortunately, this is the only result we have found in the non-binary case.

The fact that $t$-wise intersecting, binary codes gives rise to separating codes can be generalised. The statement of the proposition, to the effect that 1 implies 2, will follow from the special case $t = 3, j = 2$ of Proposition 14 below.

**Proposition 12**
Every linear $(2,2)$-separating code is 3-wise intersecting.

**Proof:** If $k = 2$, three-wise intersection is equivalent to 2-wise intersection according to our definition. Any $(2,2)$-separating code is $(2,1)$-separating and hence 2-wise-intersecting by [CE00a].

Suppose $C$ is $(2,2)$-separating, and consider three independent codewords $\mathbf{a}, \mathbf{b}, \mathbf{c}$. We shall prove that these three words have intersecting supports. Consider the $(2,2)$-configuration $(\mathbf{0}, \mathbf{c}+\mathbf{a}; \mathbf{a}, \mathbf{b})$. Since $C$ is $(2,2)$-separating, there is a position $i$ where $\mathbf{a}$ is $\alpha \neq 0$ and $\mathbf{b}$ is $\beta \neq 0$, and $\mathbf{c}+\mathbf{a}$ is $\gamma \notin \{\alpha, \beta\}$. Now $\mathbf{c}$ is $\gamma - \alpha \neq 0$ on position $i$. □

Due to this proposition, we can use many bounds on separating codes as bounds on intersecting codes. For instance, by Theorem 2, every code with $4d > 3m$ is 3-wise intersecting. In the binary case, we get that $2d > m_2$ implies 3-wise intersection by Proposition 6.

**Proposition 13**
If $C$ is a $t$-wise intersecting, binary, linear code, and $\Gamma \subseteq C$ is a nonlinear subcode such that any $t$ non-zero codewords are linearly independent, then $\Gamma$ is $(j, t+1-j)$-separating for all $j$ such that $1 \leq j \leq t$.

**Proof:** Choose any (two-part) sequence $Y'$ of $t+1$ codewords from $\Gamma$,

$$Y' := (\mathbf{a}'_1, \ldots, \mathbf{a}'_j; \mathbf{c}'_1, \ldots, \mathbf{c}'_{t+1-j}).$$

By Remark 1.2, $Y'$ is $(j, t+1-j)$-separated if and only if $Y := Y' - \mathbf{c}'_{t+1-j}$ is. Hence it suffices to show that

$$Y = (\mathbf{a}_1, \ldots, \mathbf{a}_j; \mathbf{c}_1, \ldots, \mathbf{c}_{t-j}, \mathbf{0})$$

is $(j, t+1-j)$-separated.

Since any $t$ codewords in $Y'$ are linearly independent, so are the $t$ first codewords of $Y$. The zero vector is of course not independent of anything.

Now, consider

$$\{\mathbf{a}_1 + \mathbf{c}_1, \ldots, \mathbf{a}_1 + \mathbf{c}_{t-j}; \mathbf{a}_1, \ldots, \mathbf{a}_j\},$$

which is a set of linearly independent codewords from $C$, and hence all non-zero on some coordinate $i$. Since $\mathbf{a}_1 + \mathbf{c}_l$ is non-zero on coordinate $i$, $\mathbf{c}_l$ must be zero for all $l$. Hence $Y$, and consequently $Y'$, is separated on coordinate $i$. □

**Proposition 14**
If $C$ is a $t$-wise intersecting binary linear code, and $\Gamma \subseteq C$ is a nonlinear subcode such that any $t-1$ non-zero codewords are linearly independent, then $\Gamma$ is $(j, t+1-j)$-separating for all even $j$ such that $1 < j \leq t$.

**Proof:** We define $Y$ as in the previous proof, and the $t-1$ first codewords of $Y$ are linearly independent. If $\mathbf{c}_{t-j}$ is linearly independent of the others, then we are

done by the first proof; hence we assume that $\mathbf{c}_{t-j}$ is dependent on the $t-1$ first codewords, and since any $t-1$ codewords are independent, it must in fact be the sum of the $t-1$ first codewords.

By the same argument as in the previous proof, we get one coordinate $i$, where $\mathbf{a}_1+\mathbf{c}_1,\ldots,\mathbf{a}_1+\mathbf{c}_{t-1-j},\mathbf{a}_1,\ldots,\mathbf{a}_j$ are all one, and $\mathbf{c}_1,\ldots,\mathbf{c}_{t-1-j}$ are zero. Now, $\mathbf{c}_{t-j}$ is the sum of the $t-1$ first codewords, of which $j$ are 1 and the rest are zero on coordinate $i$. Since $j$ is even, $\mathbf{c}_{t-j}$ is zero and $Y$ is separated. □

Note that if $t$ is even, then either $j$ or $t+1-j$ is even; thus we get the following corollary.

