Compression for Fixed-Width Memories

Ori Rottenstreich, Amit Berman, Yuval Cassuto and Isaac Keslassy
Technion
{or@tx, bermanam@tx, ycassuto@ee, isaac@ee}.technion.ac.il

***Abstract*—To enable direct access to a memory word based on its index, memories make use of fixed-width arrays, in which a fixed number of bits is allocated for the representation of each data entry. In this paper we consider the problem of encoding data entries of two fields, drawn independently according to known and generally different distributions. Our goal is to find two prefix codes for the two fields, that jointly maximize the probability that the total length of an encoded data entry is within a fixed given width. We study this probability and develop upper and lower bounds. We also show how to find an optimal code for the second field given a fixed code for the first field.**

I. Introduction

Fixed-width memories are particularly appealing for networking applications. They enable a direct access to a memory word regardless of its index. This property is mandatory, for instance, in hash-based implementations of forwarding tables [1].

We consider the encoding of data entries with $d = 2$ fields, drawn independently according to two known distributions, within a bound of L bits corresponding to the fixed width of a memory word. We would like to find two prefix codes for the two fields, so as to maximize the probability that the total length of the two codewords in the encoding of a data entry is at most L bits. Unfortunately, we will show that popular techniques for data compression such as *Huffman coding* [2] are not necessarily optimal for this metric.

Consider for instance a data entry with two fields, such that the value of the first field is among four possible elements $\{a, b, c, d\}$, and the second is one of the two elements $\{x, y\}$. As summarized in Table I.(A)-(B), these possible values appear with different probabilities, and the values of the two fields are drawn independently. For instance, the first field has the value a w.p. (with probability) 0.4, and b w.p. 0.3. We encode the two fields using two prefix codes σ_1, σ_2. As shown in Table I.(C), the selection of these codes determines the obtained width for the $4 \cdot 2 = 8$ possible data entries. Here, the minimal obtained width of an encoded data entry is 2 bits, and the maximal width is 4 bits. If the allowed fixed-width is $L = 3$, only four of the data entries can be encoded successfully (presented with $\surd$), while four others will have to be stored in a different memory hierarchy and will result in a slower access time. Accordingly, we say that the obtained success probability for this encoding scheme $C_D = (\sigma_1, \sigma_2)$ is the sum of probabilities for the successfully-encoded entries, i.e.

$$P_{success}(L = 3, C_D) = 0.24 + 0.16 + 0.09 + 0.06 = 0.55.$$

(A) First field with code σ_1

Element	Prob.	codeword
a	0.4	0
b	0.3	110
c	0.15	10
d	0.15	111

(B) Second field with code σ_2

Element	Prob.	codeword
x	0.6	0
y	0.4	1

(C) Possible entries encoded by the encoding scheme $C_D = (\sigma_1, \sigma_2)$

Entry	Prob.	Encoding	Width (ℓ)	$\ell \leq (L = 3)$
(a,x)	0.24	0 0	2	$\surd$
(a,y)	0.16	0 1	2	$\surd$
(b,x)	0.18	110 0	4	-
(b,y)	0.12	110 1	4	-
(c,x)	0.09	10 0	3	$\surd$
(c,y)	0.06	10 1	3	$\surd$
(d,x)	0.09	111 0	4	-
(d,y)	0.06	111 1	4	-

TABLE I
Example of Encoding Scheme. (A) and (B) illustrate the entry distribution of two fields, and their respective codes σ_1 and σ_2. (C) shows the resulting encoding scheme $C_D = (\sigma_1, \sigma_2)$, using the concatenation of the corresponding two codewords for the two fields. (Spaces are presented for simplicity and do not exist in practice.) In this example, four possible data entries are encoded successfully within $L = 3$ bits, and $P_{success}(L = 3, C_D) = 0.24 + 0.16 + 0.09 + 0.06 = 0.55$.

Recently, an encoding scheme that minimizes the maximal encoding width of a given set of data entries was suggested in [3]. Unfortunately, the fixed width of a memory array is typically predetermined by its manufacturer and cannot be changed dynamically by the user. Furthermore, the worst-case approach of [3] may be too stringent. Therefore, the new probabilistic model we consider here of maximizing the success probability for a fixed L is a more realistic one. We note that the occasional failures to meet the fixed width can easily be accommodated in practice by sending the data entry to a slower (e.g. DRAM) memory instead of the main (e.g. SRAM) memory. A high success probability will guarantee good access time on average.

