Information Retrieval with Varying Number of Input Clues

Ville Junnila[*]and **Tero Laihonen**
Department of Mathematics and Statistics
University of Turku, FI-20014 Turku, Finland
viljun@utu.fi and terolai@utu.fi

Abstract

Information retrieval in associative memories was studied in a recent paper by Yaakobi and Bruck (2012). Associations between memory entries give us the t-neighbourhood of an entry. In their model, an information unit is retrieved from the memory with the aid of input clues which are chosen from a reference set. In this paper, we consider the situation where the information unit is found unambiguously using the associated t-neighbourhoods of the input clues. A varying number of input clues is allowed, but a limit m_u on the maximum number of them is imposed. Of course, we would like m_u to be as small as possible. We consider the problem over the binary Hamming space $\mathbb{F}^n$ and focus on the minimum of m_u, denoted by $\nu(n;t)$. Using linear reference sets, we show that $\nu(n;2) \leq 5$ for any $n \geq 9$. We also give infinite families of reference sets which provide good bounds on $\nu(n;t)$ for $t = 3$. In addition, efficient methods are given to obtain bounds on $\nu(n;t)$ for any t from known reference sets.

We also discuss the applications of this model to the Levenshtein's sequence reconstruction problem and sensor network monitoring.

Keywords: Information retrieval; associative memory; Hamming space; shortened code; linear code

1 Introduction

In current memory systems, there exist two main problems. The first problem is how large amounts of information can be stored and the second one concerns how information can be efficiently retrieved from the memory systems. The recent technological development has provided quite satisfactory solutions to the first problem. One approach towards the second problem is the notion of so called associative memories, in which memory entries are associated to each other and information is retrieved according to these associations (unlike the random access memory RAM). Associative memories try to mimic the human memory, which is fundamentally associative. For various models and aspects of associative memory see, for example, [9, 5, 1]. In this paper, we will consider the

[*]Research supported by the Finnish Cultural Foundation.

approach introduced by Yaakobi and Bruck [12] which concentrates on a natural problem of information retrieval when the information is stored associatively.

The introduction is structured as follows. In Section 1.1, we first define two models for information retrieval in associative memories; namely, the model of [12] and its new sequential version, which is the main subject of the paper although we provide new results to both models (as summarized in Remark 32). Then, in Section 1.2, these two models are compared, in particular, focusing on the benefits of the new version over the existing one. In Section 1.3, we consider two applications, Levenshtein's sequences reconstruction problem and sensor network monitoring, which are closely connected to the models of information retrieval. Finally, in Section 1.4, previous results on the subject and the structure of the paper are briefly discussed.

1.1 The models

Let $G = (V, E)$ be a simple, undirected and connected graph. Each vertex of the graph G represents a memory entry, where an information unit is stored. Associations between information units provide the edge structure of G and we say that two information units are t-associated if they are within distance t from each other. Assume that we are trying to retrieve an information unit $x \in V$ from the memory. For the retrieval, we are given *input clues* (or *input vertices*) which are chosen from a given *reference set* $C \subseteq V$ and are t-associated with x. The maximum number of input clues is limited by a positive integer m_u. According to the set of input clues $U \subseteq C$, we obtain as an output a set of vertices (information units), denoted by $S_t(U)$, which are t-associated to all the input clues. Clearly, $x \in S_t(U)$. The maximum size of the output set (over any set of input clues of size at most m_u) is called the *uncertainty* of the memory system. In this paper, we focus on the case where the searched information unit is determined uniquely, in other words, the output set $S_t(U)$ contains only x. Moreover, for the efficient use of the memory, we assume that each vertex (or information unit) is an output for some set of input clues, that is, we have access to all information units. Furthermore, we assume that the input clues are given sequentially one after another and that we wait for new input clues until a unique output vertex x is obtained or we reach m_u, the maximum number of input clues allowed.

Let us next describe more precisely the concept explained above. Assuming that $x, y \in V$, the graphic distance between x and y, i.e., the number of edges in any shortest path between x and y, is denoted by $d(x, y)$. For a non-negative integer t, we say that two vertices $x \in V$ and $y \in V$ are *t-associated* if $d(x, y) \leq t$. The set of vertices t-associated to a vertex $x \in V$ is called the *ball of radius t centered at x* and is denoted by

$$B_t(x) = \{z \in V \mid d(x, z) \leq t\}.$$

Let C be a nonempty subset of V, i.e., a *code* in V. Elements of a code C are called *codewords*. The set of codewords belonging to C and t-associated to a vertex $x \in V$ is denoted by

$$I_t(C; x) = B_t(x) \cap C.$$

Moreover, if the underlying code C is known from the context, we write in short $I_t(C; x) = I_t(x)$. Given a nonempty subset $U \subseteq V$, the set of vertices

t-associated to all elements of U is denoted by

$$S_t(U) = \bigcap_{c \in U} B_t(c).$$

Furthermore, we define $S_t(U) = V$ if $U = \emptyset$.

Let a code C be the reference set of the associative memory from which the input clues are chosen. Recall that we consider a scenario where each vertex x is a unique output for some set of input clues. This requirement can be formulated as follows: for each vertex $x \in V$, we have $S_t(I_t(x)) = \{x\}$. Indeed, the set $I_t(x)$ consists of all the possible input clues t-associated to x and for the requirement to hold we need x to be the unique vertex t-associated to all of them. By this observation, we are ready to define the function $m_t(C; x)$ which gives the number of input clues needed for uniquely determining (if possible) the retrieved vertex x. If the vertex x cannot be uniquely determined, i.e., $S_t(I_t(x)) \neq \{x\}$, then we set $m_t(C; x) = \infty$, else we define $m_t(C; x)$ to be the minimum number k such that for any $U \subseteq I_t(C; x)$ with $|U| = k$ we have $S_t(U) = \{x\}$. If the reference set C and/or the radius t is known from the context, then we write in short $m_t(C; x) = m(C; x) = m(x)$. Notice that trivially $S_t(U) = \{x\}$ if $U \subseteq I_t(x)$ and $|U| \geq m(x)$. Now we are ready to add the requirement for the upper bound on the number of input clues m_u, and present the formal definition of sequential information retrieval in associative memories.

Definition 1. Let $G = (V, E)$ be a simple, undirected and connected graph and C be a code in V. Assume further that $t \geq 0$ and $m_u \geq 1$ are integers. We say that a pair (G, C) is a *sequential* (t, m_u)*-associative memory with reference set* C if for each $x \in V$ we have $m_t(C; x) \leq m_u$. A sequential (t, m_u)-associative memory is abbreviated as $\mathcal{SAM}_G(t, m_u)$. We also say that C *gives* an $\mathcal{SAM}_G(t, m_u)$ if C is its reference set.

The previous definition is formulated for the case where a unique information unit or vertex is outputted. However, assuming N is a positive integer, the definition can be straightforwardly generalized for the case where — instead of a unique vertex — a set of vertices of size at most N is outputted. In other words, we say that information units can be retrieved from the associative memory with a given *uncertainty* N. For the definition in this case, we first generalize the function $m_t(C; x)$ as follows. If $|S_t(I_t(x))| \leq N$, then we define $m_t(C, N; x)$ to be the minimum number k such that for any $U \subseteq I_t(C; x)$ with $|U| = k$ we have

$$|S_t(U)| \leq N, \tag{1}$$

else we set $m_t(C, N; x) = \infty$. Then we say that a pair (G, C) is a *sequential* (t, m_u, N)*-associative memory with reference set* C if for each $x \in V$ we have $m_t(C, N; x) \leq m_u$. We abbreviate this as $\mathcal{SAM}_G(t, m_u, N)$. If C is known from the context, we write $m_t(N; x)$.

Considering the model of associative memories, one of the natural questions is how many input clues are needed in order to determine the retrieved information with desired accuracy. The parameter m_u gives an upper bound on the number of input clues needed. In order to retrieve information quickly from associative memories, we wish to have as small m_u as possible for a given N. This implies the following definition.

Definition 2. If there exists a reference set $C \subseteq V$ giving an $\mathcal{SAM}_G(t, m_u, N)$ for some m_u, then we define $\nu(G; t, N)$ to be the smallest such m_u, else we set $\nu(G; t, N) = \infty$. Moreover, if $N = 1$, then we write in short $\nu(G; t, N) = \nu(G; t)$.

In this paper, one of the main objectives is to study the value $\nu(G; t)$. However, there are also other issues that can be optimized, for example, the size of the reference set giving an associative memory. In particular, if we have two distinct reference sets giving an associative memory with the same limit m_u, then it seems natural to prefer the smaller one of the reference sets.

Previously, a similar concept of information retrieval in associative memories, where a *fixed* number m of input clues are given for all $x \in V$ was studied in [6], [7] and [12]. The following definition for that case requires additionally to the previous discussion above that there has to be at least m input clues for each $x \in V$. In other words, it is required that $|I_t(x)| \geq m$ for all $x \in V$.

Definition 3. ([6], [12]) Let $G = (V, E)$ be a simple, undirected and connected graph and C be a code in V. Assume further that $t \geq 0$, $m \geq 1$ and $N \geq 1$ are integers. We say that a pair (G, C) is a (t, m, N)-*associative memory with the reference set* C if

(i) $|I_t(x)| \geq m$ for any $x \in V$ and

(ii) $|S_t(U)| \leq N$ for any subset $U \subseteq C$ of size $|U| = m$.

A (t, m, N)-associative memory can be shortened as $\mathcal{AM}_G(t, m, N)$. We also say that C *gives* an $\mathcal{AM}_G(t, m, N)$ if C is its reference set.

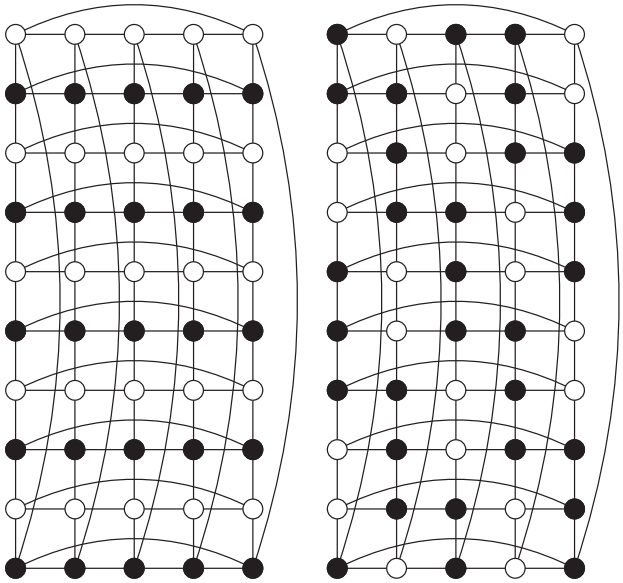
If $N = 1$, then we denote $\mathcal{AM}_G(t, m) = \mathcal{AM}_G(t, m, N)$. Notice that the condition (ii) of the previous definition can also be formulated as follows: for any $x \in V$, we have $m_t(C, N; x) \leq m$. It is immediate that if C is a reference set of an $\mathcal{AM}_G(t, m, N)$, then C also gives an $\mathcal{SAM}_G(t, m_u, N)$ with $m_u = m$, but not usually the other way around (see, for instance, Example 4). However, if C is a reference set of an $\mathcal{SAM}_G(t, m_u, N)$ and, in addition,

$$|I_t(C; x)| \geq m_u \qquad \forall x \in V, \tag{2}$$

then C also gives an $\mathcal{AM}_G(t, m, N)$ with $m = m_u$. In order to illustrate the previous definitions, we present the following two examples. In the first example, associative memories are considered when $N = 1$.

Example 4. In this example, we consider a toroidal grid graph $\mathcal{S}$ of height 10 and width 5. The graph is illustrated in Figure 1. In Figure 1(a), the shaded vertices form the code C_1. We show that C_1 gives an $\mathcal{SAM}_{\mathcal{S}}(1, 3)$ (with uncertainty $N = 1$). Let us count $m_1(x) = m_1(C_1; x)$ for each vertex x. We consider separately the cases $x \in C_1$ and $x \notin C_1$.