**Corollary 4**
If $C$ is a $t$-wise intersecting, linear, binary code for even $t$, and $\Gamma \subseteq C$ is a nonlinear subcode such that any $t-1$ non-zero codewords are linearly independent, then $\Gamma$ is $(j,t+1-j)$-separating for all $j$ such that $1 \leq j \leq t$.

A 3-wise intersecting, binary code is $(2,2)$-separating according to the above proposition, but no binary linear code is $(3,1)$-separating; hence the restriction that $j$ be even cannot be dropped in general.

It is perhaps not obvious how these propositions may be used to construct non-linear separating codes with a reasonable rate. The remainder of the section is devoted to explaining this.

**Lemma 2**
Given an $[n,rm+1]$ linear, binary code $C$, we can extract a non-linear subcode $\Gamma$ of size $2^r+1$ such that any $2m+1$ codewords are linearly independent.

Note that the rate of $\Gamma$ is approximately $R/m$ where $R=(rm+1)/n$ is the rate of $C$.

**Proof:** Let $C'$ be the $[2^r, 2^r-1-rm, 2m+2]$ extended BCH code. The columns of the parity check matrix of $C'$ make a set $\Gamma'$ of $2^r$ vectors from $\mathsf{GF}(2)^{rm+1}$, such that any $2m+1$ of them are linearly independent. Now there is an isomorphism $\phi:\ \mathsf{GF}(2)^{rm+1} \to C$, so let $\Gamma = \phi(\Gamma') \cup \{\mathbf{0}\}$. □

**Theorem 4**
Given an $[n,nR]$ $t$-wise intersecting binary (asymptotical) code, there is a construction of a non-linear code $\Gamma$ of rate $R/\lfloor (t-1)/2 \rfloor$, which is $(j,t+1-j)$-separating.

**Proof:** First consider $t$ even, and write $t=2m+2$, where $m>0$, the case $t=2$ is trivial from Proposition 11, anyway. By Proposition 14, we want to extract $\Gamma$ such that any $2m+1$ codewords are independent, and such $\Gamma$ exist with rate $R/m$ by Lemma 2.

Then consider odd $t$, and write $t = 2m+1$, where $m > 0$. By Proposition 13, we want to extract $\Gamma$ such that any $2m+1$ codewords are independent, and such $\Gamma$ exist with rate $R/m$ by Lemma 2. □

**Example 4.1** *In [CZ94], it was shown that for sufficiently large $n$, and for any rate $R < 1 - \frac{1}{t}\log(2^t - 1)$, there are $t$-wise intersecting linear, binary $[n,k]$ codes of rate $R$. Though non-constructive, this result guarantees the existence, for any $t$, of non-linear, binary codes which are $(j, t+1-j)$-separating for all $j$ and have rate arbitrarily close to*

$$\frac{1 - \frac{1}{t}\log(2^t - 1)}{\lfloor (t-1)/2 \rfloor}.$$

## 4.2 Binary constructions

In this section we will give some sample constructions of separating codes, based on results on $t$-wise intersection from [CZ94].

**Proposition 15** [CZ94]
The punctured dual of the 2-error-correcting BCH code with parameters $[2^{2t+1} - 2, 4t+2, 2^{2t} - 2^t - 1]$, is $t$-wise intersecting.

**Example 4.2** *For $t = 4$, we get from Proposition 15 a 4-wise intersecting code with parameters $[2^9 - 2, 18]$. Now the shortende code $\Gamma'$ of the $2^{17}$ codewords having a 1 in the last position (say) is clearly such that any 3 of its elements are independent, thus we get a $(3,2)$-separating $(2^9 - 3, 2^{17} + 1)$ code $\Gamma := \Gamma' \cup \{\mathbf{0}\}$ by Corollary 4. We can concatenate $\Gamma$ with the code $\mathfrak{A}(N)$ with parameters $[N, RN, 5N/6 + 1]_{2^{18}}$ from Lemma 1 to get a $(3,2)$-separating code with rate $R \approx 0.00557$.*

If $\Gamma(n, M)$ is $(t, t')$-separating, then so are the 2 subcodes $\Gamma_0$ (resp. $\Gamma_1$) having 0 (resp. 1) in the first coordinate. Taking the largest and removing the first coordinate (which no longer separates anything), gives a shortened $(n-1, \lceil M/2 \rceil)$ $(t, t')$-separating code.

**Proposition 16**
There is a constructive infinite sequence of binary $(j, t+1-j)$-separating codes of rate $2^{-3(t-1)}(1 + o(1))$.