The case of $d = 2$ fields is motivated by the popular L2 MAC tables that contain entries of two fields. The first field describes the Target Port, while the second is an aggregation of other attributes. Combining the distributions of the two fields into their product distribution is not feasible in networking applications, because it will require maintaining dictionaries of exorbitant sizes. The problem of fixed-width prefix encoding is related to the problem of compression with low probability of buffer overflow [4], [5]. However, to the best of our knowledge the case of multiple distributions is not covered by the prior work. Due to space limitations, we do not discuss in this study the more general case of more than $d = 2$ fields.

II. Model and Problem Formulation

A. Terminology

We start by describing the terminology we use throughout this study. For short, we refer to a data entry simply as an entry.

Definition 1 (Entry Distribution)**.** *An* entry distribution $D = ((S_1, P_1), (S_2, P_2)) = \Big((\{s_{1,1}, \ldots, s_{1,n_1}\},\ (p_{1,1}, \ldots, p_{1,n_1})),\ (\{s_{2,1}, \ldots, s_{2,n_2}\},\ (p_{2,1}, \ldots, p_{2,n_2}))\Big)$ *is characterized by two (ordered) sets of elements with their corresponding vectors of positive appearance probabilities. An entry* (a_1, a_2) *has two fields drawn randomly and independently according to the distribution* D *s.t.* $\Pr(a_1 = s_{1,i}) = p_{1,i}$ *and* $\Pr(a_2 = s_{2,i}) = p_{2,i}$ *with* $p_{1,i}, p_{2,i} > 0$. *The numbers of possible elements in the first and second field of an entry are* $n_1 = |S_1|$ *and* $n_2 = |S_2|$, *respectively.*

Example 1. *The entry distribution of the two fields illustrated in Table I, can be summarized as* $D = ((S_1, P_1), (S_2, P_2)) = \Big((\{a, b, c, d\}, (0.4, 0.3, 0.15, 0.15)), (\{x, y\}, (0.6, 0.4))\Big)$.

Definition 2 (Prefix Code)**.** *For a set of elements* S*, a code* σ *is an injective mapping* $\sigma : S \to B$*, where* B *is a set of binary codewords of size* $|B| = |S|$*. A code is called a* prefix code *if no binary codeword in* B *is a prefix (start) of any other binary codeword in* B*.*

Definition 3 (Encoding Scheme)**.** *An* encoding scheme C_D *of an entry distribution* D *is a pair of two prefix codes* $C_D = (\sigma_1, \sigma_2)$*. That is, each* σ_j *is a prefix code of the set of elements* S_j *in the first or the second field.*

Our main motivation for using prefix codes is the simplicity they enable in the representation of encoded entries in the memory. For each encoded entry, we simply keep the concatenation of the two codewords for each of its fields. It is easy to verify that the properties of the prefix codes (and indeed only of the first of them) guarantee that different entries yield different (concatenated) encodings.

For a binary string x, let $\ell(x)$ denote the length in bits of x. With an encoding scheme $C_D = (\sigma_1, \sigma_2)$, we say that the encoding width of an entry (a_1, a_2) is $\ell(\sigma_1(a_1)) + \ell(\sigma_2(a_2))$.

Next, we define the encoding width bound. With this bound, we can distinguish between different encoding schemes based on their obtained encoding widths for the possible entries.

Definition 4 (Encoding Width Bound)**.** *Given an encoding scheme* $C_D = (\sigma_1, \sigma_2)$ *and an encoding width bound of* L *bits, we say that an entry* (a_1, a_2) *is encoded successfully if its encoding width is not larger than the encoding width bound, i.e.* $\Big(\ell(\sigma_1(a_1)) + \ell(\sigma_2(a_2))\Big) \le L$.

B. Optimal Encoding Scheme for an Entry Distribution

We would like now to define the main problem that we address in this study. Given an entry distribution $D = ((S_1, P_1), (S_2, P_2))$ and an encoding width bound L, we would like to find an encoding scheme $C_D = (\sigma_1, \sigma_2)$ that maximizes the probability that an encoding of an arbitrary entry would be successful. For the scheme C_D, we denote this probability by $P_{success}(L, C_D)$.

We remind that we limit each of the two codes σ_1, σ_2 to be prefix. Therefore, the lengths of the binary codewords in each of the codes must satisfy Kraft's inequality, i.e. $\sum_{a \in S_j} 2^{-\ell(\sigma_j(a))} \le 1$ (for $j \in [1, 2]$). In addition, these lengths are clearly positive integers.