- If $x \in C_1$, then $m_1(x) = 3$. Indeed, $m_1(x) > 2$ because for two adjacent codewords $x, c \in I_1(C_1; x)$ we have $S_1(\{x, c\}) = \{x, c\}$ contradicting $N = 1$. On the other hand, choosing the set of all the three codewords $U = I_1(C_1; x)$ we have $S_1(U) = \{x\}$ which implies that $m_1(x) = 3$.
- If $x \notin C_1$, then choosing $U = I_1(C_1; x)$ gives $S_1(U) = \{x\}$ and hence $m_1(x) \leq 2$. Moreover, $m_1(x) > 1$, because any single input clue $c \in I_1(C_1; x)$ has $|S_1(\{c\})| = 5 > N = 1$. Thus, $m_1(x) = 2$.



(a) Code giving an $\mathcal{SAM}_{\mathcal{S}}(1,3)$ (b) Code giving an $\mathcal{AM}_{\mathcal{S}}(1,3)$

Figure 1: The reference sets of associative memories $\mathcal{SAM}_{\mathcal{S}}(1,3)$ and $\mathcal{AM}_{\mathcal{S}}(1,3)$ are formed by the shaded vertices.

All in all, we have $m_1(x) \leq m_u$ for $m_u = 3$, so C_1 gives an $\mathcal{SAM}_{\mathcal{S}}(1,3)$. However, since $|I_1(C_1;x)| = 2$ for any non-codeword x, the code C_1 does not give an $\mathcal{AM}_{\mathcal{S}}(1,3)$.

Let then the code C_2 be formed by the shaded vertices in Figure 1(b). It is straightforward to verify that for any x of $\mathcal{S}$ we have $|I_1(C_2;x)| = 3$ and $S_1(I_1(C_2;x)) = \{x\}$. Hence, the code C_2 gives an $\mathcal{AM}_{\mathcal{S}}(1,3)$.

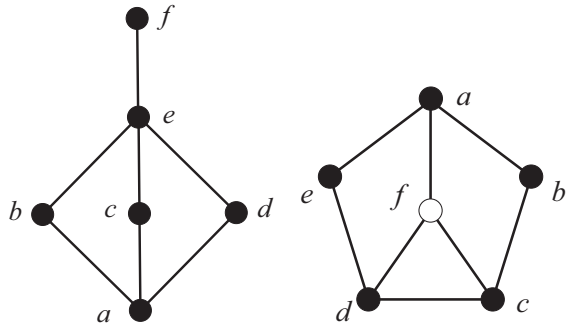
Observe that although $m_t(x) = |I_t(x)|$ for C_1 in the previous example, this is not usually the case. For example, for the infinite family of codes in the proof of Theorem 15 we have $m_3(x) \leq 7$ but $|I_3(x)|$ can be arbitrarily large.

In the second example, we consider the difference of sequential and non-sequential associative memories when the uncertainty $N = 2$.

Example 5. Let $G_1 = (V, E)$ be the graph of Figure 2(a). We show that $C = V$ gives an $\mathcal{SAM}_{G_1}(1,3,N)$ with uncertainty $N = 2$. We need to show that $m_1(C,2;x) = m_1(2;x) \leq 3$ for all $x \in V$.

Consider first $x = a$. We show first that $m_1(2;a) \leq 2$. In other words, for any $U \subseteq I_1(C;a)$ with $|U| = 2$, the inequality (1) is satisfied with $N = 2$. Let U be a subset of $I_1(C;a)$ with $|U| = 2$. Assume first that $a \in U$. If $U = \{a,b\}$, then $S_1(U) = \{a,b\}$ and hence $|S_1(U)| \leq N = 2$ as required in (1). Symmetrically, $|S_1(U)| \leq 2$ if $U = \{a,c\}$ or $U = \{a,d\}$. Suppose now that $a \notin U$. Then $S_1(U) = \{a,e\}$ for any such U, so again $|S_1(U)| \leq N = 2$. Consequently, $m_1(2;a) \leq 2$. Moreover, $m_1(2;a) = 2$ since $|S_1(\{a\})| = 4 > N = 2$.

Consider next $x = b$. Now $U = \{a,e\} \subseteq I_1(C;b)$ gives $S_1(U) = \{b,c,d\}$. Consequently, more clues $U \subseteq I_1(C;b)$ than two are needed to achieve $|S_1(U)| \leq N = 2$, and therefore, $m_1(2;b) \geq 3$. On the other hand, $|I_1(C;b)| = 3$ and $|S_1(I_1(C;b))| = 1 \leq N$, so $m_1(2;b) = 3$. Analogously, $m_1(2;x) = 3$ for $x \in$



(a) The graph G_1 (b) The graph G_2

Figure 2: The graph G_1 for uncertainty $N = 2$. The code giving an $\mathcal{AM}_{G_2}(1, 3)$ is formed by the shaded vertices.

$\{c, d\}$. Moreover, it is easy to check that $m_1(2; x) = 2$ for $x \in \{e, f\}$. Consequently, $C = V$ gives an $\mathcal{SAM}_{G_1}(1, 3, 2)$.

However, the code $C = V$ does not give a (non-sequential) $\mathcal{AM}_{G_1}(1, m, 2)$ for any m, because the condition (ii) of Definition 3 requires $m \geq 3$ due to the fact that $S_1(\{a, e\}) = \{b, c, d\}$, but the condition (i) does not allow this since $|I_1(V; f)| = 2 < m$.

Notice that when we deal with small fixed t (in a graph with relatively small cardinality of balls) and we get constant upper bound on m_u, it is not hard to find the sought information unit $x \in V$ after receiving the input clues U just by calculating $S_t(U)$ — we are dealing with the intersection of constant number of balls with small radius. However, for example, in the Hamming spaces we can find x in a very efficient way, see Section 3.2 and Remark 29.

1.2 Comparison of the models

In the previous section, we defined two models for associative memories; namely, the sequential and non-sequential ones. In order to motivate the study of the new version, we are going to compare these models focusing to the main benefits of the new model over the existing one in this section. These benefits are listed in the following and then discussed more closely in later examples, remarks and theorems.

- There are graphs in which sequential associative memories exist but not non-sequential ones. A family of such graphs is presented in Example 6.
- In sequential associative memories, we have an upper bound for the number of input clues needed for determining the sought information unit with desired accuracy. However, recall that for a given vertex x the maximum number of clues needed is $m_t(x) \leq m_u$. Hence, sometimes fewer clues than m_u is enough. Moreover, we do not even always need $m_t(x)$ clues for the retrieval as is discussed more closely in Remark 7. These features are obviously characteristic for the sequential model and do not apply to the non-sequential one, where all the m clues are received at once.

- There are situations where it is more natural to receive the input clues sequentially instead of all at once. As an example of this, see the problem of sensor network monitoring in Section 1.3.2.

- In Theorem 9 and Remark 10, we show that sequential associative memories have the property that new elements (possible clues) can be added to the reference set without causing major problems; the upper bound m_u may be increased in the process, but still the new reference set gives an $\mathcal{SAM}_G(t, m_u, N)$ for some m_u. However, in the non-sequential model, this is not the case as is shown in Example 8. The possibility of adding elements to the reference set is natural regarding the application of sensor network monitoring as discussed more closely in Section 1.3.2.

- Recall that the conditions of sequential associative memories is looser than the conditions of non-sequential ones. Hence, it is natural that the upper bound m_u for the number of clues needed in sequential memories may be smaller than the value m in the non-sequential case. This feature is illustrated in Example 31.

In the following example, we show that in a complete bipartite graph a sequential associative memory always exists but that this is not the case with the non-sequential one.

Example 6. Let $K_{s,h}$ be a complete bipartite graph with the independent sets U and V such that $|U| = s > 1$ and $|V| = h > 1$. Consider then the existence of $\mathcal{SAM}(1, m_u)$ and $\mathcal{AM}(1, m)$ in $K_{s,h}$ (now the uncertainty $N = 1$). Observe that if there exists a reference set C giving an $\mathcal{SAM}(1, m_u)$ or $\mathcal{AM}(1, m)$ (for some m and m_u), then for any vertex u we must have $S_1(I_1(u)) = \{u\}$. If there exists a vertex $u \notin C$ (say $u \in U$), then we have $S_1(I_1(u)) = U$, a contradiction. Therefore, we obtain that every vertex must belong to C.

Assume then that $s > h$ and the set $C = U \cup V$. It is straightforward to verify that C gives an $\mathcal{SAM}(1, s + 1)$. Considering the non-sequential case, we first observe that for the vertices $u \in U$ and $v \in V$ we have $I_1(u) = \{u\} \cup V$ and $I_1(v) = \{v\} \cup U$. Therefore, there exists no reference set giving an $\mathcal{AM}(1, m)$ (for any m) since $s \neq h$.

In the following remark, we show that in the situation of Example 4 the maximum number of input clues m_u (and not even $m_t(x)$) is not always needed to determine x.

Remark 7. Associative memory with sequential approach has the following advantage over the regular one of Definition 3. If we get the input clues one after another, then we can find an unknown information unit sometimes even earlier than after $m_t(x)$ clues. For example, in Figure 1(a), consider that we would like to find a codeword $c \in C_1$. There are three input clues available in $I_1(C_1; c)$, say c, c_1 and c_2. If we receive as input clues c_1 and c_2, then we immediately know that the information unit is c, so less clues are needed than $m_1(c) = 3$ (but if we receive as input clues c and c_1, we still do not uniquely know the sought information unit).

In the following example, we show that it is not always possible to add a vertex to the reference set giving a non-sequential associative memory.

Example 8. Let $G_2 = (V, E)$ be the graph of Figure 2(b). It is easy to check that $C = V \setminus \{f\}$ (the shaded vertices) gives an $\mathcal{AM}_{G_2}(1, 3)$ and, therefore, also gives an $\mathcal{SAM}_{G_2}(1, 3)$. Let us add the vertex f to the code (hence the new code equals V). The code V does not give an $\mathcal{AM}_{G_2}(1, m)$ for any m. Indeed, the condition (i) of Definition 3 gives $m \leq 3$, since $|I_1(V; b)| = 3$. On the other hand, the condition (ii) requires $m \geq 4$ because $|S_1(\{c, d, f\})| = |\{c, d, f\}| > N = 1$.

However, the new code V with an added codeword gives $\mathcal{SAM}_{G_2}(1, m_u)$ for $m_u = 4$. In general, one can add codewords to a code giving an $\mathcal{SAM}_G(t, m_u)$ and still get an $\mathcal{SAM}_G(t, m'_u)$ for $m'_u \geq m_u$, which becomes clear in Theorem 9.

1.3 Applications

In this section, we consider the connection of the above models to two previously studied problems; namely, Levenshtein's sequences reconstruction problem and locating objects in sensor networks.

1.3.1 Levenshtein's sequences reconstruction problem

Levenshtein's *sequences reconstruction problem* [10, 12] is motivated by questions arising in the fields like chemistry and biology, where the only way to overcome errors is to repeatedly transmit (say M times) the same codeword (no other method like redundancy is feasible). In other words, a codeword $x \in \mathbb{F}^n$, $\mathbb{F} = \{0, 1\}$, is transmitted through M channels where at most t errors can occur in each (see Figure 3). Based on the M different outputs $y_1, \ldots, y_M$ of the channels, a list decoder $\mathcal{D}_{\mathcal{L}}$ gives estimations $\{x_1, \ldots, x_\ell\}$ (where $\ell \leq \mathcal{L}$) on the transmitted word x. In [12], the minimum number of channels to guarantee the existence of a *successful* decoder (successful means that the transmitted codeword x belongs to $\{x_1, \ldots, x_\ell\}$) is studied. Let $(\mathbb{F}^n, C)$ be a sequential (t, m_u, N)-associative memory with reference set C. Then C provides a code for a successful decoder with $M = N + 1$ channels where N is the uncertainty and the parameter m_u gives an upper bound on the length of the list provided by the decoder, namely $\mathcal{L} < m_u$. Indeed, suppose we received the different words $y_1, \ldots, y_{N+1}$ from the $M = N + 1$ channels. The decoder outputs all the codewords $U = \{x_1, \ldots, x_k\}$ which are within distance t from all of the y_i's (all the codewords that could have been sent when these words are received). In other words, $U = S_t(\{y_1, \ldots, y_{N+1}\}) \cap C$ — notice that since distance is symmetric, we have $y_1, \ldots, y_{N+1} \in S_t(U)$. Clearly, $x \in U$, since there occurred at most t errors in each channel. Now we claim that the length $k = |U|$ of the outputted list satisfies

$$k < \min\{m_t(N; y_1), \ldots, m_t(N; y_{N+1}\}. \tag{3}$$

Suppose to the contrary that $k \geq m_t(N; y_i)$ for some $i = 1, \ldots, N + 1$. Now $U \subseteq I_t(y_i)$ with $k = |U| \geq m_t(N; y_i)$. Since C is a reference set with uncertainty N, we know by (1), that any $m_t(N; y_i)$ (or more) codewords $U \subseteq I_t(y_i)$ intersect in at most N words. Now the contradiction follows, because $|S_t(U)| \geq N + 1$. Hence the length of the list satisfies $k < m_u$.