This proposition follows directly from the following lemma:

**Lemma 3** [CZ94]
There is a constructive infinite sequence of $t$-wise intersecting binary codes with rate arbitrarily close to

$$R_t = \left(2^{1-t} - \frac{1}{2^{2t+1}-1}\right)\frac{2t+1}{2^{2t}-1} = 2^{2-3t}(t+o(t)).$$

**Proof:** By concatenating geometric $[N, K, D]_q$ codes from Lemma 1 satisfying $D > N(1-2^{1-t})$ with $q = 2^{4t+2}$, and with a rate arbitrarily close to $2^{1-t} - 1/(\sqrt{q}-1)$, with the $[2^{2t+1}-2, 4t+2, 2^{2t}-2^t-1]$ code of Proposition 15, we obtain the result. □

**Example 4.3** *The 3-wise binary intersecting* $[126, 14]$ *code (case* $t = 3$ *of Proposition 15), yields a* $(2,2)$*-separating code with parameters* $(126, 2^{14})$.

**Example 4.4** *Let* $q = p^{2m}$. *Consider (see Lemma 1) a family of codes* $\mathfrak{A}(N)$ *with parameters* $[N, NR, N\delta]_q$ *with* $N \geq N_0(\alpha)$ *and*

$$R + \delta \geq 1 - (p^m - 1)^{-1} - \alpha.$$

*Choosing* $p = 2, m = 7$, $\delta = 3/4 + \epsilon$, *(see Proposition 7) and concatenating* $\mathfrak{A}(N)$ *and* $C$, *the binary* $[126, 14, 55]$ *code, yields a constructive infinite sequence* $\{\mathfrak{A}(N) \circ C\}_N$ *of binary linear* $(2,2)$*-separating codes with rates arbitrarily close to* $0.026$.

## 4.3 Upper bounds on intersecting codes

We now present an upper bound on the rate of such codes.

**Theorem 5**
A $t$-wise intersecting code $C_t[n,k,d]$ gives rise by projection to a $(t-1)$-wise intersecting code $C_{t-1}[d, k-1]$.

**Proof:** Let $a \in C$ be a fixed element of minimum weight $d$. Denote by $C_a$ the $[n, k-1]$ supplementary subspace of $\{\mathbf{0}, a\}$ in $C$. Consider any $(t-1)$ independent codewords $\{b^1, \ldots, b^{t-1}\}$ in $C_a$. Then $\{a, b^1, \ldots, b^{t-1}\}$ is full rank, hence these $t$ codewords of $C$ intersect (on the support of $a$). Thus $C/a$, the projection of $C_a$ on the support of $a$ is a $(t-1)$-intersecting $[d, k-1]$ code. □

To get an upper bound on the dimension of such codes in the binary case, we use recursively any upper bound from coding theory, for instance the McEliece et al. bound (see [MS77]):

$$R \leq H_2\left(\frac{1}{2} - \sqrt{\frac{d}{n}\left(1 - \frac{d}{n}\right)}\right).$$

For $t = 3$, we get the following sequence of codes:

$$C_3[n, k, d], \quad C_2[d, k-1, d'], \quad C_1[d', k-2],$$

where $C_i$ is $i$-wise intersecting, and has write rate $R_i$.

From $C_1$, we have that $k - 2 \leq d'$, which implies that

$$R_2 = (k-1)/d \leq (d'-1)/d \leq d'/d.$$

By the McEliece bound, this implies $R_2 \leq 0.28$. Finally we have

$$R_1 = \frac{k}{n} \leq \frac{0.28d + 1}{n} \leq 0.108,$$

where the final bound follows by applying again the McEliece bound. The following corollary arise from the same technique and some other values for $t$.

**Corollary 5**
The asymptotic rate of the largest $t$-wise intersecting binary code is at most $R_t$, with $R_2 \approx 0.28, R_3 \approx 0.108, R_4 \approx 0.046, R_5 \approx 0.021, R_6 \approx 0.0099$.

Note that the McEliece bound is only valid asymptotically. In particular, the $[126, 14]$ 3-wise intersecting code from Example 4.3 has rate $1/9 > R_3$.

## 4.4 An upper bound on separation

An extension of arguments from [Sag94] and [KS88] gives the following theorem.

**Theorem 6**
If $\Gamma_i(n = d_{i-1}, M_i, d_i)$ is $(i, j)$-separating, then it gives rise by projection to a code $\Gamma_{i+1}$ with the following properties:

1. $\Gamma_{i+1}$ is $(i-1, j-1)$-separating, with parameters $(d_i, M_{i+1} = M_i - 2, d_{i+1})$;
2. $q^{d_{i+1}} \geq M_i - 2$.