We can now express the problem as the following optimization problem. Here, $I(\cdot)$ is the indicator function that takes the value of 1 if the condition that it receives as a parameter is satisfied, and 0 otherwise.

$$\begin{aligned} \max \quad & P_{success} = \\ & \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} p_{1,i} \cdot p_{2,j} \cdot I\Bigg(\ell(\sigma_1(s_{1,i})) + \ell(\sigma_2(s_{2,j})) \le L\Bigg) \\ \text{s.t.} \quad & \sum_{a \in S_j} 2^{-\ell(\sigma_j(a))} \le 1, \forall j \in [1, 2] \quad \text{(1a)} \\ & \ell(\sigma_j(a)) > 0 \ \forall j \in [1, 2], \forall a \in S_j \quad \text{(1b)} \\ & \ell(\sigma_j(a)) \in \mathbb{Z} \ \forall j \in [1, 2], \forall a \in S_j. \quad \text{(1c)} \end{aligned}$$

Assuming an entry distribution $D = ((S_1, P_1), (S_2, P_2))$, we denote by $OPT(L)$ the optimal success probability, i.e. the maximal possible value of $P_{success}$ that can be obtained by any encoding scheme C_D as a function of a positive integer encoding width bound L. Formally,

$$OPT(L) = \max_{C_D = (\sigma_1, \sigma_2)} P_{success}(L, C_D). \quad (2)$$

We say that an encoding scheme C_D is *optimal* for a given L iff it satisfies $P_{success}(L, C_D) = OPT(L)$.

III. Observations

In this section, we suggest some basic observations regarding our problem. For the sake of simplicity, throughout the paper we assume that each field can hold the same number of $n = n_1 = n_2 = 2^W$ possible elements. We also assume that for $j \in [1, 2]$, the elements $S_j = \{s_{j,1}, \ldots, s_{j,n_j}\}$ are ordered in a non-increasing order of their probabilities such that $p_{j,i_1} \ge p_{j,i_2}$ if $i_1 < i_2$.

For any entry distribution $D = ((S_1, P_1), (S_2, P_2))$, the optimal success probability $OPT(L) \in [0, 1]$ is clearly a non-decreasing function of the parameter L. Let the encoding scheme $C^F = (\sigma_1^F, \sigma_2^F)$ be composed of two *fixed-length* codes, such that both encode each of the 2^W elements of S_1, S_2 by a codeword of W bits.

Property 1. *The optimal success probability satisfies*
(i) *For* $L \ge 2W$ $OPT(L) = 1$, *and for* $L < 2W$

$$OPT(L) \le 1 - \left(\sum_{i=2^{W-1}}^{2^W} p_{1,i}\right) \cdot \left(\sum_{i=2^{W-1}}^{2^W} p_{2,i}\right).$$

(ii) *If* $W \ge 2$, $OPT(L = 2) = p_{1,1} \cdot p_{2,1}$.

Proof: *(i)* $OPT(L) = 1$ for $L \geq 2W$ can be proved by the encoding scheme C^F, in which the obtained encoding width of any entry is clearly $W + W = 2W$. Thus $1 = P_{success}(L, C^F) \leq OPT(L)$. In addition, by Kraft's inequality, in any encoding scheme C_D, at least $(2^{W-1}+1)$ elements in S_1 and at least $(2^{W-1}+1)$ in S_2 are encoded with at least W bits. In the best case, these longer codewords will be assigned to the lowest-probability elements. If $L < 2W$ an entry composed of two such elements is not encoded successfully. This gives the second part of (i).

(ii) If $L = 2$, let's show that there is at most a single entry that can be encoded successfully. A legally encoded entry must be an entry composed of a pair of elements (one from S_1 and another from S_2), where each element is encoded within a single bit by σ_1 and σ_2. By Kraft's inequality, in both codes, there must be at most one such element when $W \geq 2$. The maximal success probability is obtained in an encoding scheme C_D that encodes in a single bit the most common elements in both sets $s_{1,1} \in S_1, s_{2,1} \in S_2$. Then, $OPT(2) = P_{success}(L, C_D) = p_{1,1} \cdot p_{2,1}$. ■

The encoding scheme C^F satisfies $P_{success}(L, C^F) = 0$ for $L < 2W$. Then, based on the proof of the last property we can immediately observe the following.

Property 2. *For $W \geq 2$, the encoding scheme C^F composed of two fixed-length codes with codewords of W bits is optimal for $L \geq 2W$, and is not optimal for $L \in [2, 2W-1]$.*

The next lemma suggests the intuitive result that we should prefer to use shorter codewords for elements that appear more often. It is also interesting because it reduces the search space for the optimal code.