In this paper we focus on $N = 1$, so we study situation where we need *only two* channels. Notice that there are advantages in using codes giving an $\mathcal{SAM}_{\mathbb{F}^n}(t, m_u)$ instead of the non-sequential ones. For instance, we show that sequential associative memories provide decoders with shorter output lists as pointed out in Example 31.

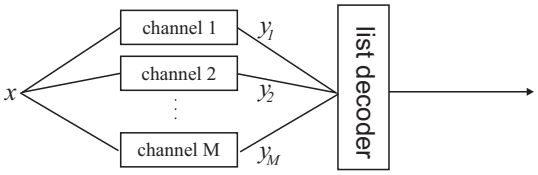


Figure 3: Channel model of the sequences reconstruction problem.

1.3.2 Sensor network monitoring

Let us look next at *sensor network monitoring* with RF-based localization proposed in [3, 11]. In particular, we discuss location detection by RF-sensors in indoor environments. Sensors in a building are mapped to vertices of a graph and a pair of vertices is connected by an edge if the two corresponding sensors are within each other's communication range (so the edge is bidirectional/undirected). A small portion of all sensors (those corresponding to codewords of $C \subseteq V$) are kept active while the others can be put in energy-saving mode. The system periodically (but not necessarily simultaneously) broadcasts ID packets from designated sensors. An observer should be able to determine her location $x \in V$ based on the ID packets that she receives from the sensors within radio range.

A code $C \subseteq V$ is called *t-identifying* [8] if the set $I_t(C; x)$ is nonempty for all $x \in V$ and

$$I_t(C; x) \neq I_t(C; y)$$

for all $x, y \in V$, $x \neq y$. If the code C corresponding to the active sensors is a t-identifying code, then the observer can determine her location x based on $I_t(C; x)$ (the received ID packets), because the set of received ID packets are distinct for all $x \in V$.

The reference sets C of Definition 1, can be used for the localization also since they provide a unique and nonempty $I_t(C; x)$ as will be shown in Theorem 9(ii). These codes have advantages over the regular identifying codes. For example, the parameter m_u bounds the number of input clues (ID packets from $I_t(C; x)$) needed to determine x. Another nice property is that the location x can be determined more efficiently. These features are considered more closely in Section 3.2.

Comparing reference sets coming from sequential and non-sequential associative memories, it seems that the non-sequential ones are more natural regarding identifying codes. Indeed, since the observer does not necessarily obtain the input clues (ID packets) all at once, it is natural to use a code of Definition 1 instead of Definition 3. Moreover, we are not particularly interested in the property that each location is covered by at least some fixed number of sensors. In this regard, it should be noted that allowing new active sensors to the network, i.e., new vertices to the reference set, is not a problem with sequential memories, but might be with the non-sequential ones of Definition 3 (see Example 8 and Theorem 9).

1.4 Previous works and structure of the paper

Previously, information retrieval in associative memories has been examined in [12] and [13], where the problem has been studied in the binary Hamming spaces and the Grassmann graphs, respectively. The problem in the binary Hamming spaces and in the infinite square grid has been further studied in [6]. The case where not all the entries are necessarily accessible is also considered in [6] — this means, that if we use only entries $V_1 \subseteq V$, then the requirement $m_t(x) \leq m_u$ concerns vertices x only in V_1. Associative memories in more general (undirected) graphs are investigated in [7].

In this paper, we will continue to study information retrieval in the binary Hamming space $\mathbb{F}^n$. We write $\nu(G;t) = \nu(n;t)$ for $G = \mathbb{F}^n$. The structure of the paper is as follows. In Section 2, we provide some basic structures and lower bounds on $\nu(n;t)$. Then, in Section 3, we consider bounds on $\nu(n;3)$ by giving two infinite families of reference sets. We also provide a shortening method to obtain results on $\nu(n;t)$ for any t from known reference sets. In Section 4, we provide two more such methods. Finally, in Section 5, we give optimal results for $\nu(n;t)$ when $t = 2$ for all $n \geq 3$ and the reference sets are linear codes.

2 Basics

We begin the section with the following useful theorem.

Theorem 9. *Let C be a code in $G = (V, E)$. Then the following conditions are equivalent:*

(i) For all vertices $x \in V$, the value $m_t(C;x)$ is finite.

(ii) For all distinct vertices $x, y \in V$, we have $I_t(C;x) \setminus I_t(C;y) \neq \emptyset$.

Moreover, if C satisfies the previous conditions, then C gives an $\mathcal{SAM}_G(t, m_u)$ with

$$m_u = \max_{\substack{x,y \in V \\ x \neq y}} \{|I_t(C;x) \cap I_t(C;y)|\} + 1. \tag{4}$$

Proof. Assume first that the condition (i) holds. Assume to the contrary that x and y are distinct vertices of V and $I_t(C;x) \setminus I_t(C;y) = \emptyset$. This implies that $I_t(C;x) \subseteq I_t(C;y)$ and hence $\{x, y\} \subseteq S_t(I_t(C;x))$. But now we have a contradiction with the fact that $m_t(C;x)$ is finite, i.e., $S_t(I_t(C;x)) = \{x\}$. Suppose then that the condition (ii) holds. If the condition (i) is not satisfied for some $x \in V$, then there exists a vertex $y \in V$ (different from x) such that $\{x, y\} \subseteq S_t(I_t(C;x))$. This implies that $I_t(C;x) \subseteq I_t(C;y)$, and a contradiction with the assumption follows.

Let C be a code satisfying the conditions and let the integer m_u be as given in (4). We show that $m_t(C;x) \leq m_u$ for any $x \in V$. If $|I_t(C;x)| \leq m_u$, then we are immediately done as the condition (i) holds for x giving $m_t(C;x) \leq |I_t(C;x)|$. Hence, we may assume that $|I_t(C;x)| > m_u$. Let then U be any subset of $I_t(C;x)$ with exactly m_u codewords. If there exists another vertex $y \in V$ such that $y \in S_t(U)$, then we have $U \subseteq I_t(C;x) \cap I_t(C;y)$ and a contradiction follows since $|I_t(C;x) \cap I_t(C;y)| \geq m_u$. Thus, we obtain that C gives an $\mathcal{SAM}_G(t, m_u)$. □

The condition (ii) is related to codes in [4]. The previous theorem immediately implies the following remark.

Remark 10. Let G be a graph and C a code giving an $\mathcal{SAM}_G(t, m_u)$. If C' is a code obtained from C by adding new codewords to the code, then we observe that C' does not violate the condition (ii) of the previous theorem. Hence, the code C' gives an $\mathcal{SAM}_G(t, m'_u)$ for $m'_u \geq m_u$ as discerned by (4).

In what follows, we first present some definitions and notation concerning binary Hamming spaces and then consider structural properties of reference sets in $\mathbb{F}^n$ and some general lower bounds on $\nu(n; t)$.

For the rest of the paper, let n be a positive integer. Denote the binary field by $\mathbb{F} = \{0, 1\}$. The binary Hamming space $\mathbb{F}^n$ is a graph with the vertex set $\mathbb{F}^n$. The vertices of $\mathbb{F}^n$ are called *words*. Let $x = x_1x_2 \cdots x_n$ and $y = y_1y_2 \cdots y_n$ be words of $\mathbb{F}^n$. There is an edge between x and y if they differ in exactly one coordinate. The *support* of a word x is defined as $\mathrm{supp}(x) = \{i \mid x_i = 1\}$ and the *weight* of x is $w(x) = |\mathrm{supp}(x)|$. The *Hamming distance* between the words x and y is $d(x, y) = w(x + y)$. For all $i = 1, 2, \ldots, n$, let $e_i \in \mathbb{F}^n$ denote the word for which $\mathrm{supp}(e_i) = \{i\}$. The all-zero word is denoted by $\mathbf{0} = 00 \cdots 0$ and the all-one word by $\mathbf{1} = 11 \cdots 1$. We also denote a sequential (t, m_u)-associative memory in a binary Hamming space $\mathbb{F}^n$ in short by $\mathcal{SAM}_n(t, m_u)$.

In the following theorem, we present some observations concerning the structure of reference sets. Recall that U is a set of input clues from $I_t(x)$ (the set of all possible input clues for x).

Theorem 11. *Let $t \geq 2$ be an integer. Assume that C is a reference set giving an $\mathcal{SAM}_n(t, m_u)$ for some positive integer m_u and let $x \in \mathbb{F}^n$. If $U \subseteq I_t(x)$ and $|U| \geq m_t(x)$, then there exist three distinct codewords $c_1, c_2, c_3 \in U$ such that $d(c_1, x) = d(c_2, x) = t$ and $t - 1 \leq d(c_3, x) \leq t$.*

Proof. Let U be a subset of $I_t(x)$ such that $|U| \geq m_t(x)$ (clearly $m_t(x) \leq m_u$ is finite). Let us begin with a simple observation:

- For any $s \in \{1, 2 \ldots, n\}$ there exists $c \in U$ such that $d(c, x) = t$ and $c \notin B_t(x + e_s)$. This can be seen as follows. Notice first that there exists a codeword $c \in U$ such that $c \in B_t(x) \setminus B_t(x + e_s)$. Indeed, otherwise $U \subseteq B_t(x) \cap B_t(x + e_s)$ and a contradiction follows since $\{x, x + e_s\} \subseteq S_t(U)$. Thus, as $B_{t-1}(x) \subseteq B_t(x + e_s)$, we have $d(c, x) = t$ and the observation follows.

This observation immediately gives us $c_1 \in U$ with $d(c_1, x) = t$ in the claim. Let i be an integer such that $i \in \mathrm{supp}(x + c_1)$. Then we obtain using the observation with $s = i$ that there exists a codeword $c_2 \in U$ such that $d(c_2, x) = t$ and $c_2 \notin B_t(x + e_i)$. Notice that c_1 is different from c_2 since $i \notin \mathrm{supp}(x + c_2)$. Let then j be an integer such that $j \in \mathrm{supp}(x + c_2)$. Clearly, the codewords c_1 and c_2 belong to $B_t(x + e_i + e_j)$. There exists a codeword $c_3 \in U$ such that $c_3 \in B_t(x) \setminus B_t(x + e_i + e_j)$ because otherwise $\{x, x + e_i + e_j\} \subseteq S_t(U)$. Therefore, as $B_{t-2}(x) \subseteq B_t(x + e_i + e_j)$, we have $t - 1 \leq d(c_3, x) \leq t$. Clearly, c_3 is different from c_1 and c_2. □

In the following theorem, we present lower bounds on $\nu(n; t)$ based on the previous theorem. Recall that $\nu(n; t)$ is the smallest of the upper bounds m_u on the number of input clues.

Theorem 12. *For all $t \geq 2$ we have $\nu(n;t) \geq 4$. Moreover, we have $\nu(n;t) \geq 5$ if $t \geq 4$. In addition, $\nu(n;1) = 3$ for all $n \geq 2$.*

Proof. Assume to the contrary that there exists a reference set C giving an $\mathcal{SAM}_n(t, m_u)$ with $m_u \leq 3$. Let then x be a codeword of C. Consider a subset $U \subseteq I_t(x)$ such that $|U| = m_t(x)$ and $x \in U$. As $|U| = m_t(x) \leq m_u \leq 3$, there exist at most two codewords of U such that their distance from x is $t-1$ or t. Thus, a contradiction with Theorem 11 follows.