**Proof:** It is based on the case $q = i = j = 2$ studied in [KS88]. Let $\Gamma_i$ be like in the statement of the theorem. With no loss of generality, it contains two codewords $a, a'$ differing exactly in the first $d_i$ coordinates. Now any two non-intersecting coalitions $T, T'$ of respective sizes $i$ and $j$, with $a \in T, a' \in T'$ have nonintersecting feasible sets by hypothesis. Since $a$ and $a'$ coincide on coordinates $\{d_i + 1, \ldots, n\}$, this property is kept by projecting on $\{1, \ldots, d_i\}$ and removing $a$ and $a'$ from their respective coalitions. This proves 1. To get 2, just note that all projections must be different at the first iteration (when $i = 2$) and apply induction. □

We have, by use of non-binary linear programming bounds (see [Aal90]):

$$R(\delta) \leq H_q(((q-1)-(q-2)\delta - 2\sqrt{(q-1)\delta(1-\delta)})/q).$$

In the ternary case, this gives the following upperbounds on the rates of $(t,t)$-separating codes : $0.357, 0.168, 0.09$ for $t = 2,3,4$ respectively.

In the linear ternary case, the previous bounds are shifted, giving $0.168, 0.09$ for $t = 2,3$ respectively.

# Chapter 5

# Asymptotic Results

## 5.1 Binary Codes

In Table 5.1, we present some upper and lower bounds for codes with different separating capabilities. Most of the bounds are known from previous works, the rest is given in form of examples in this section.

**Example 5.1** *Let $C$ be a 4-wise intersecting code with rate*

$$R \approx R = 1 - \frac{\log 15}{4} \approx 0.0233,$$

*as described in Example 4.1. Make a non-linear subcode $\Gamma \subseteq C$ as in Lemma 2 such that any three codewords are linearly independent. This gives a $(3,2)$-separating code with rate $R/\lfloor (t-1)/2 \rfloor = R \approx 0.0233$.*

**Lemma 4**
Given an $[n, rm]$ linear, binary code $C$, we can extract a non-linear subcode $\Gamma$ of size $2^r$ such that any $2m$ non-zero codewords are linearly independent.

**Proof:** Let $C'$ be the $[2^r - 1, 2^r - 1 - rm, 2m+1]$ BCH code. The columns of the parity check matrix of $C'$ make a set $\Gamma'$ of $2^r - 1$ vectors from $\mathsf{GF}(2)^{rm}$, such that no $2m$ of them are linearly independent. Now there is an isomorphism $\phi : \mathsf{GF}(2)^{rm} \to C$, so let $\Gamma = \phi(\Gamma') \cup \{\mathbf{0}\}$. □

**Example 5.2** *For $t = 5$, we get from Proposition 15 a 5-wise intersecting code with parameters $[2^{11} - 2, 22]$, which leads to a non-linear $(3,3)$-separating code $\Gamma$ with rate $R' \approx 11/2046$ by Lemma 4. We concatenate $\Gamma$ with the code $\mathfrak{A}(N)$ with parameters $[N, RN, 8N/9+1]_{2^{22}}$ from Lemma 1 to obtain a $(3,3)$-separating code with rate $R \approx 1/1674 = 0.000597$.*

| | Lower bounds | | | | Upper bounds | |
|---|---|---|---|---|---|---|
| | Linear | | Non-Linear | | Linear | No |
| $(t,t')$ | Const. | Non-const. | Const. | Non-const. | | |
| $(2,1)$ | 0.156[1] | 0.21[1] | — | 0.21 [Kör95] | 0.28[1] | 0.5 |
| $(3,1)$ | N/A | N/A | 0.0448 [CE00a] | — | N/A | |
| $(4,1)$ | N/A | N/A | 0.0181 [CE00a] | — | N/A | |
| $(2,2)$ | 0.026[4] | 0.0642 [Sag94] | — | 0.0642 [Sag94] | 0.108 [CEL01] | 0.28 |
| $(3,2)$ | N/A | N/A | 0.00557[2] | 0.0233[3] | N/A | |
| $(3,3)$ | N/A | N/A | 0.000597[5] | 0.0156 [BCE+01] | N/A | 0.065 |

[1] Bounds from intersecting codes [CZ94].
[2] Example 4.2.
[3] Example 5.1.
[4] Example 4.4.
[5] Example 5.2.

Table 5.1: Bounds on rates for infinite families of binary codes with various separating properties.

## 5.2 On linear $(2,2)$-separation

We will dwell a little extra on the case of $(2,2)$-separation, and present existence proof of codes with certain rates over different fields. The first lemma is fairly well-known, and can be found in [MS77].

