Iterative Programming of Noisy Memory Cells

Michal Horovitz, *Member, IEEE*, Eitan Yaakobi, *Senior Member, IEEE*, Eyal En Gad, and Jehoshua Bruck, *Life Fellow, IEEE*

***Abstract*— In this paper, we study a model that mimics the programming operation of memory cells. This model was first introduced by Lastras-Montano *et al.* for continuous-alphabet channels, and later by Bunte and Lapidoth for discrete memoryless channels (DMC). Under this paradigm we assume that cells are programmed sequentially and individually. The programming process is modeled as transmission over a channel, such that it is possible to read the cell state in order to determine its programming success, and in case of programming failure, to reprogram the cell again. Reprogramming a cell can reduce the bit error rate, however this comes with the price of increasing the overall programming time and thereby affecting the writing speed of the memory. An *iterative programming scheme* is an algorithm which specifies the number of attempts to program each cell. Given the programming channel and constraints on the average and maximum number of attempts to program a cell, we study programming schemes which maximize the number of bits that can be reliably stored in the memory. We extend the results by Bunte and Lapidoth and study this problem when the programming channel is either discrete-input memoryless symmetric channel (including the BSC,BEC, BI-AWGN) or the Z channel. For the BSC and the BEC our analysis is also extended for the case where the error probabilities on consecutive writes are not necessarily the same. Lastly, we also study a related model which is motivated by the synthesis process of DNA molecules.**

***Index Terms*— Non-volatile memories, iterative programming, discrete-input memoryless symmetric channel, binary symmetric channel (BSC), binary erasure channel (BEC), Z channel.**

I. INTRODUCTION

MANY of the existing and the future volatile and non-volatile memories consist of memory cells. This includes for example STT-RAM, STT-MRAM, phase-change memories (PCM), flash memories, as well as strands of DNA molecules. The information in these memories is stored in cells that can store one or multiple bits. The state of each cell can be changed in several ways depending on the memory technology, such as changing its resistance or voltage level. The process of changing the cell state, which we call here *programming*, is crucial in the design of these memories as it determines the memory's characteristics such as speed, reliability, endurance, and more. Hence, optimizing the programming process has always been an important feature in the development of these memories.

Two of the more important goals when programming memory cells are speed and reliability. In this work we aim to understand the relation between these two figures of merits. Namely, we consider a model in which the cells are programmed sequentially, one after the other [4], [5]. Assume n binary cells are programmed. The cell programming process is modeled as transmission over some memoryless channel, for example the *binary symmetric channel* (*BSC*), the *binary erasure channel* (*BEC*), the *binary-input additive white Gaussian noise* (*BI-AWGN*) *channel*, or the Z *channel*. It is assumed that when a cell is programmed we can check the success of its programming operation and in case of failure we may choose to program it again. If there is no time restriction for programming the cells, an optimal solution is to program each cell until it reaches its correct value. For example, if the programming operation is modeled as the BSC with crossover probability p, then the expected number of programming attempts until reaching success is $1/(1-p)$. If $p = 0.1$, this increases the programming operation time by roughly 11%. However, if the system allows to increase the programming time by only 5%, then a different strategy should be considered.

More formally, we assume that there are n cells, for n sufficiently large, which are programmed according to some *iterative programming scheme* PS. We define the *average delay* of the programming scheme PS over a channel denoted by W as the expected number of programming attempts per cell, and the *maximum delay* is the maximum number of attempts to program a cell. Given some constraints, D and T, on the average and the maximum delay, respectively, our goal in this paper is to find a programming scheme that will maximize the number of information bits that can be reliably stored into the cells. Intuitively, the question is whether to spend time ensuring the cells are programmed correctly, or spend time and space for programming more redundancy cells in order to correct the errors. When programming a large number of cells one after the other we find the average delay to be the figure

Manuscript received February 12, 2021; revised September 15, 2021; accepted November 18, 2021. Date of publication November 25, 2021; date of current version February 17, 2022. Partial support was provided by NSF grants CCF-1717884 and CCF-1218005. An earlier version of this paper was presented in part at the IEEE Information Theory Workshop in August 2019 [DOI:10.1109/ITW44776.2019.8989404]. The associate editor coordinating the review of this article and approving it for publication was R. Venkataramanan. *(Corresponding author: Eitan Yaakobi.)*

Michal Horovitz is with the Department of Computer Science, Tel-Hai College, Upper Galilee 1220800, Israel, and also with the Galilee Research Institute—Migal, Kiryat Shmona, Upper Galilee 11016, Israel (e-mail: horovitzmic@telhai.ac.il).

Eitan Yaakobi is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

Eyal En Gad is with Micron Technology, San Jose, CA 95134 USA (e-mail: eengad@micron.com).

Jehoshua Bruck is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: bruck@caltech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3130660.

Digital Object Identifier 10.1109/TCOMM.2021.3130660

of merit that indicates the expected time to program all the cells.

We present this model as a way to solve the memory cell programming problem, however this is also a valid model for transmission processes, where there is noiseless feedback on the transmission success. That is, we consider the problem of transmitting bits, or more generally packets, over a channel with noiseless feedback. Then, in case of transmission error, the goal is to determine an optimal strategy which specifies whether to retransmit the bit again.

We note that there are some models where feedback does not increase capacity for DMC, but our model is different. In the feedback channel discussed in [6, Page 216], the receiver does not know if the sender is trying to send the same symbol again, or whether it is trying to transmit a new symbol, while in our model, when we want to write a symbol again, we overwrite it on the same memory cell. So there is a difference in the models, and that is why in our model the capacity increases with feedback. Additionally our model is different from the classical automatic repeat request (ARQ) [1], [7], [17] [21, Chapter 22] [26], [29], [31]–[33]. In an ARQ system an error detecting code is used, and if the receiver detects some errors then it requests the sender to retransmit the same data. This process continues till no error is detected. There exist also an hybrid-ARQ model, which combines the ARQ and the error-correcting codes, called, forward-error-control (FEC). For example, the ARQ protocol for the Z channel is discussed in [7], [17], [26], [31]–[33], where in [31], [32] the authors proposed capacity achieving codes for the Z channel using this model. In our model, a failure is detected without using a detecting error code, for example, the medium system recognizes a failure during the programming attempt. This is the main difference between our model and the ARQ protocol. By this fact, we can readily understand that the capacity of our model may exceed the capacity achieved by applying an ARQ model.

Previous works considered programming schemes mostly for flash memory cells. In [14], an optimal programming algorithm was presented to maximize the number of bits that can be stored in a single cell, which achieves the zero-error storage capacity under a noisy model. In [15], an algorithm was shown for optimizing the expected cell programming precision, when the programming noise follows a random distribution. In [36], algorithms for parallel programming of flash memory cells were studied which were then extended in [28] as well as for the rank modulation scheme in [27]. Other works studied the programming schemes with continuous-alphabet channels, see [9], [18]–[20], [22], [24], [25], [34] and references therein, where Mittelzholer *et al.* [25] introduced the problem of optimizing capacity with max and average iteration constraints.

Our point of departure in this paper is the programming model which was presented in [4], [5] by Bunte and Lapidoth for discrete alphabet memory channels (DMC). In particular, in [4] the case of symmetric channels with focus on the BSC was studied. We extend the results from [4] and study the problem for the BEC and the Z channel, where the last is applicable in particular for flash memories. We generalize the solution for all *discrete-input memoryless channels, which includes continuous channels, e.g. BI-AWGN.* Furthermore, we also study the case when the error probabilities on consecutive programming operations are not the same. Even though we follow the model from [4], we note that we propose a slightly different formulation to the problem and model, which we found to be more suitable to the cases solved in this paper.

Yet another model studied in this work is motivated by DNA-based storage systems. Recently, DNA has been explored as a possible near-future archival storage solution thanks to its potential high capacity and endurance [2], [3], [8], [12], [37]. DNA synthesis is the process of artificially creating DNA molecules such that arbitrary single stranded DNA sequences of length a few hundreds bases can be generated chemically. When synthesizing DNA strands, the bases are added one after the other to form the long sequence; for more details see [16]. However, this process is prone to errors which can be of the form of insertions, deletions, and substitutions. Since the bases are added in a sequential manner it is possible to check the success of each step and thereby to correct failures or repeat the attachment of the bases. In particular, in case the attachment of a specific base does not succeed on several consecutive iterations, it is possible to add another different base which indicates a synthesis failure in this location.

We briefly go over the applicability of the models proposed in this paper by reviewing their relevance to various types of memories. In STT-RAM and STT-MRAM, the information is encoded via the orientation of a magnetic element, where a bit flip is a common error. Thus, the BSC model, which was investigated by Bunte and Lapidoth [4], [5] and will be studied in Section III in this paper, is relevant for this model. The Z channel captures the main problem of encoding data into flash memories with single-level cells. This storage medium consists of cells which represent the data according to their charge level, and the main property is that charge can only be incremented with the programming iterations. Thus, the Z channel, which mimics such asymmetric errors, is suitable for flash memories. Note that flash memories have more properties which are not captured by the Z channel. This includes the connection between the different cells in the memory such as inter-cell interference and the need for block resetting for erasures. The fact that the charge level is incremented gradually is also not captured by the Z channel. The Z channel can be discussed also for PCM. In PCM, the data is encoded as a physical configuration of atoms, where the level of a PCM cell can only be incremented. Unlike flash, PCM does not have the constraint in which cells have to be erased together as blocks, and therefore the Z channel is somewhat more applicable for PCM. The combined BEC and BSC models, which will be studied in Section VII, have some elements which can model PCM and flash memories. These mediums can potentially have an erasure state by programming the memory cell to the highest level, which is relatively easier to do than to attempt to write intermediate levels.