Assume then that $t \geq 4$ and C gives an $\mathcal{SAM}_n(t, m_u)$. Let again x be a codeword of C. Without loss of generality, we may assume that $x = \mathbf{0}$. By Theorem 11, there exists a codeword $c \in C$ such that $d(c, x) = t$. Choose then $y \in \mathbb{F}^n$ to be a word such that $w(y) = 2$ and $\mathrm{supp}(y) \subseteq \mathrm{supp}(c)$. Since $t \geq 4$, we have $d(x, y) = 2 \leq t-2$ and $d(c, y) = t-2$, i.e., $\{c, x\} \subseteq B_{t-2}(y)$. Consider a subset $U \subseteq I_t(y)$ such that $|U| = m_t(y)$ and $\{c, x\} \subseteq U$. If $|U| = m_t(y) \leq 4$, then there exist at most two codewords of U such that their distance from y is $t-1$ or t. Hence, a contradiction with Theorem 11 follows. Thus, we have $\nu(n;t) \geq 5$.

Consider finally the case $t = 1$. Suppose that we have a reference set C such that $m_1(C;x) \leq 2$ for all $x \in \mathbb{F}^n$. Let $x \in C$. Clearly one input clue from $I_1(x)$ is not enough, so $m_1(x) = 2$. Let $U = \{x, c\} \subseteq I_1(x)$ where $x \neq c$. But now $\{x, c\} \subseteq S_1(U)$. Hence, $\nu(n;1) \geq 3$. As mentioned in the introduction, an $\mathcal{AM}_n(t, m)$ gives an $\mathcal{SAM}_n(t, m_u)$ with $m_u = m$, and from [6, Theorem 10] we can find an $\mathcal{AM}_n(1, 3)$ for every $n \geq 2$. □

For the upper bounds on $\nu(n;t)$ we refer to Theorem 15 and Theorem 27. The following lemma, which has been previously presented in [6, Lemma 13], is useful in constructing reference sets for associative memories.

Lemma 13 ([6])**.** *Let x be a word of $\mathbb{F}^n$ and t be an integer such that $t \geq 2$. Assume that words $c_1, c_2, c_3 \in \mathbb{F}^n$ are such that $d(c_1, x) = t$, $d(c_2, x) = t$, $t-1 \leq d(c_3, x) \leq t$ and the supports of $x + c_1$, $x + c_2$ and $x + c_3$ are pairwise disjoint. Then the balls $B_t(c_1)$, $B_t(c_2)$ and $B_t(c_3)$ intersect in a unique word x, i.e., $S_t(\{c_1, c_2, c_3\}) = \{x\}$.*

Remark 14. Notice that Lemma 13 gives a partial converse of Theorem 11, which reveals crucial underlying structure of the reference sets needed in the problem of associative memories. Namely, regarding each information unit $x \in \mathbb{F}^n$ we need to have in $I_t(x)$ (the set of all possible input clues for x) codewords positioned as mentioned in Theorem 11 — that is, there must be three suitable codewords at distance t or $t-1$. Our aim in the following is to construct codes (with the aid of Lemma 13) that would give the required kind of $I_t(x)$.

3 Construction and comparison to identification

The current section divides into the following two subsections. In Section 3.1, we give various constructions for reference sets of associative memories focusing to the case with the radius $t = 3$. We also present the so called shortening method for constructing references sets from known ones. In Section 3.2, we analyse some of the obtained reference sets more closely and discuss their benefits as identifying codes over the regular ones.

3.1 Constructions for $t = 3$ and the shortening method

In this section, we find upper bounds on $\nu(n; 3)$. As mentioned in Remark 14, we need codes such that for any $x \in \mathbb{F}^n$ there exist three suitable codewords at distance 3 or 2. For this, we use codes with careful interplay with minimum distance and covering radius (in Theorem 15 and Theorem 19 we use codes with minimum distance five and covering radius three to obtain $\nu(n; 3) \leq 7$). The codes we use as building blocks are nearly perfect or strongly uniformly packed codes [2, p. 313].

Recall that the *minimum distance* of a code $C \subseteq \mathbb{F}^n$ is defined as

$$d_{min}(C) = \min_{c_1, c_2 \in C, c_1 \neq c_2} d(c_1, c_2)$$

and the *covering radius* as

$$R(C) = \max_{x \in \mathbb{F}^n} \min_{c \in C} d(x, c).$$

We denote $d(x, C) = \min\{d(x, c) \mid c \in C\}$ for $x \in \mathbb{F}^n$.

Denote by $\mathcal{P}_r$ the punctured Preparata code [2, p. 51] of length $n = 2^{2r} - 1$. The key idea in the following proof is that the code $\mathcal{P}_r$ provides for the set $I_3(x)$ the structure of Lemma 13 for any $x \in \mathbb{F}^n$ with $d(x, \mathcal{P}_r) \geq 2$. However, since this is not true for x such that $d(x, \mathcal{P}_r) < 2$, we take care of these vertices by using two carefully chosen translates of $\mathcal{P}_r$. Clearly, these additional codewords, in turn, change the $I_3(x)$ for $d(x, \mathcal{P}_r) \geq 2$.

Theorem 15. *Let $r \geq 2$ be any integer. Then for the number of input clues we have the upper bound*

$$\nu(2^{2r} - 1; 3) \leq 7.$$

Proof. Let $\mathcal{P}_r$ be the punctured Preparata code $n = 2^{2r} - 1$, where $r \geq 2$, with covering radius three and minimum distance five. We will show that

$$C = \mathcal{P}_r \cup (e_1 + e_2 + \mathcal{P}_r) \cup (e_3 + e_4 + \mathcal{P}_r) \tag{5}$$

gives an $\mathcal{SAM}_n(t, m_u)$ with radius $t = 3$ and $m_u = 7$. We prove this by determining the parameter $m(x) = m_3(x)$ for all $x \in \mathbb{F}^n$. Since the covering radius $R(\mathcal{P}_r) = 3$ (the same is true for the subcodes $e_1 + e_2 + \mathcal{P}_r$ and $e_3 + e_4 + \mathcal{P}_r$), we do this by considering the following four cases depending on how far x is from each of the subcodes $\mathcal{P}_r$, $e_1 + e_2 + \mathcal{P}_r$ and $e_3 + e_4 + \mathcal{P}_r$ of C.

(i) Assume first that the word $x \in \mathbb{F}^n$ is at distance two or three from all of the three subcodes. In order to determine $m(x)$ we consider the structure of $I_3(C; x)$. The set $I_3(C; x)$ clearly consists of the codewords in $I_3(\mathcal{P}_r; x)$, $I_3(e_1 + e_2 + \mathcal{P}_r; x)$ and $I_3(e_3 + e_4 + \mathcal{P}_r; x)$. These sets are disjoint, because $d_{min}(\mathcal{P}_r) = 5$ implies that the subcodes are disjoint. Let us now examine the structure of $I_3(\mathcal{P}_r; x)$. We have the following facts:

- Since the minimum distance of $\mathcal{P}_r$ is five, no two codewords in $I_3(\mathcal{P}_r; x)$ differ from x in the same coordinate. In other words, if $c, c' \in I_3(\mathcal{P}_r; x)$ and $c \neq c'$, then $\operatorname{supp}(x + c) \cap \operatorname{supp}(x + c') = \emptyset$.
- There is at most one codeword in $I_3(\mathcal{P}_r; x)$ at distance two from x — again due to the minimum distance. (By the assumption of (i), $I_1(\mathcal{P}_r; x)$ is empty).

- Since $\mathcal{P}_r$ is a nearly perfect code [2, p. 313], the number of codewords at distance two or three from x is the maximal one, namely, $(2^{2r}-1)/3$.

The same facts hold, of course, for the sets $I_3(e_1+e_2+\mathcal{P}_r;x)$ and $I_3(e_3+e_4+\mathcal{P}_r;x)$ since the subcodes $e_1+e_2+\mathcal{P}_r$ and $e_3+e_4+\mathcal{P}_r$ have the same parameters as $\mathcal{P}_r$. By the last fact, we immediately have

$$|I_3(C;x)| = 2^{2r}-1. \tag{6}$$

If the set of input clues $U \subseteq I_3(C;x)$ contains at least three codewords from $I_3(\mathcal{P}_r;x)$ (or from either $I_3(e_1+e_2+\mathcal{P}_r;x)$ or $I_3(e_3+e_4+\mathcal{P}_r;x)$), then the first two facts above allows us to use Lemma 13 for $t=3$ and it implies that the intersection $S_3(U)=\{x\}$. On the other hand, there will inevitably be such three codewords in U (all from one of the sets) if $|U|\geq 7$, that is for any $U\subseteq I_3(C;x)$ with $|U|\geq 7$ we have $S_3(U)=\{x\}$. Consequently, $m(x)\leq 7$. (It would be easy to find U with $|U|=6$ such that $|S_3(U)|\geq 2$ showing that actually $m(x)=7$.)

(ii) Consider then the case that x is at distance two or three from exactly two of the subcodes, say C_1 and C_2, and at distance at most one from one of them, say C_3. Now what was said earlier holds for $I_3(C_1;x)$ and $I_3(C_2;x)$. The set $I_3(C_3;x)$, however, consists of a single codeword of C_3 due to the fact that the minimum distance of C_3 equals five. If we are given at least six input clues $|U|\geq 6$, $U\subseteq I_3(C;x)$, then among them there are at least three codewords from $I_3(C_1;x)$ (or analogously from $I_3(C_2;x)$). Consequently, Lemma 13 guarantees that the intersection $S_3(U)$ equals $\{x\}$. Hence, $m(x)\leq 6$. (Again we could show that $m(x)=6$.)

(iii) Let us then assume that x is at distance two or three from exactly one of the subsets, say C_1, and at distance at most one from the rest two, say C_2 and C_3. In this case, if we are given at least five codewords $U\subseteq I_3(C;x)$ as input clues, then x is the unique element in the intersection $S_3(U)$, since there must be at least three of them from $I_3(C_1;x)$. Consequently, $m(x)\leq 5$. (Again actually $m(x)=5$.)

(iv) Finally, we show that the distance of $x\in\mathbb{F}^n$ cannot be at most one to all of the subcodes (hence one of the previous three cases must occur). If $d(x,\mathcal{P}_r)=0$, then $d(x,e_1+e_2+x)=2$ and $e_1+e_2+x\in e_1+e_2+\mathcal{P}_r$. Moreover, $d(x,e_1+e_2+\mathcal{P}_r)=2$ due to the minimum distance $d_{min}(e_1+e_2+\mathcal{P}_r)=5$. So, it suffices to assume that $d(x,\mathcal{P}_r)=1$. Let $d(x,c)=1$ where $c\in\mathcal{P}_r$ and $x=e_i+c$. If $i>2$, then $d(x,e_1+e_2+c)=3$ where $e_1+e_2+c\in e_1+e_2+\mathcal{P}_r$ and also $d(x,e_1+e_2+\mathcal{P}_r)=3$. Finally, if $i=1$ or $i=2$, then $d(x,e_3+e_4+c)=3$ and so $d(x,e_3+e_4+\mathcal{P}_r)=3$. This completes the proof of the assertion. □

We will next utilize a shortening method of codes. We denote u consecutive zeros by 0^u. The *u-times shortened code* of $C\subseteq\mathbb{F}^n$ is defined via

$$\mathfrak{s}_u(C)=\{c_1c_2\ldots c_{n-u}\mid c_1c_2\ldots c_{n-u}0^u\in C\}.$$

That is, we choose only those codewords of C ending in u zeros and subsequently delete these zeros. Notice that here we do not require C to be linear and that we could choose other fixed ending $v\in\mathbb{F}^u$ than u zeroes of the codewords.