**Lemma 5**
Asymptotically, for almost all codes, we have

$$A_i = \frac{\binom{n}{i}(q-1)^i}{q^{n(1-R)}} \approx \frac{e^{nH(i/n)}2^{i\ln(q-1)}}{e^{n(1-R)\ln q}},$$

where $H$ is the natural entropy function and $R = k/n$ is the rate.

Since we are dealing with the asymptotical case, we normalise by setting $i = n\omega$, and we define a function $f(\omega, R, q)$ by

$$A_{\omega n} = A_i = e^{nf(\omega,R,q)}.$$

From Lemma 5, we get

$$f(\omega, R, q) = H(\omega) + \omega\ln(q-1) - (1-R)\ln q. \tag{5.1}$$

Note that for a given $A_i$, there are two solutions for $i$. Setting $A_i \approx 1$, the two solutions will be the minimum and the maximum weights. These are of course also the zeroes of $f$.

| | | Technique I | | Technique II | | Others |
|---|---|---|---|---|---|---|
| $q$ | $\delta_{\max}$ | $\delta$ | Rate | $\delta$ | Rate | Rate |
| 2 | 0.5000 | 0.75 | N/A | 0.4286 | 0.01477 | 0.0642 [Sag94] |
| 3 | 0.6667 | 0.75 | N/A | 0.5695 | 0.01859 | 0.0352[1] |
| 4 | 0.7500 | 0.75 | N/A | 0.6385 | 0.02206 | 0.029[2] |
| 5 | 0.8000 | 0.75 | 0.00459 | 0.6786 | 0.02532 | |
| 7 | 0.8571 | 0.75 | 0.02043 | 0.7218 | 0.03153 | |
| 8 | 0.8750 | 0.75 | 0.02774 | 0.7340 | 0.03457 | |
| 9 | 0.8889 | 0.75 | 0.03427 | 0.7426 | 0.03766 | |
| 11 | 0.9091 | 0.75 | 0.04530 | N/A | N/A | |
| 13 | 0.9231 | 0.75 | 0.05417 | N/A | N/A | |
| 16 | 0.9375 | 0.75 | 0.06464 | N/A | N/A | |
| 17 | 0.9412 | 0.75 | 0.06757 | N/A | N/A | |
| 19 | 0.9474 | 0.75 | 0.07279 | N/A | N/A | |

[1] The concatenated code $\mathfrak{T} \circ \mathfrak{R}_1 \circ \mathfrak{A}(N)$ from Section 3.2. (Constructive.)

[2] Example 3.1. (Constructive.)

Table 5.2: Rates for which we can guarantee the asymptotical existence of linear $(2, 2)$-separating codes. The number $\delta_{\max} = (q-1)/q$ is the maximum possible minimum distance by the Plotkin bound.

Let $\delta = d_1/n$ and $\mu = m_1/n$ be respectively the minimum and maximum normalised weights. Because $\mu$ and $\delta$ are the zeroes of $f$, we get

$$H(\delta) + \delta \ln(q-1) = H(\mu) + \mu \ln(q-1),$$

or

$$\begin{aligned}(\delta - \mu)\ln(q-1) = {} & \delta \ln \delta + (1-\delta)\ln(1-\delta) \\ & - \mu \ln \mu - (1-\mu)\ln(1-\mu).\end{aligned} \tag{5.2}$$

**Lemma 6 (Varshamov-Gilbert)**
For almost all codes, the rate and the normalised minimum distance are related by the following equation

$$H(\delta) + \delta \ln(q-1) = (1-R)\ln q.$$

**Proof:** This follows from equating $f(\omega, R, q) = 0$ as in (5.1). □

We know from Proposition 7 that if $\delta > 3/4$, then the code is $(2, 2)$-separating, hence we can, by substituting $\delta = 3/4$ in the Gilbert-Varshamov equation, get

rates for which asymptotically almost any code is $(2,2)$-separating. The rates such obtained are presented under 'Technique I' in Table 5.2. Due to the Plotkin bound, this does not give anything over small fields.

Technique II in the table is an improvement based on Theorem 2, which says that every code with $4\delta > 3\mu$ is 2-separating. We insert $\delta = 4\mu/3$ in (5.2), and get

$$\begin{aligned}\frac{\delta}{3}\ln(q-1) = \delta\ln\delta + (1-\delta)\ln(1-\delta) \\ -\frac{4\delta}{3}\ln\frac{4\delta}{3} - (1-\frac{4\delta}{3})\ln(1-\frac{4\delta}{3}),\end{aligned} \tag{5.3}$$

We have solved this equation numerically for the smallest fields, and the results are given in Table 5.2. Of course, we will always have

$$0 \leq \delta \leq \mu \leq 1,$$

which will bound $\delta \leq 3/4$ in (5.3). This results in no real solution of (5.3) for $q \geq 11$.