Definition 5 (Monotone Coding)**.** *An encoding scheme $C_D = (\sigma_1, \sigma_2)$ of entry distribution $D = ((S_1, P_1), (S_2, P_2))$ is called* monotone *if for $j \in [1, 2]$, $i_1 < i_2$ implies that $\ell(\sigma_j(s_{j,i_1})) \leq \ell(\sigma_j(s_{j,i_2}))$.*

Lemma 1. *For any entry distribution $D = ((S_1, P_1), (S_2, P_2))$ and any $L \geq 1$, there exists a* monotone optimal *encoding scheme.*

Proof: We show how to build an optimal monotone encoding scheme based on any optimal encoding scheme $C_D = (\sigma_1, \sigma_2)$. Consider two arbitrary indices i_1, i_2 that satisfy $i_1 < i_2$. Then necessarily $p_{1,i_1} \geq p_{1,i_2}$. If $\ell(\sigma_1(s_{1,i_1})) > \ell(\sigma_1(s_{1,i_2}))$, we can replace σ_1 by a new code obtained by permuting the two codewords of s_{1,i_1}, s_{1,i_2}. With this change, an entry $(a_1 = s_{1,i_1}, a_2)$ is encoded successfully after the change only if the entry $(a_1 = s_{1,i_2}, a_2)$ was encoded successfully in C_D (and vice versa). Likewise, if before the change, the first of them was encoded successfully then also the second. Then, we can deduce that such a change cannot decrease $P_{success}$ and the result follows. We do the same for σ_2 and conclude. ■

We next prove that the success probability of encodings with short average code length can be bounded from below. This will be shown by a refinement of the Markov inequality. For $D = ((S_1, P_1), (S_2, P_2))$, $C_D = (\sigma_1, \sigma_2)$ let $\mathbb{E}(C_D) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} p_{1,i} \cdot p_{2,j} \cdot \Big(\ell(\sigma_1(s_{1,i})) + \ell(\sigma_2(s_{2,j}))\Big) = \sum_{i=1}^{n_1} p_{1,i} \cdot \ell(\sigma_1(s_{1,i})) + \sum_{j=1}^{n_2} p_{2,j} \cdot \ell(\sigma_2(s_{2,j}))$.

Property 3. *The encoding scheme $C_D = (\sigma_1, \sigma_2)$ for $D = ((S_1, P_1), (S_2, P_2))$ with an average encoding width of $\mathbb{E}(C_D)$ satisfies $P_{success}(L, C_D) \geq \frac{L+1-\mathbb{E}(C_D)}{L-1} = 1 - \frac{\mathbb{E}(C_D)-2}{L-1}$.*

Proof: An unsuccessfully encoded entry has a width of at least $(L+1)$ bits while the width of any encoded entry is at least 2 bits. We then have that $\mathbb{E}(C_D) \geq \Big((1 - P_{success}(L, C_D)) \cdot (L+1) + P_{success}(L, C_D) \cdot 2\Big)$. The result then follows. ■

We now consider a special option for the encoding scheme. The scheme $C_D^H = (\sigma_1, \sigma_2)$ is constructed s.t. σ_1 is a Huffman code of S_1 (with its probability vector P_1) and σ_2 is a Huffman code of S_2 [2]. A Huffman code minimizes the expected length of a codeword. Unfortunately, as shown in the next example, an encoding scheme C_D^H composed of two Huffman codes (for S_1, S_2) is not necessarily optimal.

Example 2. *As illustrated in Fig. 1 and Fig. 2 , let $n_1 = n_2 = 4 = 2^W$ for $W = 2$. Likewise, let $D = ((S_1, P_1), (S_2, P_2)) = \Big((\{s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}\},\ (p_{1,1} = 0.9, p_{1,2} = 0.06, p_{1,3} = 0.03, p_{1,4} = 0.01)),\ (\{s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}\},\ (p_{2,1} = 0.5, p_{2,2} = 0.2, p_{2,3} = 0.15, p_{2,4} = 0.15))\Big)$ and let the encoding width bound be $L = 3$.*