The rest of the paper is organized as follows. In Section II, we formally present the definitions for the programming model and the problem studied in the paper. In Section III, we solve the programming model for the BSC and the BEC, and

In Section IV we extend the results for all the discrete-input memoryless symmetric channels, including continuous channels for example, the binary-input additive white Gaussian noise (BI-AWGN) channel. The Z channel is studied in Section V. In Section VI, we generalize this problem for the setup where consecutive programmings of a cell do not necessarily behave the same with respect to the error probability. A new model motivated by DNA, which combines the BSC and the BEC is studied in Section VII. Finally, Section VIII concludes the paper.

II. Definitions and Basic Properties

In this section we formally define the cell programming model and state the main problems studied in the paper. We also present some basic properties that will be useful in the rest of the paper.

Let W be a memoryless channel. We model the process of programming a cell as a transmission over a channel W, with the distinction that after every programming attempt, it is possible to check the cell state and to decide, in the case of an error, whether to leave the cell erroneous, or reprogram it again. We assume that there are n cells which are programmed individually. An *iterative programming scheme*, or in abbreviation *programming scheme*, is an algorithm which states the rules to program the n cells. Its *average delay* over channel W is defined to be the expected number of programming attempts per cell, where n, the number of cells, tends to infinity, and the *maximum delay* is the maximal number of attempts to program a cell. Our primarily goal in this work is to reliably store a large number of bits into the cells, while constraining the average and the maximum delay.

We define a natural class of programming schemes which are denoted by PS_t, for $t \geq 0$. The strategy of the programming scheme PS_t is to program each cell until its programming succeeds or the number of attempts is t, that is, after the t-th attempt the success is not verified and the cell may be left programmed erroneously. Applying PS_0 means that the cell is not programmed, while the programming scheme PS_∞ is the one where the cell is programmed until it stores the correct value. For notational purposes in the paper, we denote the programming scheme PS_∞ by PS_{-1}. Thus, for example, when we say $t \geq -1$, it includes the case of $t = \infty$.

For asymmetric channels, the average delay of PS_t may depend also on the content of the programmed words. For example in the Z channel the average delay depends on the number of zeros in the programmed word. Thus, for the rest of this section we refer only to symmetric channels,[1] while these concepts will be defined similarly in Section V for the Z channel.

For $t \geq -1$ and a symmetric channel W, we denote by $\mathcal{D}(W,t)$ the average delay of the programming scheme PS_t when the programming process is modeled by the channel W. For example (see Lemma 4), assume the channel is the *binary symmetric channel* (*BSC*) with crossover probability p, or the *binary erasure channel* (*BEC*) with erasure probability p, which will be denoted by $\mathrm{BSC}(p)$ and $\mathrm{BEC}(p)$, respectively. Then, $\mathcal{D}(W,t) = \frac{1-p^t}{1-p}$ for $t \geq 0$ and $\mathcal{D}(W,-1) = \frac{1}{1-p}$ (see Lemma 4), where W is the $\mathrm{BSC}(p)$ or the $\mathrm{BEC}(p)$. Unless stated otherwise, for the BEC we assume that $0 \leq p < 1$ and for the BSC, $0 \leq p \leq 1/2$.

When a cell is programmed according to a programming scheme PS_t, we can model this process as a parallel transmission over t copies of the channel W and there is an error if and only if there is an error in each of the t parallel channels. We denote this as a new channel W_t. Note that a programming scheme has no effect on the types of the errors, but it may change the probability of the cell to be in error. For example, if one cell is programmed using the programming scheme PS_{t_1} while another cell is programmed by the programming scheme PS_{t_2}, for $t_1 \neq t_2$, then the probabilities of these cells to be erroneous may be unequal. We denote the capacity of the channel W_t by $\mathcal{C}(W,t)$. For example, for $t \geq 1$ and $W = \mathrm{BSC}(p)$, $W_t = \mathrm{BSC}(p^t)$ and the capacity of the channel W_t is $\mathcal{C}(W,t) = 1 - h(p^t)$ where in this paper $h(x)$ is the binary entropy function in bits, where $0 \leq x \leq 1$. Note that for every channel W, it holds that $\mathcal{C}(W,0) = \mathcal{D}(W,0) = 0$ and $\mathcal{C}(W,-1) = 1$.

In this paper we focus on programming schemes that consist of combinations of several schemes from $\{PS_t\}_{t \geq -1}$. Let $\beta_1, \ldots \beta_\ell \in \mathbb{Q} \cap (0,1]$ where $\mathbb{Q}$ is the set of the rational numbers, and let $PS\,(\,(\beta_1,t_1),(\beta_2,t_2),\ldots,(\beta_\ell,t_\ell)\,)$ be a programming scheme of n cells which works as follows. For all $1 \leq i \leq \ell$, $\beta_i n$ of the cells are programmed according to the programming scheme PS_{t_i}.[2] Formally, for $T \geq 0$, where T indicates the maximum number of attempts to program a cell, we define the following set of programming schemes.

$$\mathcal{P}_T = \left\{ PS\left((\beta_1,t_1),(\beta_2,t_2),\ldots,(\beta_\ell,t_\ell)\right) : \beta_1,\ldots,\beta_\ell \in \mathbb{Q} \cap (0,1], 0 \leq t_1,\ldots,t_\ell \leq T, \sum_{i=1}^{\ell} \beta_i = 1 \right\}, \quad (1)$$

The set of programming schemes $\mathcal{P}_{-1}$ is defined similarly where $-1 \leq t_1,\ldots,t_\ell$.

$$\mathcal{P}_{-1} = \left\{ PS\left((\beta_1,t_1),(\beta_2,t_2),\ldots,(\beta_\ell,t_\ell)\right) : : \beta_1,\ldots,\beta_\ell \in \mathbb{Q} \cap (0,1], -1 \leq t_1,\ldots,t_\ell, \sum_{i=1}^{\ell} \beta_i = 1 \right\}.$$

For $T \geq -1$, it can be readily verified that for a programming scheme $PS = PS\left((\beta_1,t_1),\ldots,(\beta_\ell,t_\ell)\right) \in \mathcal{P}_T$ over a symmetric channel W, the average delay, denoted by

[1] We refer to the definition of symmetric DMCs as defined in [10] which are also called *partitioned symmetric* [11], where the set of the outputs can be partitioned into subsets in such a way that for each subset the matrix of transitions probabilities (using inputs as rows and outputs as columns) has the property that within each partition the rows are permutations of each other and the columns are permutations of each other. Additionally, we refer to continuous symmetric channels.

[2] We assume here and in the rest of the paper that n is sufficiently large so that $\beta_i n$ is an integer number for all i.

$\mathcal{D}(W, PS)$, is given by

$$\mathcal{D}(W, PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{D}(W, t_i).$$

Similarly, the *capacity*[3] of the programming scheme PS over the channel W is denoted by $\mathcal{C}(W, PS)$, and is naturally defined to be

$$\mathcal{C}(W, PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{C}(W, t_i),$$

where, as defined above, $\mathcal{C}(W, t_i)$ is the capacity of the channel W_{t_i}. Note that the definition of the capacity, $\mathcal{C}(W, PS)$, corresponds to the set of all achievable rates for reliably storing information in the cells. Specifically, when applying the programming scheme PS to program cells over the channel W, the following properties hold:

- for every $R < \mathcal{C}(W, PS)$, there exists a sequence of codes $\mathsf{C}_n = (2^{nR}, p_e^{(n)}, n)$, such that $p_e^{(n)} \to 0$ as $n \to \infty$,
- any sequence of codes $\mathsf{C}_n = (2^{nR}, p_e^{(n)}, n)$ such that $p_e^{(n)} \to 0$ as $n \to \infty$, must satisfy $R < \mathcal{C}(W, PS)$,

where C_n is a code of size 2^{nR}, n is the length of the codewords, and $p_e^{(n)}$ is the decoding error probability when using the code C_n.

The main problem we study in this paper is formulated in Problem 1 for symmetric channels. The motivation of this problem is to maximize the number of information bits that can be reliably stored in n cells when n is sufficiently large, where the average and maximum delays are bounded by D and T, respectively. The case of $T = -1$ corresponds to an unbounded maximum delay.

Problem 1: Given a symmetric channel W, an average delay D, and a maximum delay T, find a programming scheme, $PS \in \mathcal{P}_T$, which maximizes the capacity $\mathcal{C}(W, PS)$, under the constraint that $\mathcal{D}(W, PS) \le D$. In particular, given W, D, and T, find the value of

$$F_1(W, D, T) \triangleq \sup_{PS \in \mathcal{P}_T : \mathcal{D}(W,PS) \le D} \{\mathcal{C}(W, PS)\}.$$

Assume we are given a symmetric channel W, an average delay D, and a programming scheme $PS = PS\left((\beta_1, t_1), \ldots, (\beta_\ell, t_\ell)\right) \in \mathcal{P}_T$, such that $\mathcal{D}(W, PS) > D$. In order to meet the constraint of the average delay D by using the programming scheme PS, we program only, $\beta = \frac{D}{\mathcal{D}(W,PS)}$ fraction of the cells with the programming scheme PS, and the remaining $(1 - \beta)n$ cells are not programmed. Hence, we define the programming scheme $PS(W, D)$ as (2), shown at the bottom of the page. It can be readily verified that the properties in the next proposition hold.

[3]The use of the terminology "capacity" here is abuse of terminology since it depends on both the channel and the programming scheme. However, this term is used to indicate the achievable maximum information rate when the programming scheme PS is used over the channel W.

Proposition 2: Given a symmetric channel W, an average delay D, and a programming scheme $PS \in \mathcal{P}_T$, the following properties hold

1) $\mathcal{D}(W, PS(W, D)) = \min\{\mathcal{D}(W, PS), D\}$*, and*
2) $\mathcal{C}(W, PS(W, D)) = \min\left\{1, \frac{D}{\mathcal{D}(W,PS)}\right\} \cdot \mathcal{C}(W, PS)$.