Theorem 16. *Let C be a code giving an $\mathcal{SAM}_n(t,m_u)$. If*

$$|I_t(\mathfrak{s}_u(C);x)|\geq m(C;x0^u) \tag{7}$$

for all $x \in \mathbb{F}^{n-u}$*, then* $m(\mathfrak{s}_u(C); x) \leq m(C; x0^u)$ *for all* $x \in \mathbb{F}^{n-u}$*. Thus* $\mathfrak{s}_u(C)$ *gives an* $\mathcal{SAM}_{n-u}(t, m_u)$*.*

Proof. Observe first that for all $x \in \mathbb{F}^{n-u}$

$$I_t(\mathfrak{s}_u(C); x) = \{c_1 c_2 \dots c_{n-u} \mid c_1 c_2 \dots c_{n-u} 0^u \in I_t(C; x0^u)\}. \tag{8}$$

Suppose next to the contrary that $m(\mathfrak{s}_u(C); x) > m(C; x0^u)$ for some $x \in \mathbb{F}^{n-u}$ (maybe $m(\mathfrak{s}_u(C); x) = \infty$). Consequently, we would have distinct words $x, y \in S_t(U)$ for some $U \subseteq I_t(\mathfrak{s}_u(C); x)$ with $|U| = m(C; x0^u)$ (this choice is possible, because $|I_t(\mathfrak{s}_u(C); x)| \geq m(C; x0^u)$). Denote $U = \{b_1, b_2, \dots, b_{|U|}\}$. Considering now the code C, the intersection

$$S_t(\{b_1 0^u, b_2 0^u, \dots, b_{|U|} 0^u\})$$

in $\mathbb{F}^n$ would contain two distinct words $x0^u$ and $y0^u$. However, this is impossible, because C gives an $\mathcal{SAM}_n(t, m_u)$ and $|U| = m(C; x0^u)$. This implies that $m(\mathfrak{s}_u(C); x) \leq m(C; x0^u)$ for all $x \in \mathbb{F}^{n-u}$.

□

With the aid of the shortening method we are able to extend the result of Theorem 15.

Corollary 17. *We have* $\nu(n; 3) \leq 7$ *for any* $n = 2^{2r} - 1 - u$ *where* $r \geq 2$ *and* $0 \leq u \leq (2^{2r} - 1)/3 - 5$*.*

Proof. Let C be the code defined in the proof of Theorem 15. We observed that there were three cases. Namely,

(i) if the distance of $x \in \mathbb{F}^n$ to all subcodes $\mathcal{P}_r$, $e_1 + e_2 + \mathcal{P}_r$ and $e_3 + e_4 + \mathcal{P}_r$ equals two or three, then the parameter $m(x) \leq 7$ and $|I_3(C; x)| = 2^{2r} - 1$.

(ii) if the distance of x equals two or three to exactly two of the subcodes and at most one to one subcode, then $m(x) \leq 6$ and $|I_3(C; x)| = 2(2^{2r} - 1)/3 + 1$.

(iii) if the distance of x equals two or three to exactly one of the subcodes, then $m(x) \leq 5$ and $|I_3(C; x)| = (2^{2r} - 1)/3 + 2$.

We prove the claim using Theorem 16 for suitable u. To that end, we need to estimate $|I_3(\mathfrak{s}_u(C); z)|$ for all $z \in \mathbb{F}^{n-u}$. By (8), it suffices to investigate how many codewords there are in $I_3(C; z0^u)$ ending in u zeros. Let C_1 be any of the three subcodes. If the distance of $z0^u$ is two or three to C_1, for each 0 among the last u coordinates there can be at most one codeword in $I_3(C_1; z0^u)$, which differs in that position (and gets thrown away in the shortening process). Consequently, if $x = z0^u$ is of type (i) above and we shorten u-times, then $|I_3(\mathfrak{s}_u(C); z)| \geq 2^{2r} - 1 - 3u$. Similarly, if $x = z0^u$ is of type (ii) (resp. type (iii)), then $|I_3(\mathfrak{s}_u(C); z)| \geq 2(2^{2r} - 1)/3 - 2u$ (resp. $|I_3(\mathfrak{s}_u(C); z)| \geq (2^{2r} - 1)/3 - u$). Notice that the codewords in the cases (ii) and (iii) within distance at most one from $x = z0^u$ can be removed in the shortening process in addition to the ones discussed earlier. If now u is at most $(2^{2r} - 1)/3 - 5$, then the assertion follows from Theorem 16. □

Example 18. Although the previous theorem does not cover the case $u = 1$ for $r = 2$, we can check by a computer that the 1-time shortened code $\mathfrak{s}_1(C)$, where $C = \mathcal{P}_2 \cup (e_1 + e_2 + \mathcal{P}_2) \cup (e_3 + e_4 + \mathcal{P}_2)$, gives an $\mathcal{SAM}_{14}(3, 7)$ (but not an $\mathcal{AM}_{14}(3, 7)$).

Theorem 19. *We have* $\nu(2^{2r+1}-1-u;3) \leq 7$ *for all* $r \geq 2$ *and* $0 \leq u \leq (2^{2r+1}-2)/6-5$.

Proof. We will use similar reasoning as in Theorem 15. Let $\mathcal{BCH}(2,r)$ be the primitive two-error-correcting BCH code [2, p. 48] of length $n = 2^{2r+1}-1$, $r \geq 2$. Denote further

$$C = \mathcal{BCH}(2,r) \cup (e_1 + e_2 + \mathcal{BCH}(2,r)) \cup (e_3 + e_4 + \mathcal{BCH}(2,r)).$$

This will give us an $\mathcal{SAM}_n(3,7)$. Notice that the covering radius of $\mathcal{BCH}(2,r)$ equals three and the minimum distance is five. Therefore, if the distance of $x \in \mathbb{F}^n$ equals two or three from $\mathcal{BCH}(2,r)$, then no two codewords of $I_3(\mathcal{BCH}(2,r);x)$ differ from x in the same coordinate. Now

$$|I_3(\mathcal{BCH}(2,r);x)| = (n-1)/6 \tag{9}$$

if the distance of x is two or three to $\mathcal{BCH}(2,r)$, because $\mathcal{BCH}(2,r)$ is strongly uniformly packed [2, p. 313]. The proof of Theorem 15 applies *mutatis mutandis* giving $m(x) \leq 7$ in the case (i), $m(x) \leq 6$ in (ii) and $m(x) \leq 5$ in (iii). Also the argument of (iv) goes analogously. We know now that C gives an $\mathcal{SAM}_n(3,7)$ for $n = 2^{2r+1}-1$. We can also apply the u-times shortening of Theorem 16 for $u \leq (n-1)/6-5$ since it remains true that for any coordinate with 0 in it there can be at most one codeword with 1 in that position in $I_3(C_1;x)$ for each subcode C_1 at distance two or three from x. □

3.2 Comparison with identifying codes

In this section, we discuss the advantage of using codes of Definition 1 instead of the usual identifying codes (see Section 1.3.2). We show the advantage concretely by comparing the usual 3-identifying codes to the infinite family of codes (5) in Theorem 15 giving an $\mathcal{SAM}_n(3,7)$.

Suppose first that C is a 3-identifying code in $\mathbb{F}^n$. In order to determine x, the observer has to have an access to the list of all the different 2^n sets $I_3(y)$, $y \in \mathbb{F}^n$, and then the observer compares the obtained set $I_3(x)$ (the ID packets from the active sensors) to these sets $I_3(y)$. As the observer goes through the list, she eventually finds a word $y \in \mathbb{F}^n$ for which $I_3(x) = I_3(y)$. Thus, she can determine that $x = y$. So, she will have to compare the obtained set to the 2^n sets (and even after the receiving the first ID packet $c \in I_3(x)$, there are $|B_3(c)| \sim n^3$ sets to compare $I_3(x)$ with).

Let now C be one of the codes in the infinite family of (5) giving an $\mathcal{SAM}_n(3,7)$. Let us see how we can determine x in this case. Next we will show that we can find x after receiving only seven ID packets (input clues) and doing n simple bitwise comparisons. Moreover, the observer neither has to have access to the list of the sets $I_3(y)$, $y \in \mathbb{F}^n$ nor to the list of codewords of C at all.

Indeed, we can use majority algorithm (see [10]) on each bit of the input clues to decide the bits of x. By the structure of C in (5) (see proof of Corollary 17), on each bit there can be at most three input clues where the clues differ from x. So, the majority (at least four input clues) have the same bit as x. For example,

if we received the seven ID packets (here $n = 15$)

$$\begin{array}{rl} & 000011000001111 \\ & 001010100001011 \\ & 001011001000011 \\ & 011111000101011 \\ & 110011000001111 \\ & 111010100001011 \\ & 111111000001111 \\ \hline x = & 011011000001011 \end{array}$$

then we can determine x easily as above.

Hence, the observer only has to do simple comparison on n coordinates (and not 2^n sets as for 3-identifying codes) in order to find x. She even does not have to store the whole $I_3(x)$, but use *any* seven elements obtained first (notice that although ID packets are sent periodically, they are not necessarily sent simultaneously). For another instance of this method, see Remark 29.

When considering t-identifying codes, one of the key aspects is the *cardinality* — one wishes to activate as few sensors as possible. So, how well does the infinite sequence of codes in (5) giving an $\mathcal{SAM}_n(3;7)$ do in that respect? Recall that here $n = 2^{2r}-1$, $r \geq 2$. It is well-known that the cardinality of any t-identifying code is at least $2^n/|B_t(x)| \sim 2^n/n^t$ (see [8]). For $n = 2^{2r}-1$ and $t = 3$, the cardinality of an 3-identifying code is at least of order $\sim 2^{2^{2r}-6r-1}$. The codes in (5) has cardinality of order $\sim 2^{2^{2r}-4r+2}$, because they have the size three times the punctured Preparata code (see [2, p. 313]). Hence the codes of (5) do well in comparison with the usual 3-identifying codes in this respect also.

4 Two methods for reference sets

In this section, we will give in addition to the shortening in Theorem 16, two methods to obtain new results for the associative memory from known ones. The first one changes the radius (while keeping the same length) and the second one gives higher length and (often) larger radius. The latter method gives us, for example, $\nu(2^k;3) \leq 10$ for all $k \geq 4$ in Theorem 23. Notice that in the previous section the length $n = 2^k$ is not covered.

We denote by $\bar{x}$ the *complement* of a word $x \in \mathbb{F}^n$, that is, $\bar{x} = \mathbf{1} + x$. In the following result, we utilize the fact that $B_{n-t-1}(x) = \mathbb{F}^n \setminus B_t(\bar{x})$ in the Hamming spaces. Therefore, results on the radius t yield immediately results on the radius $n-t-1$.

Theorem 20. *Let $0 \leq t \leq n-1$. If $C \subseteq \mathbb{F}^n$ gives an $\mathcal{SAM}_n(t, m_u)$, then it also gives an $\mathcal{SAM}_n(n-t-1, m'_u)$ where*

$$m'_u = |C| - \min_{\substack{x,y \in \mathbb{F}^n \\ x \neq y}} |I_t(x) \cup I_t(y)| + 1. \tag{10}$$

Proof. Let C be a code giving an $\mathcal{SAM}_n(t, m_u)$. We have the correspondence between radii t and $n-t-1$ by noticing that

$$I_{n-t-1}(x) = C \setminus I_t(\bar{x}), \quad \forall x \in \mathbb{F}^n.$$

In order to show that C gives an $\mathcal{SAM}_n(n-t-1, m'_u)$ for some finite m'_u we use Theorem 9. If we are prevented to have a finite m'_u, then $m_{n-t-1}(C;x) = \infty$ for some x. Consequently, by Theorem 9, for some distinct x and y we have $I_{n-t-1}(C;x) \setminus I_{n-t-1}(C;y) = \emptyset$. Thus, $I_{n-t-1}(x) \subseteq I_{n-t-1}(y)$. This implies, using our correspondence between the radii, that $I_t(\bar{y}) \subseteq I_t(\bar{x})$ and hence $m_t(C;\bar{y}) = \infty$, a contradiction. On the other hand, by (4), we obtain

$$\begin{aligned} m'_u &= \max_{\substack{x,y\in\mathbb{F}^n \\ x\neq y}} \{|I_{n-t-1}(x) \cap I_{n-t-1}(y)|\} + 1 \\ &= \max_{\substack{x,y\in\mathbb{F}^n \\ x\neq y}} \{|(C \setminus I_t(\bar{x})) \cap (C \setminus I_t(\bar{y}))|\} + 1 \\ &= \max_{\substack{x,y\in\mathbb{F}^n \\ x\neq y}} \{|C \setminus (I_t(\bar{x}) \cup I_t(\bar{y}))|\} + 1. \end{aligned}$$

This yields the claim for m'_u. □

Example 21. It can be checked that the code

$$C = \{0000, 0001, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1110, 1111\}$$

gives an $\mathcal{SAM}_4(1,3)$ with

$$\min_{\substack{x,y\in\mathbb{F}^4 \\ x\neq y}} |I_1(x) \cup I_1(y)| = 5.$$

The previous theorem shows that C gives also $\mathcal{SAM}_4(2,8)$, i.e., $\nu(4;2) \leq 8$. By computer, one can check that $\nu(4;2) \geq 8$. Hence $\nu(4;2) = 8$.