As shown in Fig. 1, let σ_1, σ_2 be Huffman codes of S_1, S_2 and let $C_D^H = (\sigma_1, \sigma_2)$ be the encoding scheme composed of these codes. Since $(p_{j,3} + p_{j,4}) < p_{j,1}$ and $p_{j,2} < p_{j,1}$ for $j \in [1, 2]$ then necessarily σ_j satisfies $\ell(\sigma_j(s_{j,1})) = 1, \ell(\sigma_j(s_{j,2})) = 2, \ell(\sigma_j(s_{j,3})) = 3, \ell(\sigma_j(s_{j,4})) = 3$. To calculate $P_{success}(L, C_D^H)$, we can see that there are 3 entries that are encoded successfully within $L = 3$ bits: $(s_{1,1}, s_{2,1})$, $(s_{1,1}, s_{2,2})$, $(s_{1,2}, s_{2,1})$. Accordingly, $P_{success}(L, C_D^H) = 0.9 \cdot 0.5 + 0.9 \cdot 0.2 + 0.06 \cdot 0.5 = 0.66$.

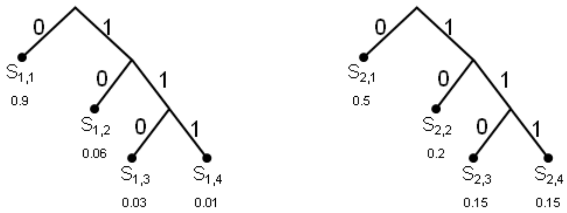
A second encoding scheme $C_D = (\sigma_1', \sigma_2')$ is presented in Fig. 2. This scheme satisfies $\ell(\sigma_1'(s_{1,1})) = 1, \ell(\sigma_1'(s_{1,2})) = 2, \ell(\sigma_1'(s_{1,3})) = 3, \ell(\sigma_1'(s_{1,4})) = 3$ (as in C_D^H) while $\ell(\sigma_2'(s_{2,1})) = \ell(\sigma_2'(s_{2,2})) = \ell(\sigma_2'(s_{2,3})) = \ell(\sigma_2'(s_{2,4})) = 2$. Here, 4 entries can be encoded successfully: $(s_{1,1}, s_{2,1})$, $(s_{1,1}, s_{2,2})$, $(s_{1,1}, s_{2,3})$, $(s_{1,1}, s_{2,4})$ and $P_{success}(L, C_D) = 0.9 \cdot 0.5 + 0.9 \cdot 0.2 + 0.9 \cdot 0.15 + 0.9 \cdot 0.15 = 0.9 > 0.66 = P_{success}(L, C_D^H)$.

Although the encoding scheme C_D^H is not necessarily optimal, by Property 3 it satisfies $P_{success}(L, C_D^H) \geq \frac{L+1-\mathbb{E}(C_D^H)}{L-1} = 1 - \frac{\mathbb{E}(C_D^H)-2}{L-1}$ where $\mathbb{E}(C_D^H)$ is the minimal possible average encoding width among all possible encoding schemes.

IV. Bounds on the Optimal Success Probability

In this section we present upper and lower bounds on the optimal success probability for a given entry distribution $D = ((S_1, P_1), (S_2, P_2))$ and an encoding width bound of L bits.

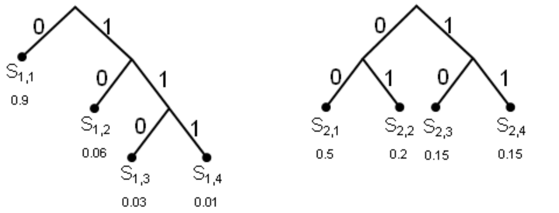
We start with a lower bound on $OPT(L)$. We consider several encoding schemes for D and have as a lower bound the maximal success probability achieved in one of these



(a) σ_1

(b) σ_2

Fig. 1. Illustration of the encoding scheme $C_D^H = (\sigma_1, \sigma_2)$ composed of two *Huffman* codes σ_1 of $S_1 = \{s_{1,1}, s_{1,2}, s_{1,3}, s_{1,4}\}$ (presented in (a) with probabilities (0.9,0.06,0.03,0.01)) and σ_2 of $S_2 = \{s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}\}$ (in (b) with probabilities (0.5,0.2,0.15,0.15)). For instance, $\sigma_1(s_{1,1}) = 0$ and $\sigma_2(s_{2,3}) = 110$. C_D^H satisfies $P_{success}(L = 3, C_D^H) = 0.66$.