Note that $\mathcal{D}(W, PS) = 0$ if and only if $PS = PS\left((1, 0)\right)$. In this case, $\mathcal{C}(W, PS(W, D)) = \mathcal{C}(W, PS) = 0$ by the definition of PS_0.

We next state another concept which will be helpful in solving Problem 1. The *normalized capacity* of a symmetric channel W using a programming scheme PS is defined to be

$$\overline{\mathcal{C}}(W, PS) = \begin{cases} \frac{\mathcal{C}(W,PS)}{\mathcal{D}(W,PS)}, & \text{if } \mathcal{D}(W, PS) > 0, \\ \mathcal{C}(W, PS), & \text{otherwise.} \end{cases} \quad (3)$$

The normalized capacity is the ratio between the maximum number of information bits that can be reliably stored and the average number of programming attempts. For abbreviation of notation we use $\overline{\mathcal{C}}(W, t)$ to denote the normalized capacity $\overline{\mathcal{C}}(W, PS_t)$.

Proposition 3 presents a strong connection between the normalized capacity of a channel W using programming scheme PS and its capacity under a constraint D.

Proposition 3: For a symmetric channel W, an average delay D, and a programming scheme PS, the following holds

$$\mathcal{C}(W, PS(W, D)) = \min\{D, \mathcal{D}(W, PS)\} \cdot \overline{\mathcal{C}}(W, PS).$$

Proof: By Proposition 2, if $\mathcal{D}(W, PS) \ge D$ then

$$\mathcal{C}(W, PS(W, D)) = \frac{D}{\mathcal{D}(W, PS)} \cdot \mathcal{C}(W, PS) = D \cdot \frac{\mathcal{C}(W, PS)}{\mathcal{D}(W, PS)} = D \cdot \overline{\mathcal{C}}(W, PS).$$

Otherwise, $\mathcal{D}(W, PS) < D$ and $\mathcal{C}(W, PS(W, D)) = \mathcal{C}(W, PS)$ by the definition of $PS(W, D)$, and $\mathcal{C}(W, PS) = \mathcal{D}(W, PS) \cdot \overline{\mathcal{C}}(W, PS)$, by the definition of the normalized capacity. □

In this paper we study the BSC, the BEC, and the Z channel. In these channels the cells store binary information, where in the BSC a programming failure changes the bit value in the cell, in the BEC a failure causes an erasure of an information bit, and lastly in the Z channel only the programming of cells which are programmed with value zero can fail.

For the Z channel, which is not a symmetric channel, the average delay depends also on the code, in particular, on the number of zeros in the codewords. Thus, the Z channel is discussed in a different section, Section V, in which we state similar definitions to Problem 1 and to the related concepts, $\mathcal{D}(W, t)$, $PS(W, D)$, and the normalized capacity.

Table I summarizes most of the notations used in this paper.

$$PS(W, D) = \begin{cases} PS, & \text{if } \mathcal{D}(W, PS) \le D \\ PS\left((1-\beta, 0), (\beta\beta_1, t_1), \ldots, (\beta\beta_\ell, t_\ell)\right), & \text{otherwise.} \end{cases} \quad (2)$$

TABLE I
SUMMARY OF NOTATIONS

Notation	Description
PS_t	Programming scheme with at most t attempts
$\mathcal{P}_T$	The set of all programming schemes with maximum delay T
$\mathcal{D}(W, PS)$	The average delay of PS over the channel W
$\mathcal{D}(W, t)$	The average delay of PS_t over the channel W
$\mathcal{D}(p, t)$	The average delay of PS_t over a memoryless symmetric channel with error probability p
$\mathcal{C}(W, PS)$	The capacity of channel W using programming scheme PS
$\mathcal{C}(W, t)$	The capacity of channel W using programming scheme PS_t
$F_1(W, D, T)$ Problem 1	The maximum capacity of channel W using $PS \in \mathcal{P}_T$ under an average delay constraint D
$PS(W, D)$	Adjusted PS to meet the constraint D for channel W
$\overline{\mathcal{C}}(W, PS)$	The normalized capacity of channel W using programming scheme PS
$\overline{\mathcal{C}}(W, t)$	The normalized capacity of channel W using programming scheme PS_t
$\mathrm{BSC}(p)$	The binary symmetric channel with crossover probability p, $0 \leq p \leq 1/2$
$\mathrm{BEC}(p)$	The binary erasure channel with erasure probability p, $0 \leq p < 1$
$Z(p)$	The Z channel with error probability p, $0 \leq p < 1$
$Z(p, \alpha)$	The Z channel with error probability p, and α fraction of ones in each codeword, $0 \leq p, \alpha \leq 1$
$PS_{t,q}$	PS_t where in the last attempt a question-mark is written with probability $1 - q$
$\mathcal{C}(W, t, q)$	The capacity of channel W using $PS_{t,q}$

III. The BSC and the BEC

In this section we study Problem 1 for the BSC and the BEC. Note that the results for the BSC have already been studied in [4]. However, we present them here in order to compare with the BEC and since these results will be used in Section VI for the case of programming with different error probabilities, and in Section VII for a new model which combines between the BSC and the BEC. Additionally, the translation between the notations by Bunte and Lapidoth [4] and our formulation is not immediate, and hence we found this repetition to be important for the readability and completeness of the results in the paper. For the same reasons, we provide proofs for the results on the BSC in this section. We note also that a general case, the discrete-input memoryless symmetric channels, is proved later in Section IV. However, for clarification and readability we discuss first the easier specific cases (the BSC and the BEC) in this section. Then, in Section IV the results are generalized for all the discrete-input memoryless symmetric channels by enhancing the methods and the techniques which are presented here.

According to well known results on the capacity of the BSC and the BEC we first establish the following lemma.

Lemma 4: For the programming scheme PS_t, $t \geq -1$, and for the $\mathrm{BSC}(p)$ and the $\mathrm{BEC}(p)$, the following properties hold:

1) *For all* $t \geq 1$, $\mathcal{C}(\mathrm{BSC}(p), t) = 1 - h(p^t)$,
2) *For all* $t \geq 1$, $\mathcal{C}(\mathrm{BEC}(p), t) = 1 - p^t$,
3) *For all* $t \geq 0$,
$$\mathcal{D}(p, t) \triangleq \mathcal{D}(\mathrm{BSC}(p), t) = \mathcal{D}(\mathrm{BEC}(p), t) = \frac{1 - p^t}{1 - p},$$
4) $\mathcal{D}(p, -1) \triangleq \mathcal{D}(\mathrm{BSC}(p), -1) = \mathcal{D}(\mathrm{BEC}(p), -1) = \frac{1}{1-p}$.

Proof:

For the programming scheme PS_t, $t \geq 1$, a cell will be erroneous if all its t programmings have failed, which happens with probability p^t. According to the known results on the capacity of the BSC and the BEC, we conclude claims 1 and 2 in the lemma regarding the capacity of the channels $\mathrm{BSC}(p)$ and $\mathrm{BEC}(p)$ using PS_t.

The average delay of PS_t for $t \geq 1$ is computed as follows. The average number of programmings a cell equals to $\sum_{j=1}^{t} j \cdot p_j$ where p_j is the probability that a cell is programmed exactly j times. Let q_i be the probability that a cell is programmed at least i times, that is, $q_i = \sum_{j=i}^{t} p_j$. Therefore, $\sum_{i=1}^{t} q_i = \sum_{i=1}^{t} \left(\sum_{j=i}^{t} p_j\right) = \sum_{j=1}^{t} j \cdot p_j$. Then, the average number of attempts to program a cell is equal to $\sum_{i=1}^{t} q_i$. Additionally, we note that $q_i = p^{i-1}$. Thus, we conclude that
$$\mathcal{D}(p, t) = \sum_{i=1}^{t} q_i = \sum_{i=0}^{t-1} p^i = \frac{1 - p^t}{1 - p}.$$
For $t = 0$ the average delay is zero, and for $t = -1$ the average delay is
$$\mathcal{D}(p, -1) = \sum_{i \geq 1} q_i = \sum_{i \geq 0} p^i = \frac{1}{1 - p}.$$
Note that $\mathcal{D}(p, t)$ is the average delay for any memoryless symmetric channel with error probability p. □

The next theorem compares between the normalized capacity when using PS_t and PS_{t+1}, for each $t \geq 1$. This result is used next in Corollary 6 which establishes the solution for Problem 1 for these two channels.

Theorem 5: For all $t \geq 1$ the following properties hold:

1) $\overline{\mathcal{C}}(\mathrm{BSC}(p), t) \leq \overline{\mathcal{C}}(\mathrm{BSC}(p), t + 1)$,
2) $\overline{\mathcal{C}}(\mathrm{BEC}(p), t) = 1 - p$.

Proof: It is possible to verify that the function
$$f(x) = \frac{1 - h(x)}{1 - x}$$
is decreasing in the range $0 \leq x \leq 1/2$, and by Lemma 4 we get
$$\overline{\mathcal{C}}(\mathrm{BSC}(p), t) = \frac{\mathcal{C}(\mathrm{BSC}(p), t)}{\mathcal{D}(\mathrm{BSC}(p), t)} = (1 - p) \cdot f(p^t).$$
Thus,
$$\begin{aligned}\overline{\mathcal{C}}(\mathrm{BSC}(p), t) &= (1 - p) \cdot f(p^t) \\ &\leq (1 - p) \cdot f(p^{t+1}) = \overline{\mathcal{C}}(\mathrm{BSC}(p), t + 1).\end{aligned}$$

For the BEC, by Lemma 4 we have

$$\overline{\mathcal{C}}\left(\mathrm{BEC}(p),t\right)=\frac{\mathcal{C}\left(\mathrm{BEC}(p),t\right)}{\mathcal{D}\left(\mathrm{BEC}(p),t\right)}=1-p.$$

□

The solutions to Problem 1 for the $\mathrm{BSC}(p)$ and the $\mathrm{BEC}(p)$ are presented in Corollary 6, where the result for the BSC has already been presented in [4]; see Proposition 5 therein. Note that if $D \geq \frac{1}{1-p}$ then the average delay is not constrained, since the average delay of any PS does not exceed the average delay of PS_{-1} which equals to $\frac{1}{1-p}$ (see also [4]).