# Chapter 6

# Anticodes and Maximum Weights

We have used minimum and maximum support weights a couple of times in this report. Whereas minimum support weights are much studied, the results on maximum weights are rare. It is well-known though, that for linear codes, we have a connection between minimum and maximum weights through anticodes. In fact, any bound on minimum weights can be turned into a bound on maximum weights.

Some of the idea behind maximum support weights is that $m_{t-1}$ provide a bound on the size of $F(T)$ for any coalition $T$ of size $t$. A coalition is a subset $T \subseteq C$, and it is clear that the detectable bits are those in $\tilde{\chi}(T)$. Hence $\#F(T) \leq q^{m_{t-1}}$ where $t = \#T$.

## 6.1 Preliminaries

A linear $[n, k]$ code $C$ over $\mathsf{GF}(q)$ can be represented by a projective multiset

$$\gamma : \mathsf{PG}(k-1, q) \to \{0, 1, 2, \ldots\},$$

where $\mathsf{PG}(k-1, q)$ is the projective $(k-1)$-space over $\mathsf{GF}(q)$, and $\gamma(x)$ is the number of times $x$ occurs as a column in the generator matrix $G$ of $C$.

We extend the multiset to the power set:

$$\gamma(S) = \sum_{x \in S} \gamma(x), \quad \forall S \subseteq \mathsf{PG}(k-1, q).$$

The code has length $n = \gamma(\mathsf{PG}(k-1, q))$. The number $\gamma(S)$ is called the value of $S$.

Let

$$\delta_0 := \max\{\gamma(x) \mid x \in \mathsf{PG}(k-1, q)\}.$$

If $\delta_0 = 1$, $\gamma$ is a set and $C$ is a projective code.

The *anticode* of $C$ is defined [DS98] by the multiset $\gamma'$, given by

$$\gamma'(x) = \delta_0 - \gamma(x), \quad x \in \mathsf{PG}(k-1,q). \tag{6.1}$$

We have defined the minimum and maximum support weights $d_r$ and $m_r$ for the code $C$. From previous works on projective multisets [HKY92, TV95] it is well known that for each subcode $D_r \subseteq C$ of dimension $r$, there is is a subspace $\Pi_{k-1-r} \subseteq \mathsf{PG}(k-1,q)$ of codimension $r$, such that

$$\gamma(\Pi_{k-1-r}) + w(D_r) = n.$$

Therefore the maximum value of a subspace of codimension $r$ is $n - d_r$, and the minimum value of such a subspace is $n - m_r$. It follows from the same argument that $d_k - d_{k-1} = \delta_0$.

Consider the value of $\Pi_{k-1-r}$ in the anticode. There are $(q^{k-r}-1)/(q-1)$ points in $\Pi_{k-1-r}$. Hence we get from (6.1) that

$$\gamma'(\Pi_{k-1-r}) = \delta_0 \frac{q^{k-r}-1}{q-1} - \gamma(\Pi_{k-1-r}). \tag{6.2}$$

It is obvious from this equation that a subcode of minimum weight in $C$ has maximum weight in the anticode (and vice versa).

**Lemma 7**
If $C$ is an $[n,k]$ code, then the length of its anticode is

$$n' = \delta_0 \frac{q^k-1}{q-1} - n.$$

**Proof:** From (6.2), we get that

$$n' = \gamma_{C'}(\mathsf{PG}(k-1,q)) = \delta_0 \frac{q^k-1}{q-1} - \gamma_C(\mathsf{PG}(k-1,q)) = \delta_0 \frac{q^k-1}{q-1} - n.$$

□

**Lemma 8**
If $(d_1, \ldots, d_k)$ is the weight hierarchy of $C$, and $(d'_1, \ldots, d'_{k'})$ the weight hierarchy of its anticode, then

$$m'_r = (d_k - d_{k-1}) \frac{q^k - q^{k-r}}{q-1} - d_r,$$
$$d'_r = (d_k - d_{k-1}) \frac{q^k - q^{k-r}}{q-1} - m_r.$$

**Proof:** From (6.2) we get that

$$n' - m'_r = (d_k - d_{k-1})\frac{q^{k-r}-1}{q-1} - (n - d_r),$$
$$n' - d'_r = (d_k - d_{k-1})\frac{q^{k-r}-1}{q-1} - (n - m_r).$$

If we apply Lemma 7, we get

$$\begin{aligned} m'_r &= \delta_0\frac{q^k-1}{q-1} - n - \delta_0\frac{q^{k-r}-1}{q-1} + (n - d_r) \\ &= \delta_0\frac{q^k - q^{k-r}}{q-1} - d_r, \\ d'_r &= (d_k - d_{k-1})\frac{q^{k-r}-1}{q-1} - (n - m_r) \\ &= \delta_0\frac{q^k - q^{k-r}}{q-1} - m_r, \end{aligned}$$

as required. □

To find codes with low values for $m_t$ we can search for codes with high minimum support weights $d_t$ and a low value $\delta_0$. Probably we can restrict ourselves to projective codes, such that $\delta_0 = 1$.