(a) σ'_1

(b) σ'_2

Fig. 2. Illustration of a second encoding scheme $C_D = (\sigma'_1, \sigma'_2)$ with an improved success probability $P_{success}(L = 3, C_D) = 0.9$.

schemes. In each scheme, we divide the L bits between the two fields into ℓ and $(L - \ell)$ bits. Then, in all successfully encoded entries, the first field is encoded in ℓ bits while the second in $(L - \ell)$ bits. We consider values of ℓ such that $\ell, (L - \ell)$ are both positive and are not greater than W, i.e. $\ell \in [a, b] = [\max(1, L-W), \min(L-1, W)]$. The lower bound is presented in the following theorem.

Theorem 2. *The optimal success probability satisfies*

$$OPT(L) \geq \max_{\ell \in [a,b]} \left(\left(\sum_{i=1}^{2^\ell - 1} p_{1,i} \right) \cdot \left(\sum_{i=1}^{2^{L-\ell}-1} p_{2,i} \right) \right). \quad (3)$$

Proof: For a given ℓ, we consider a code σ_1 of S_1 in which the first $n_{1,1} = 2^\ell - 1$ elements (with larger probability) are encoded within ℓ bits while the last $n_{1,2} = n - n_{1,1}$ elements are encoded in $(\ell + \lceil \log_2(n_{1,2}) \rceil)$ bits. We can see that $n_{1,1} \cdot 2^{-\ell} + n_{1,2} \cdot 2^{-(\ell + \lceil \log_2(n_{1,2}) \rceil)} = (2^\ell - 1) \cdot 2^{-\ell} + n_{1,2} \cdot 2^{-(\ell + \lceil \log_2(n_{1,2}) \rceil)} \leq 1 - 2^{-\ell} + 2^{-\ell} = 1$, and as a consequence such a prefix code exists. We similarly encode all the first $n_{2,1} = 2^{(L-\ell)} - 1$ elements of S_2 in $(L - \ell)$ bits and the other $n_{2,2} = n - n_{2,1}$ elements in $((L-\ell) + \lceil \log_2(n_{2,2}) \rceil)$ bits. Then, any entry composed of two elements from the first $n_{1,1}, n_{2,1}$ elements in S_1, S_2 has an encoding width of $\ell + (L - \ell) = L$ bits and as a result is encoded successfully. ■

Next, we discuss an upper bound on $OPT(L)$. By Lemma 1 it is enough to show this only for monotone encoding schemes. We use the notations $\ell_{1,i} = \ell(\sigma_1(s_{1,i}))$, $\ell_{2,i} = \ell(\sigma_2(s_{2,i}))$ for $i \in [1, n]$ and have $\ell_{1,i} \leq \ell_{1,j}, \ell_{2,i} \leq \ell_{2,j}$ if $i < j$.

By Kraft's inequality, for $\ell \in [1, W - 1]$, at least one of the first 2^ℓ elements in S_1, S_2 is encoded with at least $\ell + 1$ bits, because there are additional elements that still need to be encoded. Based on this observation and the assumed order of the codeword lengths, we can deduce lower bounds on these lengths. For instance, if $W \geq 3$, we cannot encode the first two elements in a single bit, and one of the first four elements must be encoded with at least three bits. Thus $\ell_{j,2} \geq 2, \ell_{j,4} \geq 3$ for $j \in [1, 2]$ since we consider a monotone encoding scheme and accordingly $\ell_{j,i} \geq 2$ for $i \in [2, 3]$ and $\ell_{j,i} \geq 3$ for $i \in [4, 8]$. More generally, we can show that $\ell_{j,i} \geq \lceil \log_2(i + I(i < 2^W)) \rceil$ for $i \in [1, n = 2^W]$. Based on this lower bound, we denote by $f(\ell)$ for $\ell \in [1, W]$ the index of the first element that must be encoded in at least ℓ bits, i.e. $f(\ell) = \min_{i \in [1,n]} \left(\lceil \log_2(i + I(i < 2^W)) \rceil \geq \ell \right) = 2^{\ell - 1}$. Let $[a, b] = [\max(1, L + 1 - W), \min(L, W)]$. We are now ready to present the bound.