Corollary 6: For $T \geq -1$, denote $\widehat{D}=\min\{\mathcal{D}(p,T),D\}$. The solution for Problem 1 for the BSC and the BEC is as follows.

1) *If $T \geq 0$ then*
   a) *$F_1\left(\mathrm{BSC}(p),D,T\right)=\widehat{D}\cdot\frac{(1-p)(1-h(p^T))}{1-p^T}$ and this value is obtained by the programming scheme $PS_T(\mathrm{BSC}(p),D)$,*
   b) *$F_1\left(\mathrm{BEC}(p),D,T\right) = \widehat{D}\cdot(1-p)$ and this value is obtained by the programming scheme $PS(\mathrm{BEC}(p),D)$ such that $\mathcal{D}(\mathrm{BEC}(p),PS) \geq \widehat{D}$, that is, for any $PS = PS((\beta_1,t_1),(\beta_2,t_2),\ldots(\beta_\ell,t_\ell))$ such that $\mathcal{D}(p,t_i)\geq\widehat{D}$ for all t_i.*
2) *$F_1\left(\mathrm{BSC}(p),D,-1\right) = F_1\left(\mathrm{BEC}(p),D,-1\right) = \widehat{D}\cdot(1-p)$ and this value is obtained for the BSC by the programming scheme $PS_{-1}(\mathrm{BSC}(p),D)$, and for the BEC by the programming scheme $PS(\mathrm{BEC}(p),D)$ such that $\mathcal{D}(\mathrm{BEC}(p),PS) \geq \widehat{D}$, that is, for any $PS = PS((\beta_1,t_1),(\beta_2,t_2),\ldots(\beta_\ell,t_\ell))$ such that $\mathcal{D}(p,t_i)\geq \widehat{D}$ for all t_i.*

Proof: In order to find the value of $F_1(W,D,T)$ where W is either the $\mathrm{BSC}(p)$ or the $\mathrm{BEC}(p)$, we let $PS = PS\left((\beta_1,t_1),\ldots,(\beta_\ell,t_\ell)\right)\in\mathcal{P}_T$ be a programming scheme which meets the constraint D, that is,

$$\mathcal{D}(W,PS)=\sum_{i=1}^{\ell}\beta_i\cdot\mathcal{D}(W,t_i)\leq\min\{D,\mathcal{D}(W,T)\}.$$

Then, for $W=\mathrm{BSC}(p)$ the capacity of the programming scheme PS over W satisfies

$$\begin{aligned}\mathcal{C}(W,PS) &= \sum_{i=1}^{\ell}\beta_i\cdot\mathcal{C}(W,t_i)\\ &\overset{(1)}{=} \sum_{i=1}^{\ell}\beta_i\cdot\mathcal{D}(W,t_i)\cdot\overline{\mathcal{C}}(W,t_i)\\ &\overset{(2)}{\leq} \sum_{i=1}^{\ell}\beta_i\cdot\mathcal{D}(W,t_i)\cdot\overline{\mathcal{C}}(W,T)\\ &= \overline{\mathcal{C}}(W,T)\sum_{i=1}^{\ell}\beta_i\cdot\mathcal{D}(W,t_i)\\ &\overset{(3)}{\leq} \overline{\mathcal{C}}(W,T)\cdot\min\{D,\mathcal{D}(W,T)\}\\ &\overset{(4)}{=} \mathcal{C}\left(W,PS_T(W,D)\right),\end{aligned}$$

where (1) is by the definition of the normalized capacity, (2) is by Theorem 5, (3) is by $\mathcal{D}(W,PS)=\sum_{i=1}^{\ell}\beta_i\cdot\mathcal{D}(W,t_i)\leq\min\{D,\mathcal{D}(W,T)\}$, and (4) is by Proposition 3. To complete the proof for the BSC channel, we note that $\mathcal{C}\left(W,PS_T(W,D)\right)=\widehat{D}\cdot\frac{(1-p)(1-h(p^T))}{1-p^T}$ by Proposition 2 and Lemma 4. A similar proof holds for $F_1\left(\mathrm{BEC}(p),D,T\right)$. □

Remark 7: The claims in Lemma 4 regarding the BSC were presented in Proposition 3 in [4], and the result in Corollary 6 for the BSC was presented in Proposition 5 in [4]. We note that in Proposition 3 in [4], ϵ,ζ is equivalent to $p, D-1$ in our notations, respectively. Furthermore, the gap in the solution from [4] and our result stems from the fact that we let cells to be not programmed at all, while in [4] a cell has to be programmed at least once. Thus, the translation between these two approaches can be done by substituting the average delay constraint D with $\zeta+1$.

IV. Discrete-Input Symmetric Channel

In this section we solve Problem 1 for all the discrete-input memoryless symmetric channels. Note that the BSC, the BEC, and the binary-input additive white Gaussian noise (BI-AWGN) channel are all discrete-input memoryless symmetric channels.

We will prove that the normalized capacity is non-decreasing. This is a generalization of Theorem 5 which proves this property for the BSC and the BEC. An immediate conclusion is that given W, a discrete-input memoryless symmetric channel, the solution for Problem 1 is obtained by the programming scheme $PS_T(W,D)$. We assume here that the channel output is continuous. However, the proof argument also applies to channels with discrete output by replacing the integral with summation.

Assume that $\mathcal{X}$ is discrete and $\mathcal{Y}$ is possibly continuous. Let $f(y|x)$ be the channel density. The set of successful channel outputs for input x is denoted by $\mathcal{Y}_x=\{y|p(X=x|y)>p(X=x'|y),\forall x'\neq x\}$. Note that the strict inequality is significant for channels with erasure. The complement of $\mathcal{Y}_x$ is denoted by $\bar{\mathcal{Y}}_x=\mathcal{Y}\setminus\mathcal{Y}_x$. So for any $x\in\mathcal{X}$, the probability of writing failure is $p_x=\int_{\bar{\mathcal{Y}}_x}f(y|x)dy$. Since the channel is symmetric, p_x is equal for all x, and is denoted by $p=p_x$.

Denote the channel by $W(p)$, and let $g(y|x)$ be the channel density after at most t writing attempts. Let $n=|\mathcal{X}|$. The marginal output density for uniform input is $g(y)=\sum_x\frac{g(y|x)}{n}$. The channel capacity using PS_t is given by

$$\mathcal{C}(W(p),t)=\max_{prob(x)}I(X;Y)=\sum_x\int_{\mathcal{Y}}\frac{g(y|x)}{n}\log\frac{g(y|x)}{g(y)}dy.$$

Theorem 8: Let $W(p)$ be a discrete-input memoryless symmetric channel with input set $\mathcal{X}$, output set $\mathcal{Y}$, channel density $f(y|x)$, and error probability $p=\int_{\bar{\mathcal{Y}}_x}f(y|x)dy$ for $x\in\mathcal{X}$. Then, for all $t\geq 0$, $\overline{\mathcal{C}}\left(W(p),t\right)\leq\overline{\mathcal{C}}\left(W(p),t+1\right)$.

Proof: Let $z=p^t$. Since $p<1$, z is decreasing on t, and it is enough to show that $\overline{\mathcal{C}}'=\frac{\partial\overline{\mathcal{C}}}{\partial z}\leq 0$, where for abbreviation we denote $\mathcal{C}(W(p),t)$ and $\overline{\mathcal{C}}=\overline{\mathcal{C}}(W(p),t)$ by $\mathcal{C}$ and $\overline{\mathcal{C}}$, respectively.

As proved in Lemma 4, for all $t\geq 0$, $\mathcal{D}(W(p),t)=\frac{1-z}{1-p}$. Then, $\overline{\mathcal{C}}(W(p),t)=\frac{\mathcal{C}(W(p),t)}{\mathcal{D}(W(p),t)}=\frac{(1-p)\mathcal{C}(W(p),t)}{1-z}$, and

$\overline{\mathcal{C}}' = (1-p)\frac{\partial}{\partial z}\left(\frac{\mathcal{C}}{1-z}\right) = (1-p)(\mathcal{C}'(1-z)+\mathcal{C})$, where $\mathcal{C}' = \frac{\partial \mathcal{C}}{\partial z}$. Thus, we should only prove that $\mathcal{C}'(1-z)+\mathcal{C} \le 0$.

The derivative of the output density according to z satisfies $g'(y) = \sum_x \frac{g'(y|x)}{n}$, and therefore

$$\begin{aligned}
&\mathcal{C}' \\
&= \sum_x \int_{\mathcal{Y}} \left(\frac{g'(y|x)}{n}\log\frac{g(y|x)}{g(y)} + \frac{g'(y|x)g(y) - g'(y)g(y|x)}{ng(y)ln2}\right)dy \\
&= \left(\sum_x \int_{\mathcal{Y}} \frac{g'(y|x)}{n}\log\frac{g(y|x)}{g(y)}dy\right) \\
&\quad + \int_{\mathcal{Y}} \frac{g'(y)g(y) - g'(y)g(y)}{g(y)ln2}dy \\
&= \sum_x \int_{\mathcal{Y}} \frac{g'(y|x)}{n}\log\frac{g(y|x)}{g(y)}dy.
\end{aligned}$$

That is,

$$\begin{aligned}
&\mathcal{C} + \mathcal{C}'(1-z) \\
&\qquad = \frac{1}{n}\sum_x \int_{\mathcal{Y}} (g(y|x) + (1-z)g'(y|x))\log\frac{g(y|x)}{g(y)}dy,
\end{aligned}$$

and we will prove that $\sum_x \int_{\mathcal{Y}} (g(y|x) + (1-z)g'(y|x))\log\frac{g(y|x)}{g(y)}dy \le 0$.