**Remark 6.1**
Let $C'$ be the anticode of $C$. Then $C$ is the anticode of $C'$ if and only if there is some point $x \in \mathsf{PG}(k-1, q)$ of value $\gamma_C(x) = 0$. In other words if and only if $\delta_0(C) = \delta_0(C')$.

## 6.2 Turning the Griesmer Bound

A well-known bound on the minimum support weights, is the Griesmer Bound, which we state below. We will show that, via anticodes, the Griesmer Bound induces a bound on the maximum support weights.

**Lemma 9 (Griesmer Bound)**
For any linear code, we have

$$d_r \geq \sum_{i=0}^{r-1} \left\lceil \frac{d_1}{q^i} \right\rceil.$$

**Proposition 17 (Remseirg Bound)**
For any linear code, we have

$$m_r \leq \sum_{i=0}^{r-1} \left\lfloor \frac{m_1}{q^i} \right\rfloor .$$

**Proof:** From Lemma 8, we have

$$m_r = \delta_0' \frac{q^k - q^{k-r}}{q-1} - d_r'.$$

We apply the Griesmer bound to get

$$m_r \leq \delta_0' \frac{q^k - q^{k-r}}{q-1} - \sum_{i=0}^{r-1} \left\lceil \frac{d_1'}{q^i} \right\rceil .$$

We apply again Lemma 8 to get

$$\begin{aligned} m_r &\leq \delta_0' \frac{q^k - q^{k-r}}{q-1} - \sum_{i=0}^{r-1} \left\lceil \frac{1}{q^i} \left( \delta_0 \frac{q^k - q^{k-1}}{q-1} - m_1 \right) \right\rceil . \\ &= \delta_0' \frac{q^k - q^{k-r}}{q-1} - \sum_{i=0}^{r-1} \left\lceil \delta_0 q^{k-1-i} - \frac{m_1}{q^i} \right\rceil \\ &= \delta_0' q^{k-r} \frac{q^r - 1}{q-1} - \delta_0 \sum_{i=1}^{r} q^{k-i} - \sum_{i=0}^{r-1} \left\lceil -\frac{m_1}{q^i} \right\rceil \\ &= (\delta_0' - \delta_0) q^{k-r} \frac{q^r - 1}{q-1} + \sum_{i=0}^{r-1} \left\lfloor \frac{m_1}{q^i} \right\rfloor . \end{aligned}$$

By the definition of anticodes (6.1), it is clear that $\delta_0' \leq \delta_0$, thus the proposition follows. □

# Bibliography


[Aal90] Matti Aaltonen. A new upper bound on nonbinary block codes. *Discrete Math.*, 83(2-3):139–160, 1990. 4.4

[Alo86] N. Alon. Explicit construction of exponential sized families of $k$-independent sets. *Discrete Math.*, 58(2):191–193, 1986. 3.1, 3.1

[BCE+01] A. Barg, G. Cohen, S. Encheva, G. Kabatiansky, and G. Zémor. A hypergraph approach to the identifying parent property. Submitted to SIAM J. on Disc. Methods., 2001. 3, 5.1

[BS98] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inform. Theory*, 44(5):1897–1905, 1998. Presented in part 1995, see Springer LNCS. 1

[CE00a] Gérard D. Cohen and Sylvia B. Encheva. Efficient constructions of frameproof codes. *Electronic Letters*, 36(22), 2000. 4.1, 5.1

[CE00b] Gérard D. Cohen and Sylvia B. Encheva. Intersecting codes and partially identifying codes. Technical report, Ecole Nationale Supérieure des Télécommunications, September 2000. 1.1

[CE01] Gérard D. Cohen and Sylvia B. Encheva. Identifying codes for copyright protection. Technical report, Ecole Nationale Supérieure des Télécommunications, 2001. 2.2

[CEL01] Gérard D. Cohen, Sylvia B. Encheva, and Simon Litsyn. Intersecting codes and partially identifying codes. In Daniel Augot, editor, *Workshop on Coding and Cryptography*, January 2001. 5.1

[CFN94] B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Advances in Cryptology - CRYPTO '94*, volume 839 of *Springer Lecture Notes in Computer Science*, pages 257–270. Springer-Verlag, 1994.