Theorem 3. *The optimal success probability satisfies*

$$OPT(L) \leq \min_{\ell \in [a,b]} \left(1 - \left(\sum_{i=2^{\ell-1}}^{2^W} p_{1,i} \right) \cdot \left(\sum_{i=2^{L-\ell}}^{2^W} p_{2,i} \right) \right).$$

Proof: For $\ell \in [1, L]$ that satisfies $\ell, (L + 1 - \ell) \leq W$, the encoding width of an entry composed of two elements encoded in at least ℓ and $(L + 1 - \ell)$ bits, respectively, is at least $\ell + (L + 1 - \ell) = L + 1$ bits. Therefore, any such entry is not encoded successfully and cannot contribute to the success probability of a monotone encoding scheme. By Lemma 1, we can generalize the result to any encoding scheme, and the result follows. ■

V. Optimal Conditional Code of the Second Field

The ultimate goal of this work is an algorithm that finds for a general value of L, an optimal two-field encoding scheme that jointly maximizes the success probability. An intermediate step toward this goal is an algorithm that finds the optimal encoding of one field, *conditioned* on the encoding of the other field. Such an algorithm is the topic of this section. This algorithm has a value in its own right (e.g., when one field encoding is set by external constraints), and also as a likely component in a future optimal two-field algorithm.

Finding an optimal encoding scheme when $L \geq 2W$ is easy. By Property 2, the encoding scheme C^F is an example for such a scheme. In this section we consider the case where $L \leq (2W - 1)$. Given a code $\sigma_1 = \sigma$ of S_1, we show a polynomial-time algorithm that finds an optimal conditional code σ_2 of the set S_2. This code σ_2 maximizes the probability $P_{success}(L, C_D = (\sigma_1, \sigma_2))$ for $\sigma_1 = \sigma$. Then, we also say that $C_D = (\sigma_1, \sigma_2)$ is an optimal conditional encoding scheme.

We first show the following lemma regarding the maximal length of a codeword in an optimal conditional code. Although the mentioned bound is not necessarily tight, it is used to limit the complexity of the suggested algorithm and its exact value is not required to show the algorithm correctness.

Lemma 4. *For an entry distribution $D = ((S_1, P_1), (S_2, P_2))$ with $W \geq 2$, an encoding width bound $L \in [2, 2W - 1]$ and a code $\sigma_1 = \sigma$ of S_1, there exists an optimal conditional code σ_2 of S_2 that encodes all of its n_2 elements in at most $3W$ bits.*

Proof Outline: Given an optimal conditional encoding scheme $C_D = (\sigma_1 = \sigma, \sigma_2)$, we consider elements of S_2 that obtain together with every element of S_1, a width longer than L. We replace σ_2 by a new code that encodes all these elements by $3W$ bits. We show that this new code still preserves Kraft's inequality and is also an optimal conditional code. ■

An optimal conditional code σ_2 with the property of Lemma 4 satisfies that $(\forall a \in S_2)(\ell(\sigma_2(a)) \leq 3W)$ and $2^{-\ell(\sigma_2(a))}$ is a multiple of 2^{-3W}. We define the weight of each codeword of length ℓ_0 as the number of units of 2^{-3W} in $2^{-\ell_0}$, denoted by $N_{\ell_0} = 2^{-\ell_0}/2^{-3W} = 2^{3W-\ell_0}$. Clearly, in order to satisfy Kraft's inequality, the sum of weights of the codewords of σ_2 should be at most $2^{3W} = n^3$.

We consider entries composed of an arbitrary first element from S_1 and a second element from the first k elements in S_2 (for $k \in [0, n]$). For $N \in [0, 2^{3W}]$ and $k \in [0, n]$, we denote by $F(N, k)$ the maximal sum of probabilities of such entries that can be encoded successfully by a code σ_2 such that the sum of its weights for the first k codewords is at most N. Formally,

$$F(N,k) = \max_{\sigma_2 : \left(\sum_{j=1}^{k} N_{\ell(\sigma_2(s_{2,j}))} \leq N\right)} \left(\sum_{i=1}^{n_1} \sum_{j=1}^{k} p_{1,i} \cdot p_{2,j} \cdot I\bigg(\ell(\sigma_1(s_{1,i})) + \ell(\sigma_2(s_{2,j})) \leq L \bigg) \right). \quad (4)$$

We would like now to present a recursive formula for $F(N, k)$. Earlier, we set the values of $F(N, k)$ for the initial case of $k = 0$ as $F(N, k = 0) = 0$ for $N \geq 0$ and $F(N, k = 0) = -\infty$ for $N < 0$. We can now present the formula of $F(N, k)$ that lets us calculate its values for $k = k_0$ based on the values of the function for $k = (k_0 - 1)$.