Let us compute $g(y|x)$ for $y \in \bar{\mathcal{Y}}_x$. Note that $p(X=x|y) \le p(X=x'|y)$ for some $x' \ne x$ only if the writing fails in the first $t-1$ attempts, which happens with probability p^{t-1}. Thus, $g(y|x) = f(y|x)p^{t-1} = \frac{f(y|x)z}{p}$ for $y \in \bar{\mathcal{Y}}_x$. A similar argument shows that for $y \in \mathcal{Y}_x$, $g(y|x) = f(y|x)\frac{1-p^t}{1-p} = f(y|x)\frac{1-z}{1-p}$. Note that for $y \in \mathcal{Y}_x$, $g(y|x) = -(1-z)g'(y|x)$ and therefore $g(y|x) + (1-z)g'(y|x) = 0$. Additionally, for $y \in \bar{\mathcal{Y}}_x$, $g(y|x) = zg'(y|x)$, and then $g(y|x) + (1-z)g'(y|x) = g'(y|x)$.

So it suffices to show that $\sum_x \int_{\bar{\mathcal{Y}}_x} g'(y|x)\log\frac{g(y|x)}{g(y)}dy \le 0$. Since $g'(y|x) = f(y|x)/p \ge 0$ for $y \in \bar{\mathcal{Y}}_x$, and $g(y) = \frac{1}{n}\sum_x g(y|x)$, we should only prove that $\int_{\bar{\mathcal{Y}}_x} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy \le 0$ for all x. Since $\bar{\mathcal{Y}}_x = \mathcal{Y} \setminus \mathcal{Y}_x$, we have

$$\begin{aligned}
&\int_{\bar{\mathcal{Y}}_x} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy \\
&\qquad = \int_{\mathcal{Y}} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy - \int_{\mathcal{Y}_x} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy.
\end{aligned}$$

Applying Bayes' Theorem to the definition of $\mathcal{Y}_x$ and noting that $p(x) = 1/n$, we have $f(y|x) > f(y|x')$ for all $x' \ne x$. Therefore, for $y \in \mathcal{Y}_x$,

$$\begin{aligned}
g(y|x) &= f(y|x)\frac{1-p^t}{1-p} \ge f(y|x) \\
&> f(y|x') \ge f(y|x')p^{t-1} = g(y|x'),
\end{aligned}$$

implying that

$$\begin{aligned}
&\int_{\mathcal{Y}} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy - \int_{\mathcal{Y}_x} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy \\
&\quad < \int_{\mathcal{Y}} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy - \int_{\mathcal{Y}_x} \log\frac{ng(y|x)}{ng(y|x)}dy \\
&\quad = \int_{\mathcal{Y}} \log\frac{ng(y|x)}{\sum_{x'} g(y|x')}dy \\
&\quad = \int_{\mathcal{Y}} \log g(y|x)dy - \int_{\mathcal{Y}} \log\frac{\sum_{x'} g(y|x')}{n}dy.
\end{aligned}$$

Now using Jensen's inequality we have

$$\begin{aligned}
&\int_{\mathcal{Y}} \log g(y|x)dy - \int_{\mathcal{Y}} \log\frac{\sum_{x'} g(y|x')}{n}dy \\
&\quad \le \int_{\mathcal{Y}} \log g(y|x)dy - \int_{\mathcal{Y}} \frac{\sum_{x'} \log g(y|x')}{n}dy \\
&\quad = \int_{\mathcal{Y}} \log g(y|x)dy - \sum_{x'} \int_{\mathcal{Y}} \frac{\log g(y|x')}{n}dy.
\end{aligned}$$

Finally, the channel symmetry implies that $\int_{\mathcal{Y}} \log g(y|x)dy$ is equal for all x, implying that

$$\begin{aligned}
&\int_{\mathcal{Y}} \log g(y|x)dy - \sum_{x'} \int_{\mathcal{Y}} \frac{\log g(y|x')}{n}dy \\
&\qquad\qquad = \int_{\mathcal{Y}} \log g(y|x)dy - \int_{\mathcal{Y}} \log g(y|x)dy = 0,
\end{aligned}$$

completing the proof.

□

Now we conclude the solution for any discrete-input memoryless symmetric channel.

Corollary 9: Let $W(p)$ be a discrete-input memoryless symmetric channel with input set $\mathcal{X}$, output set $\mathcal{Y}$, channel density $f(y|x)$, and error probability $p = \int_{\bar{\mathcal{Y}}_x} f(y|x)dy$ for any x. The solution for Problem 1 for $W(p)$ is

1) *for $T \ge 0$, $F_1(W(p), D, T) = \widehat{D} \cdot \frac{1-p}{1-p^T} \cdot \mathcal{C}(W(p), T)$, and this value is obtained by the programming scheme $PS_T(W(p), D)$,*
2) *$F_1(W(p), D, -1) = F_1(W(p), D, -1) = \widehat{D} \cdot (1-p)$, and this value is obtained by the programming scheme $PS_{-1}(W(p), D)$,*

where $\widehat{D} = \min\left\{\frac{1-p^T}{1-p}, D\right\}$ and $\mathcal{C}(W(p), t) = \sum_x \int_{\mathcal{Y}} \frac{g(y|x)}{n}\log\frac{g(y|x)}{g(y)}dy$.

V. The Z Channel

In this section we study programming schemes for the Z channel with error probability p, i.e., 0 is flipped to 1 with probability p, $0 \le p < 1$, but programming 1 always succeeds. This channel is denoted by $Z(p)$.

The capacity of the channel $Z(p)$ was well studied in the literature; see e.g. [31], [32], [35]. We denote by $Z(p,\alpha)$ the Z channel where α is the probability for occurrence of 1 in a codeword, and p is the crossover $0 \to 1$ probability. The capacity of $Z(p,\alpha)$ was shown to be [31], [32], [35]

$$\mathcal{C}(Z(p,\alpha)) \triangleq h((1-\alpha)(1-p)) - (1-\alpha)h(p).$$

where the capacity of the $Z(p,\alpha)$ channel denotes the mutual information $I(Y;X) = H(Y) - H(Y|X)$ between the input random variable X with distribution $Pr(X=1) = \alpha$ and $Pr(X=0) = 1-\alpha$, and the output random variable Y with distribution $Pr(Y=1|X=0) = p$, $Pr(Y=0|X=0) = 1-p$ and $Pr(Y=1|X=1) = 1$.

In the Z channel, the average delay of programming a zero cell is exactly as in the BSC and the BEC cases, but a cell with value one is programmed only once. Therefore, the average delay depends on the number of cells which are programmed with value zero, and hence we define $\mathcal{D}(Z(p,\alpha),t)$ as the average delay of the programming scheme PS_t when the programming process is modeled by the channel $Z(p)$ and α is the fraction of ones in the codewords. The capacity $\mathcal{C}(Z(p,\alpha),t)$ is defined to be the capacity of the channel $Z(p)$ when α is the probability for occurrence of one in the codewords and the programming scheme PS_t is applied. The following lemma is readily proved.

Lemma 10: For the programming scheme PS_t and the channel $Z(p,\alpha)$, the following properties hold:

1) *For all $t \geq 0$,*

$$\begin{aligned}\mathcal{C}(Z(p,\alpha),t) &= \mathcal{C}(Z(p^t,\alpha)) \\ &= h((1-\alpha)(1-p^t)) - (1-\alpha)h(p^t),\end{aligned}$$

2) $\mathcal{C}(Z(p,\alpha),-1) = h(\alpha)$,
3) *For all $t \geq 1$, $\mathcal{D}(Z(p,\alpha),t) = \frac{(1-\alpha)(1-p^t)}{1-p} + \alpha$,*
4) $\mathcal{D}(Z(p,\alpha),-1) = \frac{1-\alpha}{1-p} + \alpha$, $\mathcal{D}(Z(p,\alpha),0) = 0$.

Let $PS((\beta_1,t_1),(\beta_2,t_2),\ldots,(\beta_\ell,t_\ell))$ be a programming scheme, and $\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\ldots,\alpha_\ell)$ where $0 \leq \alpha_i \leq 1$, for all $1 \leq i \leq \ell$. Then, we define

$$\mathcal{C}(Z(p,\boldsymbol{\alpha}),PS) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{C}(Z(p,\alpha_i),t_i),$$

that is, $\mathcal{C}(Z(p,\boldsymbol{\alpha}),PS)$ is the capacity of $Z(p)$ while using the programming scheme PS and the parameter $\boldsymbol{\alpha}$. Similarly, we define the average delay of the programming scheme PS for $Z(p)$ using the parameter $\boldsymbol{\alpha}$ as

$$\mathcal{D}(Z(p,\boldsymbol{\alpha}),PS) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}(Z(p,\alpha_i),t_i).$$

Thus, we formulate Problem 1 for the Z channel as follows.