We define the *direct sum* of codes $C_1 \subseteq \mathbb{F}^n$ and $C_2 \subseteq \mathbb{F}^h$ by

$$C_1 \oplus C_2 = \{(c_1, c_2) \mid c_1 \in C_1, c_2 \in C_2\} \subseteq \mathbb{F}^{n+h}.$$

Suppose that $C_1 \subseteq \mathbb{F}^n$ gives an $\mathcal{SAM}_n(t_1, m_u)$ and $C_2 \subseteq \mathbb{F}^h$ gives an $\mathcal{SAM}_h(t_2, m'_u)$. The following theorem says that this method provides us longer codes giving an $\mathcal{SAM}_{n+h}(t_1+t_2, M_u)$ with larger radius. The first part of the theorem is general and the proof uses simply Theorem 9 where we consider words x and y in two parts for the code $C_1 \oplus C_2$ — one part corresponding to the length of the first code and the second part corresponding the second one. Thus, we can benefit from the properties of codes C_1 and C_2 to show that the condition (ii) of Theorem 9 holds. If we wish to maintain the same radius in the process, then we utilize the simple choice $C_2 = \mathbb{F}$ which itself gives an $\mathcal{SAM}_1(0,1)$. In this case we can give a proper (sometimes even optimal) estimate on the value of $m_t(x)$.

Theorem 22. *Let $C_1 \subseteq \mathbb{F}^n$ be a code giving an $\mathcal{SAM}_n(t_1, m_u)$.*

(i) If $C_2 \subseteq \mathbb{F}^h$ gives an $\mathcal{SAM}_h(t_2, m'_u)$, then $C_1 \oplus C_2$ gives an $\mathcal{SAM}_{n+h}(t_1+t_2, M_u)$ where

$$M_u = \max_{\substack{x,y\in\mathbb{F}^{n+h} \\ x\neq y}} |I_{t_1+t_2}(C_1 \oplus C_2; x) \cap I_{t_1+t_2}(C_1 \oplus C_2; y)| + 1.$$

(ii) The code $C_1 \oplus \mathbb{F}$ gives an $\mathcal{SAM}_{n+1}(t_1, M_u)$ where

$$M_u \leq \max\left\{ 2\max_{x\in\mathbb{F}^n} |I_{t_1-1}(C_1;x)| + 1, m_u + \max_{\substack{x,y\in\mathbb{F}^n \\ x\neq y}} |I_{t_1-1}(C_1;x) \cap I_{t_1-1}(C_1;y)| \right\}.$$

Proof. We show that the condition (ii) of Theorem 9 holds for $C = C_1 \oplus C_2$ and then the claim for M_u follows from (4). Let $x = (x_1, x_2) \in \mathbb{F}^{n+h}$, where $x_1 \in \mathbb{F}^n$ and $x_2 \in \mathbb{F}^h$. Denote similarly $y = (y_1, y_2)$ with $y_1 \in \mathbb{F}^n$ and $y_2 \in \mathbb{F}^h$. Let $x \neq y$. We show next that there exists a codeword $(c_1, c_2) \in C_1 \oplus C_2$ such that it belongs to $I_{t_1+t_2}(x)$ but not to $I_{t_1+t_2}(y)$. Consider the two cases:

- If $x_1 \neq y_1$ and $x_2 \neq y_2$, then choose $c_1 \in I_{t_1}(C_1; x_1) \setminus I_{t_1}(C_1; y_1)$ and $c_2 \in I_{t_2}(C_2; x_2) \setminus I_{t_2}(C_2; y_2)$. Notice that such a codeword $c_1 \in C_1$ exists by Theorem 9 since C_1 gives an $\mathcal{SAM}_n(t_1, m_u)$. Likewise $c_2 \in C_2$ exists. Now $(c_1, c_2) \in I_{t_1+t_2}(x) \setminus I_{t_1+t_2}(y)$.
- Assume now that $x_1 = y_1$ and hence $x_2 \neq y_2$ (the case $x_2 = y_2$ and $x_1 \neq y_1$ goes analogously). By Theorem 11 that there must be a codeword $c_1 \in I_{t_1}(C_1; x_1)$ such that $d(c_1, x_1) = t_1$. Choose any codeword $c_2 \in I_{t_2}(C_2; x_2) \setminus I_{t_2}(C_2; y_2)$. Then $(c_1, c_2) \in I_{t_1+t_2}(x) \setminus I_{t_1+t_2}(y)$.

Now we get the first claim (i) using Theorem 9(ii) and (4).

Let us now consider the claim (ii). Since $\mathbb{F}$ gives an $\mathcal{SAM}_1(0, 1)$, we know, by virtue of (i), that $D = C_1 \oplus \mathbb{F}$ gives an $\mathcal{SAM}_{n+1}(t_1, M_u)$ with some finite M_u. Now we estimate M_u using the properties of C_1. Let $x = (x_1, x_2) \in \mathbb{F}^{n+1}$ and $y = (y_1, y_2) \in \mathbb{F}^{n+1}$ be different words with $x_1, y_1 \in \mathbb{F}^n$ and $x_2, y_2 \in \mathbb{F}$.

Suppose first that $x_1 = y_1$. Consequently, $x_2 \neq y_2$. If $d(c, x_1) \geq t_1$ for $c \in C_1$, then neither $(c, 0)$ nor $(c, 1)$ of D belongs to $I_{t_1}(D; x) \cap I_{t_1}(D; y)$. On the other hand, if $d(c, x_1) \leq t_1 - 1$ for $c \in C_1$, then both $(c, 0)$ and $(c, 1)$ belong to the intersection. Consequently,

$$|I_{t_1}(D; x) \cap I_{t_1}(D; y)| \leq 2 \max_{x_1\in\mathbb{F}^n} |I_{t_1-1}(C_1; x_1)|. \tag{11}$$

Assume next that $x_1 \neq y_1$ and $x_2 = y_2$, say $x_2 = y_2 = 0$ (the other case is similar). If $d(c, x_1) \leq t_1 - 1$ and $d(c, y_1) \leq t_1 - 1$ for $c \in C_1$, then both $(c, 0)$ and $(c, 1)$ belong to $I_{t_1}(D; x) \cap I_{t_1}(D; y)$. If either $d(c, x_1) = t_1$ or $d(c, y_1) = t_1$ and the other distance is at most $t_1 - 1$, then $(c, 0)$ belongs to the intersection but $(c, 1)$ does not. The case $d(c, x_1) = t_1$ and $d(c, y_1) = t_1$ gives the same result. Therefore, since $|I_{t_1}(C_1; x_1) \cap I_{t_1}(C_1; y_1)| \leq m_u - 1$ (otherwise, $m_{t_1}(C_1; x_1) > m_u$), we obtain

$$|I_{t_1}(D; x) \cap I_{t_1}(D; y)| \leq (m_u - 1) + \max_{\substack{x_1,y_1\in\mathbb{F}^n \\ x_1\neq y_1}} |I_{t_1-1}(C_1; x_1) \cap I_{t_1-1}(C_1; y_1)|. \tag{12}$$

Assume finally that $x_1 \neq y_1$ and $x_2 \neq y_2$. This case goes like the previous one except that if $d(c, x_1) = t_1$ and $d(c, y_1) = t_1$, then neither $(c, 0)$ nor $(c, 1)$ belongs to $I_{t_1}(D; x) \cap I_{t_1}(D; y)$. Hence the upper bound (12) is valid also here.

The claim (ii) now follows combining (11) and (12) with M_u from (i). □

Corollary 23. *We have $\nu(2^k; 3) \leq 10$ for all $k \geq 4$.*

Proof. For $k \geq 4$ even, we apply Theorem 22(ii) to the codes

$$C = \mathcal{P}_r \cup (e_1 + e_2 + \mathcal{P}_r) \cup (e_3 + e_4 + \mathcal{P}_r)$$

discussed in the proof of Theorem 15. There it was shown that $m_u = 7$ for those codes. We need to consider the sets $I_2(C; x)$ for the upper bound on M_u. Since the minimum distance of the punctured Preparata code $\mathcal{P}_r$ equals five, then $|I_2(\mathcal{P}_r; x)| \leq 1$. Therefore, $|I_2(C; x)| \leq 3$. Trivially, $|I_2(C; x) \cap I_2(C; y)| \leq |I_2(C; x)| \leq 3$. This reasoning provides us with the sought bound $M_u \leq \max\{2 \cdot 3 + 1, 7 + 3\} = 10$.

When $k \geq 5$ is odd, the claim follows analogously using the codes given in the proof of Theorem 19. □

Notice that the upper bound on M_u given in Theorem 22(ii) can be attained. For example, for the code $C \oplus \mathbb{F}$ where $C = \mathcal{P}_2 \cup (e_1 + e_2 + \mathcal{P}_2) \cup (e_3 + e_4 + \mathcal{P}_2)$ the bound gives $M_u \leq 10$ as seen in the previous proof, but $C \oplus \mathbb{F}$ does not give an $\mathcal{SAM}_{16}(3, M_u)$ for any $M_u \leq 9$, as easily checked by a computer.

5 Linear reference sets

In this section, we consider sequential associative memories with linear reference sets, i.e., with reference sets that are linear codes in $\mathbb{F}^n$. Although such associative memories have some independent interest (see Section 1.3.1), we mostly use them to provide more structure to the reference sets and, thus, to enable construction of better associative memories. We begin the section by recalling some preliminary definitions and notation concerning linear codes as well as reformulate a basic lemma in the case of linear reference sets.

Let C be a code in $\mathbb{F}^n$. We say that C is a *linear code* in $\mathbb{F}^n$ if C is a subspace of $\mathbb{F}^n$. The *dimension* k of C is equal to the number of words in any basis of C. A linear code C of length n and with dimension k is called an $[n, k]$ code. There exists an $(n - k)$-by-n matrix H, which is called the *parity check matrix* of C, such that

$$C = \{x \in \mathbb{F}^n \mid Hx^T = \mathbf{0}\}$$

where x^T denotes the transpose of x. Denote the columns of H by h_i ($i = 1, 2, \ldots, n$), i.e., $H = (h_1|h_2|\cdots|h_n)$. Assuming $y = y_1 y_2 \ldots y_n$ is an arbitrary word of $\mathbb{F}^n$, we call $s = Hy^T$ the *syndrome* of y. Observe that the syndrome s of y can also be calculated by summing up the columns h_i of H for such i that $y_i = 1$. Let r be a nonnegative integer and z be a word of $\mathbb{F}^n$ such that $w(z) = r$. If we have $Hz^T = s$, then the word $y + z$ belongs to C as $H(y + z)^T = \mathbf{0}$ and $d(y, y + z) = r$.

We call a pair $(\mathbb{F}^n, C)$ a sequential $[t, m_u]$-associative memory with linear reference set C if the pair is a sequential (t, m_u)-associative memory with reference set C and C is a linear code in $\mathbb{F}^n$. For a linear code C, we also say that C gives an $\mathcal{SAM}_n[t, m_u]$, and denote $\nu(n; t) = \nu[n; t]$.