[CGL01] Fan Chung, Ronald Graham, and Tom Leighton. Guessing secrets. *Electron. J. Combin.*, 8, 2001. 5.1

[Cha90] I. M. Chakravarti. Families of codes with few distinct weights from singular and non-singular hermitian varieties and quadrics in projective geometries and hadamard difference sets and designs associated with two-weight codes. *Coding theory and design theory, Part 1*, pages 35–50, 1990. 2.3, 2.4

[CZ94] Gérard Cohen and Gilles Zémor. Intersecting codes and independent families. *IEEE Trans. Inform. Theory*, 40:1872–1881, 1994. 4.1, 4.2, 15, 3, 5.1

[DS98] Stefan Dodunekov and Juriaan Simonis. Codes and projective multisets. *Electron. J. Combin.*, 5(1), 1998. Research Paper 37. 6.1

[DSW00] Tran Van Trung D.R. Stinson and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *J. Stat. Planning and Inference*, 86(2):595–617, 2000. 1

[FGU69] A. D. Friedman, R. L. Graham, and J. D. Ulman. Universal single transition time asynchronous state assignments. *IEEE Trans. Comput.*, 18:541–547, 1969. 1

[HKM77] Tor Helleseth, Torleiv Kløve, and Johannes Mykkeltveit. The weight distribution of irreducible cyclic codes with block lengths $n_1((q^l - 1)/n)$. *Discrete Math.*, 18:179–211, 1977. 1.2

[HKY92] Tor Helleseth, Torleiv Kløve, and Øyvind Ytrehus. Generalized Hamming weights of linear codes. *IEEE Trans. Inform. Theory*, 38(3):1133–1140, 1992. 6.1

[HvLLT98] Henk D. L. Hollmann, Jack H. van Lint, Jean-Paul Linnartz, and Ludo M. G. M. Tolhuizen. On codes with the identifiable parent property. *J. Combin. Theory Ser. A*, 82(2):121–133, 1998. 3.2

[KM88] J. Körner and K. Marton. New bounds for perfect hashing via information theory. *European Journal of Combinatorics*, 9:523–530, 1988. 3.2

[Kör95] János Körner. On the extremal combinatorics of the Hamming space. *J. Combin. Theory Ser. A*, 71(1):112–126, 1995. 1.1, 5.1

[KS88] J. Körner and G. Simonyi. Separating partition systems and locally different sequences. *SIAM J. Discrete Math.*, 1:355–359, 1988. 1, 4.4, 4.4

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977. 4.3, 5.2

[NT78] T. Nanya and Y. Tohma. On universal single transition time asynchronous state assignments. *IEEE Trans. Comput.*, 27:781–782, 1978. 3

[Sag65] Yu. L. Sagalovich. A method for increasing the reliability of finite automata. *Problems of Information Transmition*, 1(2):27–35, 1965. 1

[Sag75] Yu. L. Sagalovich. *State Assignment and Reliability of Automata*. Svyaz, Moscow, 1975. In Russian. 1, 2.2

[Sag94] Yu. L. Sagalovich. Separating systems. *Problems of Information Transmition*, 30(2):105–123, 1994. 1, 1.1, 2.1, 2.2, 4.4, 5.1, 5.2

[Sch00] Hans Georg Schaathun. The weight hierarchy of product codes. *IEEE Trans. Inform. Theory*, 46(7):2648–2651, November 2000.

[Sch01] Hans Georg Schaathun. Upper bounds on weight hierarchies of extremal non-chain codes. *Discrete Math.*, 241(1–3):449–469, 2001.

[SSW00] J. N. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. Available at `http://www.cacr.math.uwaterloo.ca/~dstinson/.`, September 2000. 1.1

[SvTW00] D. R. Stinson, Tran van Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *J. Statist. Plann. Inference*, 86(2):595–617, 2000. Special issue in honor of Professor Ralph Stanton. 1.1

[SW98] D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discrete Math.*, 11(1):41–53 (electronic), 1998. 1, 1.1

[SWZ00] D. R. Stinson, R. Wei, and L. Zhu. New constructions for perfect hash families and related structures using combinatorial designs and codes. *J. Combin. Des.*, 8(3):189–200, 2000. 3, 3, 3.1

[Tsf91] Michael A. Tsfasman. Algebraic-geometric codes and asymptotic problems. *Discrete Appl. Math.*, 33(1-3):241–256, 1991. Applied algebra, algebraic algorithms, and error-correcting codes (Toulouse, 1989). 3.2

[TV95] Michael A. Tsfasman and Serge G. Vlăduţ. Geometric approach to higher weights. *IEEE Trans. Inform. Theory*, 41(6, part 1):1564–1588, 1995. Special issue on algebraic geometry codes. 6.1

[Wei91] Victor K. Wei. Generalized Hamming weights for linear codes. *IEEE Trans. Inform. Theory*, 37(5):1412–1418, 1991. 1.2