Problem 11: 1 - the Z channel. Given the channel $Z(p)$, an average delay D, and a maximum delay T, find a programming scheme, $PS \in \mathcal{P}_T$, and a vector $\boldsymbol{\alpha}$ which maximize the capacity $\mathcal{C}(Z(p,\boldsymbol{\alpha}),PS)$, under the constraint that $\mathcal{D}(Z(p,\boldsymbol{\alpha}),PS)) \leq D$. In particular, given $Z(p)$, D, and T, find the value of

$$F_1(Z(p),D,T) = \sup_{PS\in\mathcal{P}_T,\boldsymbol{\alpha}:\mathcal{D}(Z(p,\boldsymbol{\alpha}),PS)\leq D,\boldsymbol{\alpha}\in[0,1]^\ell,0\leq\ell} \{\mathcal{C}(Z(p,\boldsymbol{\alpha}),PS)\}.$$

In order to solve Problem 1 for the Z channel, we use the normalized capacity of a programming scheme PS_t over $Z(p,\alpha)$ which is defined as in Equation (3) for $t \neq 0$ by

$$\overline{\mathcal{C}}(Z(p,\alpha),t) = \frac{\mathcal{C}(Z(p,\alpha),t)}{\mathcal{D}(Z(p,\alpha),t)},$$

and $\overline{\mathcal{C}}(Z(p,\alpha),0) = 0$. For $t \geq 1$ we have

$$\overline{\mathcal{C}}(Z(p,\alpha),t) = \frac{(1-p)\left[h((1-\alpha)(1-p^t)) - (1-\alpha)h(p^t)\right]}{(1-\alpha)(1-p^t)+\alpha(1-p)},$$

and for $t = -1$ it holds that

$$\overline{\mathcal{C}}(Z(p,\alpha),-1) = \frac{(1-p)h(\alpha)}{(1-p\alpha)}.$$

Given p,t, the maximum normalized capacity of PS_t is $\overline{\mathcal{C}}(Z(p),t) = \max_{0\leq\alpha\leq1}\left\{\overline{\mathcal{C}}(Z(p,\alpha),t)\right\}$, and we denote by $\alpha^*(p,t)$ the value of α which achieves this capacity. That is,

$$\overline{\mathcal{C}}(Z(p),t) = \overline{\mathcal{C}}(Z(p,\alpha^*(p,t)),t) = \max_{0\leq\alpha\leq1}\left\{\overline{\mathcal{C}}(Z(p,\alpha),t)\right\},$$

and the average delay $\mathcal{D}(Z(p),t)$ is defined by

$$\mathcal{D}(Z(p),t) = \mathcal{D}(Z(p,\alpha^*(p,t)),t).$$

Next, we define the programming scheme $PS_t(Z(p),D)$ similarly to the definition in Equation (2). $PS_t(Z(p),D)$ is a scheme in which the cells are programmed by PS_t until the average delay is D, and then the rest of the cells are not programmed. That is, denote by $\beta = \frac{D}{\mathcal{D}(Z(p),t)}$, and

$$PS_t(Z(p),D) = \begin{cases} PS_t, & \text{if } \mathcal{D}(Z(p),t) \leq D \\ PS((1-\beta,0),(\beta,t)), & \text{otherwise.}\end{cases}$$

Given a constraint on the maximum delay, T,we define $t^*(T) = \arg\max_{0\leq t\leq T}\{\overline{\mathcal{C}}(Z(p),t)\}$ for $T \geq 0$ and $t^*(-1) = \arg\max_{-1\leq t}\{\overline{\mathcal{C}}(Z(p),t)\}$.

Thus, we can conclude the following corollary which is proved in a similar technique as Corollary 6. The proof is presented in Appendix A.

Corollary 12: $F_1(Z(p),D,T) = \min\{\mathcal{D}(Z(p),T),D\} \cdot \overline{\mathcal{C}}(Z(p),t^*(T)) = \mathcal{C}\left(Z(p),PS_{t^*(T)}(Z(p),D)\right)$ and this value is obtained by $PS_{t^*(T)}(Z(p),D)$ with parameter $\alpha^*(p,t^*(T))$.

The solution for the Z channel can be obtained by finding the value of $t^*(T)$ and $\alpha^*(p,t^*(T))$. We could not solve this explicitly, however we present some computational results. By the partial derivative of $\overline{\mathcal{C}}(Z(p,\alpha),t)$ with respect to α, we get that given p and t, $\alpha^*(p,t)$ is a root of the following function[4]

$$\begin{aligned}f(p,t) = &(1-p)(1-p^t)\log((1-\alpha)(1-p^t)) \\ &+(2p^t-1-p^{t+1})\log(1-(1-\alpha)(1-p^t)) \\ &+(1-p)h(p^t).\end{aligned}$$

In Fig. 1 we present plots of the normalized capacity $\overline{\mathcal{C}}(Z(p),t)$ for $t \in \{-1,1,2,3,4\}$. The x-axis is p, and each plot represents the value of $\overline{\mathcal{C}}(Z(p),t)$ for a specific t. We also add the plot of the function $1-p$ to compare between $\overline{\mathcal{C}}(Z(p),t)$ and $1-p$ which is the maximum normalized capacity for the BSC(p) and the BEC(p). We note that $1-p$ is smaller than $\overline{\mathcal{C}}(Z(p),t)$ for almost all values of t. Following these computational results, we conjecture that $\overline{\mathcal{C}}(Z(p),t) \leq \overline{\mathcal{C}}(Z(p),t+1)$ for all $t \geq 0$, and thus $t^*(T) = T$ and $F_1(Z(p),D,T) = \min\{D,\mathcal{D}(Z(p),T)\} \cdot \overline{\mathcal{C}}(Z(p),T)$.

VI. Different Error Probabilities

In this section we generalize the programming model we have studied so far. We no longer assume that there is only a single channel which mimics the cell programming attempts, but each programming attempt has its own channel. We study and formulate this generalization only for the BSC and the

[4]All logarithms in this paper are taken according to base 2.

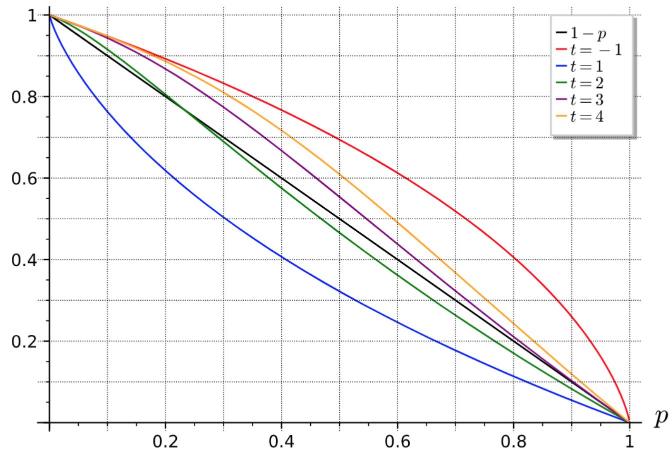

Fig. 1. The normalized capacity of $Z(p)$ using PS_t for some values of t, comparing to $1-p$, the maximum normalized capacity of the $\mathrm{BSC}(p)$ and the $\mathrm{BEC}(p)$.

BEC, however modifications for other channels can be handled similarly.

For the rest of this section, we refer to the channel W as either $\mathrm{BSC}(p)$ or $\mathrm{BEC}(p)$. We assume that it is possible to reprogram the cells, however the error probabilities on different programming attempts may be different. For example, for *hard* cells in flash memories [23], [30], i.e., cells that their programming is more difficult, if the first attempt of a cell programming has failed, then the probability for failure on the second trial may be larger since the cell is hard to be programmed. In other cases, the error probability in the next attempt may be smaller since the previous trials might increase the success probability of the subsequent programming attempts. The phenomenon of hard cells can be modeled by the idea of uncertainty in the parameters of the cells [34]. However, there are some cases, e.g., when the hardness of the cells is known, that are captured by the model of different error probabilities.

Let $\mathbf{P} = (p_1, p_2, \dots) = (p_i)_{i=1}^{\infty}$ be a sequence of probabilities, where p_t is the error probability on the t-th programming attempt. We model the programming process as a transmission over the *channel sequence*, $W(\mathbf{P}) = W(p_i)_{i=1}^{\infty}$, where on the t-th trial, the programming is modeled as transmission over the channel $W(p_t)$. That is, all the channels in $W(\mathbf{P})$ have the same type of errors, but may have different error probabilities. Recall that for the BSC we assume that $0 \le p_i \le 1/2$ for all $i \ge 1$, while for the BEC, $0 \le p_i < 1$.

For $t \ge -1$ and a channel sequence $W(\mathbf{P})$, we denote by $\mathcal{D}(W(\mathbf{P}), t)$ the average delay of the programming scheme PS_t, which is the expected number of times to program a cell when the programming process is modeled by $W(\mathbf{P})$. For example, for the BSC (see Lemma 14),

$$\mathcal{D}(\mathrm{BSC}(\mathbf{P}), t) = \sum_{i=0}^{t-1} \left(\prod_{j=1}^{i} p_j \right).$$

When a cell is programmed according to a programming scheme PS_t, we can model this process as transmission over the channels sequence $W(p_i)_{i=1}^{t}$, and an error occurs if and only if there is an error in each one of the t channels. Define $Q_i = \Pi_{j=1}^{i} p_j$ for $i \ge 1$. Then, for $t \ge 1$, Q_t is the probability of a programming failure when using PS_t, and hence the capacity $\mathcal{C}(W(\mathbf{P}), t)$ is defined to be $\mathcal{C}(W(Q_t))$. For example, for $W = \mathrm{BSC}$ and $t \ge 1$, $\mathcal{C}(\mathrm{BSC}(\mathbf{P}), t) = \mathcal{C}(\mathrm{BSC}(Q_t)) = 1 - h(Q_t)$.

We focus on the set $\mathcal{P}_T$ of the programming schemes that was defined in (1). It can be readily verified that the average delay of a programming scheme $PS \in \mathcal{P}_T$, $PS = PS((\beta_1, t_1), \dots, (\beta_\ell, t_\ell))$, over the channel sequence $W(\mathbf{P})$ is given by

$$\mathcal{D}(W(\mathbf{P}), PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{D}(W(\mathbf{P}), t_i),$$

and the definition of the capacity is extended as follows

$$\mathcal{C}(W(\mathbf{P}), PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{C}(W(\mathbf{P}), t_i).$$

We are now ready to formally define the problem we study in this section.