Using the previous terminology, Lemma 13 can be reformulated as follows.

Lemma 24. *Let C be a linear code in $\mathbb{F}^n$ and H be a parity check matrix of C. Let x be a word of $\mathbb{F}^n$, s be the syndrome of x and t be an integer such that $t \geq 2$. Assume that words $z_1, z_2, z_3 \in \mathbb{F}^n$ are such that $w(z_1) = w(z_2) = t$, $t - 1 \leq w(z_3) \leq t$, $s = Hz_1^T = Hz_2^T = Hz_3^T$ and the supports of z_1, z_2 and*

z_3 are pairwise disjoint. Then the balls $B_t(x+z_1)$, $B_t(x+z_2)$ and $B_t(x+z_3)$ intersect in a unique word x, i.e., $S_t(\{x+z_1, x+z_2, x+z_3\}) = \{x\}$.

Now we are ready to focus more closely on the construction of linear reference sets in various situations. First, in Section 5.1, we consider sequential associative memories with linear reference sets in the case $t = 2$. In fact, we show that for linear reference sets we have $\nu[n; 2] = 5$ for any $n \geq 9$. Then, in Section 5.2, we proceed by studying the non-sequential case with $t = 2$. Finally, in Section 5.3, we briefly consider the cases when $t = 3$ and $t = 4$.

5.1 Linear reference sets for $t = 2$

In what follows, we consider sequential associative memories with linear reference sets when $t = 2$. More precisely, we determine the exact values of $\nu[n; 2]$ for all n. In particular, we show that $\nu[n; 2] = 5$ for all $n \geq 9$. We first concentrate on the general result for $n \geq 9$ and then discuss the remaining cases with $n \leq 8$. It should be observed that previously we know by [6, Theorem 11] only the following; there exists an $\mathcal{AM}_n(2, 5)$ for $n = 2^s - 1$, $s \geq 3$, which yields $\nu(2^s - 1; 2) \leq 5$ for $s \geq 3$.

In the following theorem, we improve the result of Theorem 12 for linear reference sets when $t = 2$. Notice that this bound is optimal, as is seen in Theorem 27.

Theorem 25. *We have $\nu[n; 2] \geq 5$.*

Proof. Assume to the contrary that there exists a linear reference set C giving an $\mathcal{SAM}_n[2, m_u]$ with $m_u \leq 4$. Let then x be a codeword of C. Without loss of generality, we may assume that $x = \mathbf{0}$. Consider a subset $U \subseteq I_2(x)$ such that $|U| = m_2(x)$ and $x \in U$. By Theorem 11, there exist three distinct codewords $c_1, c_2, c_3 \in U$ such that $d(x, c_1) = d(x, c_2) = 2$ and $1 \leq d(x, c_3) \leq 2$. If the supports of c_1, c_2 and c_3 are not mutually disjoint, then there clearly exists a word y of weight two such that x, c_1, c_2 and c_3 belong to $B_2(y)$. This implies a contradiction with the assumption $m_u \leq 4$ as $\{x, y\} \subseteq S_2(\{x, c_1, c_2, c_3\})$. Hence, we may assume that $\mathrm{supp}(c_1)$, $\mathrm{supp}(c_2)$ and $\mathrm{supp}(c_3)$ are pairwise disjoint.

Assume then that e_{i_1}, e_{i_2}, e_{j_1} and e_{j_2} are the words of weight one such that $c_1 = e_{i_1} + e_{i_2}$ and $c_2 = e_{j_1} + e_{j_2}$. Since C is a linear code, the sum $c_1 + c_2 = e_{i_1} + e_{i_2} + e_{j_1} + e_{j_2}$ also belongs to C. Observe now that the words $e_{i_1}+e_{j_1}$ and $e_{i_2}+e_{j_2}$ (notice that these are not c_1 and c_2) are both at distance two from the codewords x, c_1, c_2 and $c_1 + c_2$. Therefore, we have $\{e_{i_1} + e_{j_1}, e_{i_2} + e_{j_2}\} \subseteq S_2(\{x, c_1, c_2, c_1 + c_2\})$, and the contradiction follows. Thus, we have $\nu[n; t] \geq 5$. □

In the following theorem, we show that if we have a parity check matrix satisfying the conditions of the theorem, then the code of such parity check matrix gives an $\mathcal{SAM}_n[2, 5]$. After the theorem, we give constructions for such parity check matrices for any $n \geq 9$. Notice also that the proof of the theorem is based on the delicate use of Lemma 22.

Theorem 26. *Let q be an integer such that $q \geq 3$. Assume that C is a linear code formed by the parity check matrix H such that H has q rows and n columns, does not contain all-zero word as a column and satisfies the following conditions:*

(i) There exist at least $2^{q-1}+2$ different columns in H.

(ii) There exist at least three distinct words $y_1, y_2, y_3 \in \mathbb{F}^q$ such that each y_i^T appears twice as a column of H.

(iii) No column of H appears three times in H.

Then C gives an $\mathcal{SAM}_n[2,5]$.

Proof. In what follows, we show that for each $x \in \mathbb{F}^n$ we have $S_2(I_2(C;x)) = \{x\}$, i.e., $m_2(C;x)$ is finite, and for all distinct $x, y \in \mathbb{F}^n$ we have $|I_2(C;x) \cap I_2(C;y)| \leq 4$. Then the claim follows by Theorem 9 and, by (4), we can choose $m_u = 5$. We first prove that $m_2(C;x)$ is finite for all $x \in \mathbb{F}^n$. This proof is based on Lemma 24.

Assume that x is a word of $\mathbb{F}^n$. The syndrome of x is $s = Hx^T \in \mathbb{F}^q$. Suppose first that $s = \mathbf{0}$, i.e., x belongs to C. By the condition (ii), there exists three distinct words $z_1, z_2, z_3 \in \mathbb{F}^n$ of weight two such that their supports ($\mathrm{supp}(z_i) = \{p, p'\}$ where $h_p = y_i^T$ and $h_{p'} = y_i^T$ in H) are disjoint and $Hz_i^T = \mathbf{0}$ ($i = 1, 2, 3$). Hence, the words z_1, z_2 and z_3 of weight two satisfy the conditions of Lemma 24 and the claim immediately follows. Thus, we may assume that $s \neq \mathbf{0}$.

Observe that for each $b \in \mathbb{F}^q$ ($b \neq \mathbf{0}, s$) there exists a unique $b' \in \mathbb{F}^q$ ($b' \neq \mathbf{0}, s$) such that $s = b + b'$. In other words, there exist $2^{q-1} - 1$ pairs of words $b \in \mathbb{F}^q$ and $b' \in \mathbb{F}^q$ such that $s = b + b'$. If the syndrome s appears as a column in H (say, $s = h_p$), then by the condition (i) there exist $2^{q-1} + 1$ distinct columns of H other than s. Then, by the pigeon hole principle, there exist distinct columns b_1, b_1', b_2 and b_2' of H such that $s = b_1 + b_1'$ and $s = b_2 + b_2'$. Let then z_1 be the word whose support corresponds to b_1 and b_1' and z_2 be the word corresponding to b_2 and b_2'. Thus $Hz_1^T = b_1 + b_1' = s$ and $Hz_2^T = b_2 + b_2' = s$. Let further z_3 be the word with $\mathrm{supp}(z_3) = \{p\}$, so $Hz_3^T = h_p = s$. Clearly, z_1, z_2 and z_3 satisfy the conditions of Lemma 24 and the claim follows. Hence, we may assume that the syndrome s does not appear as a column in H. By the condition (i), there exist $2^{q-1} + 2$ distinct columns and none of them is equal to s. By the pigeon hole principle, there exist distinct columns b_1, b_1', b_2, b_2', b_3 and b_3' of H such that $s = b_1 + b_1'$, $s = b_2 + b_2'$ and $s = b_3 + b_3'$. Let z_1, z_2 and z_3 be words of $\mathbb{F}^n$ such that $Hz_1^T = b_1 + b_1'$, $Hz_2^T = b_2 + b_2'$ and $Hz_3^T = b_3 + b_3'$. Clearly, z_1, z_2 and z_3 satisfy the conditions of Lemma 24 and the claim again follows. Thus, in conclusion, we have shown that $m_2(C;x)$ is finite, i.e., $S_2(I_2(C;x)) = \{x\}$.

For the second part of the proof, we first observe that $B_1(x)$ contains at most two codewords of C for any $x \in \mathbb{F}^n$. Indeed, by the condition (iii), the syndrome $s = Hx^T$ appears at most twice as a column of H and therefore the observation $|I_1(x)| \leq 2$ holds. In [6, Proof of Theorem 11], the following fact has been shown

- for any distinct words $x, y \in \mathbb{F}^n$ there exist $x', y' \in \mathbb{F}^n$ such that $B_2(x) \cap B_2(y) \subseteq B_1(x') \cup B_1(y')$.

Thus, if x and y are different words of $\mathbb{F}^n$, we obtain by this fact that $|I_2(C;x) \cap I_2(C;y)| = |B_2(x) \cap B_2(y) \cap C| \leq |(B_1(x') \cup B_1(y')) \cap C| = |I_1(x') \cup I_1(y')|$ for some $x', y' \in \mathbb{F}^n$. Our previous observation gives $|I_1(x')| \leq 2$ and $|I_1(y')| \leq 2$ for any $x', y' \in \mathbb{F}^n$ and this yields $|I_1(x') \cup I_1(y')| \leq 4$. Now applying $|I_2(C;x) \cap I_2(C;y)| \leq 4$ to (4) in Theorem 9 completes the proof of the theorem. □

In what follows, we construct parity check matrices satisfying the conditions of the previous theorem. For the construction, first denote the binary representation of i of length q by $b_q(i)$ when q is a positive integer and i is an integer such that $0 \leq i \leq 2^q - 1$. Let then q and n be integers such that $q \geq 3$ and $2^{q-1} + 5 \leq n \leq 2^q + 2$. Then the parity check matrix H is defined as follows:

$$H = \begin{pmatrix} b_q(1) & b_q(2) & \cdots & b_q(n-3) & b_q(1) & b_q(2) & b_q(3) \end{pmatrix}. \tag{13}$$

If n is an integer such that $2^q + 3 \leq n \leq 2^{q+1} - 2$, then we define H as follows:

$$H = \begin{pmatrix} b_q(1) & b_q(2) & \cdots & b_q(2^q-1) & b_q(1) & b_q(2) & \cdots & b_q(n-(2^q-1)) \end{pmatrix}.$$

In both cases, the parity check matrix H clearly satisfies the conditions of the previous theorem. Thus, in conclusion, if q and n are integers such that $q \geq 3$ and $2^{q-1}+5 \leq n \leq 2^{q+1}-2$, then the linear code C_{lin} formed by the parity check matrix H gives an $\mathcal{SAM}_n[2,5]$. Therefore, we obtain that for any $n \geq 9$ there exists a linear code giving an $\mathcal{SAM}_n[2,5]$. Hence, combined with Theorem 25, we obtain the following result.

Theorem 27. *If n is an integer such that $n \geq 9$, then we have $\nu[n;2] = 5$.*

For integers $n \leq 8$, the exact values of $\nu[n;2]$ are determined in the following example.

Example 28. By [6, Theorem8(i)], we know that $\nu(3;2) = 7$ and actually also $\nu[3;2] = 7$. It is easy to check by a computer that $\nu[4;2] = 9$. Furthermore, $\nu[5;2] = 7$ which is obtained using the code C_5 with parity check matrix $H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \end{pmatrix}$. A nonlinear code (applying the binary representation of integers)

$$C = \{b_5(i) \mid i \in \{0, 1, 6, 7, 11, 13, 22, 23, 24, 26, 29\}\} \tag{14}$$

gives $\nu(5;2) \leq 6$. We have $\nu[6;2] = 5$ using the code C_6 with parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Moreover, the code $C_6 \oplus \mathbb{F}$ gives $\nu[7;2] = 5$ and $C_6 \oplus \mathbb{F} \oplus \mathbb{F}$ gives $\nu[8;2] \leq 7$. With a computer, one can verify that $\nu[8;2] = 7$.