Lemma 5. *The function $F(N, k)$ satisfies for $N \geq 0, k \geq 1$*

$$F(N,k) = \max_{\ell_0 \in [1,3W]} \left(F(N - N_{\ell_0}, k-1) + p_{2,k} \cdot \sum_{i=1}^{n_1} p_{1,i} \cdot \mathrm{I}\bigg(\ell(\sigma_1(s_{1,i})) + \ell_0 \leq L \bigg) \right). \quad (5)$$

Proof: To calculate $F(N, k)$, we consider all possible lengths of the codeword of $s_{2,k}$ which is the k^{th} element in S_2. A codeword length of ℓ_0 reduces the available sum of weights for the first $(k-1)$ elements by $N_{\ell_0} = 2^{3W-\ell_0}$. Likewise, an entry $(s_{1,i}, s_{2,k})$ contributes to the success probability the value $p_{1,i} \cdot p_{2,k}$ if its encoding width (given ℓ_0) is at most L. ■

The following theorem relates the maximal success probability of a conditional encoding scheme and the function $F(N, k)$.

Theorem 6. *The maximal success probability of a conditional encoding scheme is given by*

$$\max_{\sigma_2} P_{success}(L, C_D = (\sigma_1 = \sigma, \sigma_2)) = F(N = 2^{3W}, k = n).$$

Proof: As indicated earlier, to satisfy Kraft's inequality we should limit the sum of weights N to $2^{3W} = n^3$. In addition, in the general case, the success probability is calculated based on entries that can include in the second field any one of the n elements of S_2. ■

Finally, we can present a dynamic-programming algorithm that finds the optimal conditional code. We denote by $Q(N, k)$ (again for $N \in [0, 2^{3W}]$, $k \in [0, n]$) a vector of length k that contains the codeword lengths for the first k elements in a code that achieves $F(N, k)$ and satisfies the constraint according to the meaning of N. We start by setting the values of $F(N, k)$ for $k = 0$ as described above. For these values, we also set $Q(N, k)$ as an empty vector.

Then, we perform n steps, and in the k^{th} step (for $k \in [1, n]$) we calculate $F(N, k)$ for the current value of k and $N \in [0, 2^{3W} = n^3]$. To calculate each value of $F(N, k)$, we rely on Lemma 5 and consider $3W$ possible lengths of the k^{th} codeword. If the maximal value is achieved when using the length of ℓ_0, we calculate $Q(N, k)$ by adding ℓ_0 (as an additional last element) to the vector $Q(N - N_{\ell_0}, k - 1)$. By Theorem 6 the optimal success probability is given by $F(N = (2^{3W} = n^3), k = n)$. Likewise, the codeword lengths of the optimal conditional code can be found in the vector $Q(N = (2^{3W} = n^3), k = n)$. Given the codeword lengths, we can easily find a code σ_2 that obtains these lengths.

The next property summarizes the time complexity of the suggested algorithm. The complexity is polynomial in the number of possible elements in each field n and the result follows directly from the description above.

Property 4. *The time complexity of the suggested dynamic-programming algorithm is*

$$O\Big(n \cdot (n^3 + 1) \cdot 3W\Big) = O\Big(n^4 \cdot \log(n)\Big). \quad (6)$$

VI. Conclusion

In this paper we studied efficient encoding schemes for fixed-width memories. We presented a new optimization problem and studied properties of the optimal obtained success probability. While we suggested an algorithm that finds an optimal code for a second field given the code of a first field, finding an optimal pair of codes for the two fields is left as an open question.

VII. Acknowledgment

We would like to thank Neri Merhav for fruitful discussions and pointers. This work was partly supported by the EU Marie Curie CIG grant, by the Intel ICRI-CI Center and by the Israel Ministry of Science and Technology. Ori Rottenstreich is the Google Europe Fellow in Computer Networking and an Andrew and Erna Finci Viterbi Fellow. Amit Berman is a HP Institute Fellow.

References

[1] A. Kirsch and M. Mitzenmacher, “The power of one move: Hashing schemes for hardware,” *IEEE/ACM Trans. Networking*, vol. 18, no. 6, pp. 1752–1765, 2010.

[2] D. Huffman, “A method for the construction of minimum redundancy codes,” *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[3] O. Rottenstreich, M. Radan, Y. Cassuto, I. Keslassy, C. Arad, T. Mizrahi, Y. Revah, and A. Hassidim, “Compressing forwarding tables,” in *IEEE Infocom*, 2013.

[4] F. Jelinek, “Buffer overflow in variable length coding of fixed rate sources,” *IEEE Trans. on Information Theory*, vol. 14, no. 3, pp. 490–501, 1968.

[5] P. A. Humblet, “Generalization of Huffman coding to minimize the probability of buffer overflow,” *IEEE Trans. on Information Theory*, vol. 27, no. 2, pp. 230–232, 1981.