Problem 13 (- Different Probabilities): Given a sequence of probabilities $\mathbf{P}$ *with a channel* $W \in \{\mathrm{BSC}, \mathrm{BEC}\}$*, an average delay* D*, and a maximum delay* T*, find a programming scheme* $PS \in \mathcal{P}_T$*, which maximizes the capacity* $\mathcal{C}(W(\mathbf{P}), PS)$*, under the constraint that* $\mathcal{D}(W(\mathbf{P}), PS) \le D$*. In particular, find the value of*

$$F_2(W(\mathbf{P}), D, T) \triangleq \sup_{PS \in \mathcal{P}_T : \mathcal{D}(W(\mathbf{P}), PS) \le D} \{\mathcal{C}(W(\mathbf{P}), PS)\}.$$

We note that the results presented in Section III regarding Problem 1 can be derived from the solutions for Problem 13 presented in this section by substituting $p_i = p$ for all $i \ge 1$.

For $\mathbf{P} = (p_1, p_2, \dots)$ and $Q_i \triangleq \Pi_{j=1}^{i} p_j$, define $Y_t \triangleq \sum_{i=1}^{t-1} Q_i$ for $t \ge 1$ ($Y_1 = 0$), and $Y_{-1} \triangleq \sum_{i=1}^{\infty} Q_i$. The next lemma establishes the basic properties on the average delay and the capacity of these channels.

Lemma 14: For the programming scheme PS_t*, and* $\mathbf{P} = (p_1, p_2, \dots)$*, the following properties hold*:

1) *For* $t \ge 1$, $\mathcal{C}(\mathrm{BSC}(\mathbf{P}), t) = 1 - h(Q_t)$,
2) *For* $t \ge 1$, $\mathcal{C}(\mathrm{BEC}(\mathbf{P}), t) = 1 - Q_t$,
3) *For* $t \ne 0$, $\mathcal{D}(\mathbf{P}, t) \triangleq \mathcal{D}(\mathrm{BSC}(\mathbf{P}), t) = \mathcal{D}(\mathrm{BEC}(\mathbf{P}), t) = 1 + Y_t$.

Proof: Note that Q_t is the probability of an error in the first t attempts. Using the known capacities of the BSC and the BEC, we get the values for the capacities in cases 1 and 2.

The average delay of the programming scheme PS_t over the channel sequence $\mathrm{BSC}(\mathbf{P})$ or $\mathrm{BEC}(\mathbf{P})$, which we denoted by $\mathcal{D}(W(\mathbf{P}), t)$, is calculated as follows. Let q_i be the probability that a cell is programmed at least i times. Note that for $1 < i < t$, $q_i = Q_{i-1}$ and $q_1 = 1$ for both cases. Then, we conclude that for $t \ge 1$,

$$\mathcal{D}(W(\mathbf{P}), t) = \sum_{i=1}^{t} q_i = 1 + \sum_{i=1}^{t-1} Q_i = 1 + Y_t,$$

and $\mathcal{D}(W(\mathbf{P}), -1) = \sum_{i=1}^{\infty} q_i = 1 + \sum_{i=1}^{\infty} Q_i = 1 + Y_{-1}$. □

For this generalization of the problem, given a programming scheme $PS \in \mathcal{P}_T$, the programming scheme $PS(W, D)$ and the normalized capacity are defined in a similar way as in the original definitions in Equations (2) and (3), respectively. The following proposition is a generalization of Proposition 3.

Proposition 15: Given a channel sequence $W(\mathbf{P})$ denoted in short by W, an average delay D, and a programming scheme PS, the following holds,

$$\mathcal{C}\left(W, PS(W,D)\right) = \min\{D, \mathcal{D}(W(,PS)\} \cdot \overline{\mathcal{C}}(W, PS).$$

Next we study the relation between $\overline{\mathcal{C}}\left(W(\mathbf{P}), t\right)$ and $\overline{\mathcal{C}}\left(W(\mathbf{P}), t+1\right)$ both for the BSC and the BEC and for arbitrary sequence of probabilities $\mathbf{P}$.

Theorem 16: For $t \geq 1$, and $\mathbf{P} = (p_1, p_2, \ldots)$ such that for all i, $0 \leq p_i \leq 1/2$, it holds that

$$\overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t+1\right).$$

Proof: First we state that for all $0 \leq x, p \leq 1/2$ it holds that

$$(1 - h(x))(1 + x) \leq 1 - h(x/2) \leq 1 - h(xp). \tag{4}$$

Now, we want to prove that

$$\frac{1 - h(Q_t)}{1 + Y_t} = \overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t+1\right) = \frac{1 - h(Q_{t+1})}{1 + Y_{t+1}}.$$

This is equivalent to proving that

$$1 - h(Q_t) \cdot \frac{1 + Y_{t+1}}{1 + Y_t} \leq 1 - h(Q_{t+1}),$$

or

$$1 - h(Q_t) \cdot \frac{1 + Y_t + Q_t}{1 + Y_t} \leq 1 - h(p_{t+1}Q_t),$$

which holds if and only if

$$1 - h(Q_t) \cdot \left(1 + \frac{Q_t}{1 + Y_t}\right) \leq 1 - h(p_{t+1}Q_t).$$

But,

$$\frac{Q_t}{1 + Y_t} \leq Q_t$$

and by substituting $x = Q_t$ and $p = p_{t+1}$ in Inequality (4) we conclude that

$$(1 - h(Q_t)) \cdot \left(1 + \frac{Q_t}{1 + Y_t}\right) \leq (1 - h(Q_t)) \cdot (1 + Q_t) \leq (1 - h(p_{t+1}Q_t)),$$

and therefore

$$\overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BSC}(\mathbf{P}), t+1\right),$$

as required. □

By Theorem 16 we conclude the following corollary, whose proof is similar to the proof of Corollary 6, using Proposition 15.

Corollary 17: For a channel sequence $W = \text{BSC}(\mathbf{P})$ the solution for Problem 13 is $F_2(W, D, T) = \mathcal{C}\left(W, PS_T(W, D)\right)$ and it is obtained by the programming scheme $PS_T(W, D)$.

In the rest of this section, we solve a special case for $\text{BEC}(\mathbf{P})$.

Theorem 18: For a sequence of probabilities $\mathbf{P} = (p_1, p_2, \ldots)$, for all $t \geq 1$,

$$\overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t+1\right),$$

if and only if

$$p_{t+1} \leq \frac{Y_{t+1}}{Y_t + 1}.$$

Proof: According to Lemma 14, the following relation holds

$$\frac{1 - Q_t}{1 + Y_t} = \overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t+1\right) = \frac{1 - Q_{t+1}}{1 + Y_{t+1}},$$

if and only if

$$(1 - Q_t) \cdot (1 + Y_{t+1}) \leq (1 - Q_{t+1}) \cdot (1 + Y_t).$$

This holds if and only if

$$-Q_t - Q_tY_{t+1} + Y_{t+1} \leq -Q_{t+1} - Q_{t+1}Y_t + Y_t$$

or

$$Y_{t+1} - Q_t - Y_t - Q_tY_{t+1} \leq -Q_{t+1} - Q_{t+1}Y_t,$$

which translates to

$$-Q_tY_{t+1} + Q_{t+1} + Q_{t+1}Y_t \leq 0,$$

and

$$Q_tp_{t+1}(1 + Y_t) \leq Q_tY_{t+1},$$

and finally

$$p_{t+1} \leq \frac{Y_{t+1}}{(1 + Y_t)}.$$

□

Theorem 19: Let $\mathbf{P} = (p_1, p_2, \ldots)$ be a sequence of probabilities such that $1 > p_1 \geq p_2 \geq p_3 \cdots$. Then, for all $t \geq 1$,

$$\overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t+1\right).$$

Proof: According to Theorem 18

$$\overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t\right) \leq \overline{\mathcal{C}}\left(\text{BEC}(\mathbf{P}), t+1\right)$$

if and only if

$$p_{t+1} \leq \frac{Y_{t+1}}{(1 + Y_t)}$$

or

$$p_{t+1}\left(1 + Y_t\right) \leq Y_{t+1}$$

and by the definition of Y_t

$$p_{t+1} + p_{t+1}\left(\sum_{i=1}^{t-1} Q_i\right) \leq \sum_{i=1}^{t} Q_i$$

and thus

$$p_{t+1} + p_{t+1}\left(\sum_{i=1}^{t-1} Q_i\right) - \sum_{i=1}^{t} Q_i \leq 0.$$

By the definition of Q_i and since $p_1 \geq p_2 \geq p_3 \cdots$ we have that

$$p_{t+1}\left(\sum_{i=1}^{t-1} Q_i\right) \leq p_t\left(\sum_{i=1}^{t-1} Q_i\right) \leq \sum_{i=2}^{t} Q_i,$$

and by $p_{t+1} \leq p_1 = Q_1$ we conclude that

$$p_{t+1} + p_{t+1}\left(\sum_{i=1}^{t-1} Q_i\right) - \sum_{i=1}^{t} Q_i \leq 0,$$

as required. □

By Theorem 19 we can finally conclude with the following corollary.

Corollary 20: For a channel sequence $W = \text{BEC}(\mathbf{P})$ where $\mathbf{P} = (p_1, p_2, \ldots)$ such that $1 > p_1 \geq p_2 \geq p_3 \cdots$, the solution for Problem 13 is $F_2(W, D, T) = \mathcal{C}(W, PS_T(W, D))$ and it is obtained by the programming scheme $PS_T(W, D)$.

VII. Combined Programming Schemes for the BSC and the BEC

In this section, we study programming schemes for the BSC, in which on the last programming attempt it is possible to either try to reprogram the failed cell again with its value or instead program it with a special question mark to indicate a programming failure. This model is motivated by several applications. For example, when synthesizing DNA strands, if the attachment of the next base to the strand fails on multiple attempts, it is possible to attach instead a different molecule to indicate this base attachment failure [16]. In flash memories we assume that if some cell cannot reach its correct value, then it will be possible to program it to a different level (for example a high voltage level that is usually not used) in order to indicate a programming failure of the cell.