It should be noticed that $\nu[n;2]$ (and $\nu(n;2)$, see Example 21) can sometimes increase when n grows as is seen in the previous example.

In the following remark, we briefly discuss the difficulty of finding the sought information unit when a set of input clues is given.

Remark 29. In Section 3.2, we discussed determining the sought information unit using majority algorithm for code in (5) with seven input clues, when the radius is $t = 3$. We can reason analogously for the code C_{lin} with *five* input clues for the radius $t = 2$.

5.2 The non-sequential case with $t = 2$

In Theorem 26, we presented a result concerning parity check matrices forming linear codes giving an $\mathcal{SAM}_n[2,5]$. In the following remark, we reformulate the result for $\mathcal{AM}_n(2,5)$.

Remark 30. Let H be a parity check matrix satisfying the conditions of Theorem 26 where we have replaced (i) by the (stronger) requirement

- (i') there exist at least $2^{q-1}+4$ different columns in H

and (ii) by

- (ii') there exist at least four distinct words $y_1, y_2, y_3, y_4 \in \mathbb{F}^q$ such that each y_i^T appears twice as a column of H.

Let C be the linear code formed by H. By Theorem 26, this trivially implies that C gives an $\mathcal{SAM}_n[2,5]$, $n \geq 12$. Moreover, using similar arguments as in the proof of Theorem 26, it can be shown that for each $x \in \mathbb{F}^n$ we have $|I_2(C;x)| \geq 5 = m_u$. Therefore, the linear code C also gives an $\mathcal{AM}_n(2,5)$, $n \geq 12$. Notice that analogous parity check matrices for C can be constructed as in the case of an $\mathcal{SAM}_n[2,5]$ above when $n \geq 12$.

In the following example, we compare sequential and non-sequential associative memories with linear reference sets in $\mathbb{F}^n$ for length $n = 9$.

Example 31. The linear code based on the parity check matrix in (13) gives a code with $m_u = 5$ for $n = 9$. The number of rows in the matrix (the codimension of the code) is three as $q = 3$. Using exhaustive search for parity check matrices, it is easy to show that the smallest m for which a code giving $\mathcal{AM}_9(2,m)$ (without *sequentiality*) exists is $m = 7$. This is obtained, for example, using the code with parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Hence, using sequential codes instead of non-sequential, we can have a *shorter output list* of the decoder $\mathcal{D}_\mathcal{L}$ discussed in Section 1.3.1. Comparing Theorem 26 and Remark 30, we also notice that in the case of non-sequential associative memory we require a larger n in order to achieve $m = 5$; more precisely, for sequential case $m_u = 5$ is obtained when $n = 9$ and for non-sequential case this is achieved when $n = 12$.

Moreover, since the length of the outputted list (3) for transmitted codeword $x \in C$ depends on the values $m_t(y)$ (instead of fixed m), the sequential codes can provide a shorter outputted list. For example, for the code C defined in (14) we know that $m_u \leq 6$, but as many as *half* of the words $y \in \mathbb{F}^5$ have $m_2(y) \in \{4,5\}$.

Remark 32. In this remark, we summarize those results of this paper, which also provide new results for the *original* model of Yaakobi and Bruck. As noticed in (2), if $|I_t(C;x)| \geq m_u$ for all $x \in \mathbb{F}^n$, then a code C giving an $\mathcal{SAM}_n(t,m_u)$ gives also an $\mathcal{AM}_n(t,m)$ (corresponding to the original model by Yaakobi and Bruck). Due to (6) and (9), the codes in the proofs of Theorem 15 and Theorem 19, provide new infinite families of codes giving also $\mathcal{AM}_n(3,7)$. For the similar observation on shortening method, see (7), but also Example 18 for the opposite. Recall also Remark 30 above for linear codes.

5.3 Linear reference sets for $t = 3$ and $t = 4$

In this section, we give some examples of an $\mathcal{SAM}_n[t, m_u]$ for $t = 3$ and $t = 4$. Let $\mathcal{H}_s$ denote the binary Hamming code of length $2^s - 1$, $s \geq 3$.

Example 33. We obtain $\nu[5; 3] \leq 13$ using the code C_5 of Example 28. The code C_6 of the same example gives $\nu[6; 3] \leq 11$. Moreover, we have $\nu[11, 3] \leq 6$ — for this, take the code $C \subseteq \mathbb{F}^{11}$ with parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

It is also easy to check that the Hamming code $\mathcal{H}_3$ gives $\nu[7; 3] \leq 7$, actually, with a computer one readily checks that $\nu[7; 3] = 7$. (However, longer Hamming codes are bad for our purposes since they would give a large m_u).

Let us now consider $t = 4$.

Theorem 34. *We have* $\nu[23; 4] \leq 9$.

Proof. Let $\mathcal{G}_{23}$ be the binary Golay $[23, 12, 7]$-code [2, p. 287]. Denote

$$C = \mathcal{G}_{23} \cup (e_1 + \mathcal{G}_{23}) \cup (e_2 + e_3 + e_4 + \mathcal{G}_{23}) \cup (e_1 + e_2 + e_3 + e_4 + \mathcal{G}_{23}).$$

Notice that C is linear. We claim that C gives an $\mathcal{SAM}_{23}[4, 9]$. We will do it by estimating $m(x) = m_4(x)$ for every $x \in \mathbb{F}^{23}$. Since $\mathcal{G}_{23}$ is a perfect code, its covering radius equals 3. Hence it is clear that any $x \in \mathbb{F}^{23}$ has distance at most three to all of the cosets of $\mathcal{G}_{23}$, $e_1+\mathcal{G}_{23}$, $e_2+e_3+e_4+\mathcal{G}_{23}$ and $e_1+e_2+c_3+e_4+\mathcal{G}_{23}$. Consider the following five cases.

(i) Let first x have distance exactly three to all of the cosets (including $\mathcal{G}_{23}$). Because $\mathcal{G}_{23}$ has minimum distance seven, the words in $I_4(\mathcal{G}_{23}; x)$ do not have a common coordinate where they differ from x. Moreover, $|I_3(\mathcal{G}_{23}; x)| = 1$. Since $\mathcal{G}_{23}$ is perfect, $|I_4(\mathcal{G}_{23}; x)| = 6$. The same is true for the other three subcodes. All the four cosets of $\mathcal{G}_{23}$ are disjoint. By Lemma 13, if we choose at least three codewords in $U \subseteq I_4(C_1; x)$ from any coset C_1, then the intersection $S_4(U) = \{x\}$. This situation is unavoidable, if we are given at least nine codewords U from $I_4(C; x)$. This yields $m(x) \leq 9$.

(ii) Suppose x has distance three to exactly three cosets and distance at most two to one, say C_1. Since the minimum distance of C_1 equals seven, we know that $|I_4(C_1; x)| = 1$. Now given eight codewords from $I_4(C; x)$ guarantees that three (or more) belong to a coset, which is not C_1, and they intersect uniquely in x. Hence, $m(x) \leq 8$.

(iii) If x has distance three to exactly two cosets and distance at most two to the other two cosets, then $m(x) \leq 7$.

(iv) Assume that x has distance three to exactly one coset and at most two to the rest of them. In this case, $m(x) \leq 6$.

(v) The word x cannot have distance two or less to all of the cosets as we will see next. If $d(x, c) = 3$ for some $c \in \mathcal{G}_{23}$, we are done, so let us consider the following cases.

- Assume first that $d(x,c)=2$ for some $c \in \mathcal{G}_{23}$ and denote $x = c + e_i + e_j$ with $i < j$. If $i \neq 1$, then $d(x, e_1 + c) = 3$ and $e_1 + c$ belongs to the coset $e_1 + \mathcal{G}_{23}$. If $i = 1$ and $j \in \{2,3,4\}$, then $d(x, e_2+e_3+e_4+c) = 3$ and we are done. Suppose next that $i = 1$ and $j > 4$. Now $d(x, e_1+e_2+e_3+e_4+c) = 4$ and, since the coset $e_1+e_2+e_3+e_4+\mathcal{G}_{23}$ is perfect, there exists a codeword $c' \in e_1 + e_2 + e_3 + e_4 + \mathcal{G}_{23}$ in it with $d(x,c') = 3$.

- Suppose $d(x,c) = 1$ for some $c \in \mathcal{G}_{23}$, say $x = c+e_i$. If $i \in \{1,2,3,4\}$, then $d(x, e_1 + e_2 + e_3 + e_4 + c) = 3$ and we are done. Assume then that $i > 4$. Then $d(x, e_2 + e_3 + e_4 + c) = 4$ and because the coset $e_2 + e_3 + e_4 + \mathcal{G}_{23}$ is perfect, we have $d(x,c') = 3$ for some $c' \in e_2 + e_3 + e_4 + \mathcal{G}_{23}$.

- If $x = c$ for some $c \in \mathcal{G}_{23}$, then $d(x, e_2 + e_3 + e_4 + c) = 3$ and we are done.

Summing up, $m(x) \leq 9$ for all $x \in \mathbb{F}^{23}$ and the desired result follows. □

6 Conclusion

In this paper, we consider codes for the basic problem of information retrieval from associative memory introduced by Yaakobi and Bruck [12]. In that problem, we should be able to find a stored information unit using input clues which are associated to it. The defined concept has also connections to Levenshtein's sequence reconstruction problem [10] and finding objects in sensor networks [3]. An associative memory mimics human memory and therefore it is natural to consider the situation where the input clues are received sequentially. We study the problem in the binary Hamming spaces $\mathbb{F}^n$. The main focus in this paper is on the maximum number of input clues $\nu(n;t)$ needed to find the sought information unit unambiguously. We provide upper bounds $\nu(n;2) \leq 5$ for all $n \geq 9$ and $\nu(n;3) \leq 7$ for infinitely many n. We also give methods (like shortening) regarding $\nu(n;t)$ for general t to get new codes from known ones.

For future work, it would be interesting to find code constructions giving good upper bounds (linear on n or even constant) on $\nu(n;t)$ for general fixed t. On the other hand, we believe that the lower bounds of Section 2 could be improved for larger (but fixed) radius t.

References

[1] P. Chou, The capacity of the Kanerva associative memory. *IEEE Trans. Inform. Theory 35(2): 281–298, 1989.*

[2] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering codes*, volume 54 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 1997.

[3] N. Fazlollahi, D. Starobinski and A. Trachtenberg. Connected identifying codes. *IEEE Trans. Inform. Theory*, 58(7): 4814–4824, 2012.

[4] I. Honkala and T. Laihonen. On a new class of identifying codes in graphs. *Inform. Process. Lett.*, 102(2-3):92–98, 2007.

[5] J.J. Hopfield Neural networks and physical systems with emergent collective computation abilities. *Proceedings of the National Academy of Science*, 79:2554-2558, 1982.

[6] V. Junnila and T. Laihonen. Codes for information retrieval with small uncertainty. *IEEE Trans. Inform. Theory*, 60(2):976–985, 2014.

[7] V. Junnila and T. Laihonen. Information retrieval with unambiguous output. *Information and Computation*, accepted for publication.

[8] M. G. Karpovsky, K. Chakrabarty, L. B. Levitin, On a new class of codes for identifying vertices in graphs, IEEE Trans. Inform. Theory, 44: 599–611, 1998.

[9] T. Kohonen. Self-organization and associative memory. Springer Series in Information Sciences, Vol.8. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[10] V. Levenshtein. Efficient reconstruction of sequences. *IEEE Trans. Inform. Theory*, 47(1): 2—22, 2001.

[11] S. Ray, D. Starobinski, A. Trachtenberg and R. Ungrangsi. Robust location detection with sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6): 1016-1025, 2004.

[12] E. Yaakobi and J. Bruck. On the uncertainty of information retrieval in associative memories. In *Proceedings of 2012 IEEE International Symposium on Information Theory*, pages 106–110, 2012.

[13] E. Yaakobi, M. Schwartz, M. Langberg, and J. Bruck. Sequence reconstruction for grassmann graphs and permutations. In *Proceedings of 2013 IEEE International Symposium on Information Theory*, pages 874–878, 2013.