We denote by $PS_{t,q}$ the programming scheme in which on the t-th programming attempt, which is the last one, the cell is programmed without verification with probability q, and with probability $1-q$ it is programmed with the question mark symbol $'?'$.

Let p be the programming error probability, where programming $'?'$ is always successful. The average delay of programming a cell with $PS_{t,q}$ over the $\text{BSC}(p)$, denoted by $\mathcal{D}(\text{BSC}(p), t_i, q_i)$, does not depend on q, and hence equals to $\mathcal{D}(p, t)$. However the capacity is clearly influenced by the parameter q.

The probability that a cell will be erroneous after $t-1$ programming attempts is p^{t-1}. Therefore, programming with $PS_{t,q}$ over the $\text{BSC}(p)$ can be represented by a channel with the following transitions probabilities

$$p(y|x) = \begin{cases} p^{t-1}(1-q) & \text{if } y = ? \\ p^{t-1}q(1-p) + (1-p^{t-1}) & \text{if } x = y \\ p^{t-1}qp & \text{otherwise,} \end{cases}$$

where x, y is the input, output bit of the channel, respectively. Denote $b = p^{t-1}$. The capacity of this channel is [6, Porblem 7.13]

$$\begin{aligned}\mathcal{C}(\text{BSC}(p), t, q) &= (1-b+bq)\left(1 - h\left(\frac{bpq}{1-b+bq}\right)\right) \\ &= 1-b+bq \\ &\quad -(1-b+bq)\log(1-b+bq) \\ &\quad +(1-b+bq-bqp)\log(1-b+bq-bqp) \\ &\quad +bpq\log(bpq),\end{aligned}$$

where $\mathcal{C}(W, t, q)$ represents the capacity of channel W using $PS_{t,q}$.

Note that, $\mathcal{C}(\text{BSC}(p), t, 0) = 1 - p^{t-1}$, and $\mathcal{C}(\text{BSC}(p), t, 1) = 1 - h(p^t)$. For example, for $t = 1$,

$$\mathcal{C}(\text{BSC}(p), 1, q) = q - q\log(q) + q(1-p)\log(q(1-p)) + qp\log(qp).$$

Let $PS = PS((\beta_1, t_1), (\beta_2, t_2), \ldots, (\beta_\ell, t_\ell)) \in \mathcal{P}_T$ be a programming scheme, and $\boldsymbol{q} = (q_1, q_2, \ldots, q_\ell)$ where $0 \leq q_i \leq 1$, for all $1 \leq i \leq \ell$. Then, we define

$$\mathcal{C}(\text{BSC}(p), PS, \boldsymbol{q}) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{C}(\text{BSC}(p), t_i, q_i).$$

That is, $\mathcal{C}(\text{BSC}(p), PS, \boldsymbol{q})$ is the capacity of the $\text{BSC}(p)$ when using the programming scheme PS with the parameter $\boldsymbol{q}$. Similarly, we define the average delay of the programming scheme PS for $\text{BSC}(p)$ using the parameter $\boldsymbol{q}$ as

$$\mathcal{D}(\text{BSC}(p), PS, \boldsymbol{q}) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}(\text{BSC}(p), t_i, q_i).$$

Note that $\mathcal{D}(\text{BSC}(p), PS, \boldsymbol{q}) = \mathcal{D}(\text{BSC}(p), PS)$.

For this model, Problem 1 will be formulated as follows.

Problem 21 (- The Combined Channel): Given the $\text{BSC}(p)$, an average delay D, and a maximum delay T, find a programming scheme, $PS \in \mathcal{P}_T$, and $\boldsymbol{q}$ which maximize the capacity $\mathcal{C}(\text{BSC}(p), PS, \boldsymbol{q})$, under the constraint that $\mathcal{D}(\text{BSC}(p), PS, \boldsymbol{q}) \leq D$. In particular, given $\text{BSC}(p)$, D, and T, find the value of

$$F_3(\text{BSC}(p), D, T) \triangleq \sup_{\mathcal{D}(\text{BSC}(p), PS, \boldsymbol{q}) \leq D} \{\mathcal{C}(\text{BSC}(p), PS, \boldsymbol{q})\}.$$

For this generalization of the model, the programming scheme $PS(W, D)$ and the normalized capacity are defined in a similar way as in the original definitions in Equations (2) and (3), respectively.

Given p, t we define

$$\mathcal{C}_m(\text{BSC}(p), t) = \max_{q \in [0,1]} \{\mathcal{C}(\text{BSC}(p), t, q)\},$$

and the normalized capacity $\overline{\mathcal{C}}_m(\text{BSC}(p), t) = \frac{\mathcal{C}_m(\text{BSC}(p), t)}{\mathcal{D}(p, t)}$.

In the rest of this section we prove that the best scheme is $PS_{T,1}(W, D)$ or $PS_{T,0}(W, D)$, i.e., the standard PS_T or the new PS_T in which in the last attempt all the erroneous cells are programmed with a question mark.

Lemma 22: Given p and t,

$$\mathcal{C}_m(\text{BSC}(p), t) = \max\{\mathcal{C}(\text{BSC}(p), t, 0), \mathcal{C}(\text{BSC}(p), t, 1)\}.$$

Proof: If $p = 0$ then there are no errors, and the maximum capacity is obtained for all q. Given $0 < p \leq 1/2$, if $t = 1$ then $\mathcal{C}(\text{BSC}(p), t, q) = q(1 - h(p))$ and the maximum value is obtained for $q = 1$. Otherwise, given p, t, such that $0 < p \leq 1/2$ and $1 < t$, we prove that the function $\mathcal{C}(\text{BSC}(p), t, q)$ has no local maximum in the range of $0 < q < 1$. We define $f_{t,p}(q) = \mathcal{C}(\text{BSC}(p), t, q)$ to be a function of q, and prove that $f_{t,p}(q)$ has no local maximum in the range of $0 < q < 1$. This is accomplished by showing that the second derivative of $f_{t,p}(q) = \mathcal{C}(\text{BSC}(p), t, q)$ is positive in this range.

The first derivative is

$$\frac{\partial f_{t,p}(q)}{\partial q} = b(1-p)\log(1-b+bq-bqp) + bp\log(bpq) - b\log(1-b+bq) + b,$$

and then the second derivative is

$$\frac{\partial^2 f_{t,p}(q)}{\partial q^2} = \frac{b^2(1-p)^2}{(1-b+bq-bpq)\ln 2} + \frac{(bp)^2}{bpq\ln 2} - \frac{b^2}{(1-b+bq)\ln 2}.$$

To show that $\frac{\partial^2 f_{t,p}(q)}{\partial q^2} > 0$, it is sufficient to prove that

$$\frac{(1-p)^2}{1-b+bq-bpq} + \frac{p^2}{bpq} - \frac{1}{1-b+bq} > 0.$$

We denote $x_1 = (1-b+bq-bpq)$ and $x_2 = bpq$ Thus, we want to prove that

$$\frac{(1-p)^2}{x_1} + \frac{p^2}{x_2} - \frac{1}{x_1+x_2} > 0.$$

Note that $x_1 = 1 - b(1-q(1-p)) > 0$ and $x_2 = bpq > 0$ since $0 < b, p \leq 1/2$ and $0 < q < 1$. Thus, it is sufficient to prove that

$$(1-p)^2 x_2(x_1+x_2) + p^2 x_1(x_1+x_2) - x_1x_2 > 0,$$

which holds if and only if

$$((1-p)x_2 - px_1)^2 > 0.$$

Lastly, the last equation holds since $(1-p)x_2 = px_1$ implies $0 = p(1-b)$ which is impossible since $0 < p \leq 1/2$, $1 < t$, and $b = p^{t-1}$. □

The last lemma proved that for all p, t, the capacity $\mathcal{C}(\text{BSC}(p), t, q)$ is achieved for $q = 0$ or for $q = 1$, by comparing between $\mathcal{C}(\text{BSC}(p), t, 0) = 1 - p^{t-1}$ which is obtained for $PS_{t,0}$ $(q = 0)$ and $\mathcal{C}(\text{BSC}(p), t, 1) = 1 - h(p^t)$ which is attained for $PS_{t,1}$ $(q = 1)$. Note that for $q = 0$, the last programming actually provides a complete verification for all the successful cells by substituting a question mark in all the erroneous cells. On the other hand, for $q = 1$ in the last programming all the failed cells are reprogrammed again. The decision to use $q = 0$ or $q = 1$ is determined by some threshold that depends on p and t, as will be described later in this section.

Theorem 23: For all $t \geq 0$,

$$\overline{\mathcal{C}}_m(\text{BSC}(p), t) \leq \overline{\mathcal{C}}_m(\text{BSC}(p), t+1)$$

Proof: By Lemma 22 $\mathcal{C}_n(\text{BSC}(p), t) = \max\{\mathcal{C}(\text{BSC}(p), t, 0), \mathcal{C}(\text{BSC}(p), t, 1)\}$. If $\mathcal{C}_m(\text{BSC}(p), t)$ is obtained for $q = 1$, then the claim is implied by Theorem 5. Otherwise, $\overline{\mathcal{C}}_m(\text{BSC}(p), t) = \frac{(1-p)(1-p^{t-1})}{1-p^t}$ and $\overline{\mathcal{C}}_m(\text{BSC}(p), t+1) \geq \frac{(1-p)(1-p^t)}{1-p^{t+1}}$. Then, the claim is true since $p \leq 1/2$ implies $\frac{1-p^t}{1-p^{t+1}} \geq \frac{1-p^{t-1}}{1-p^t}$. □

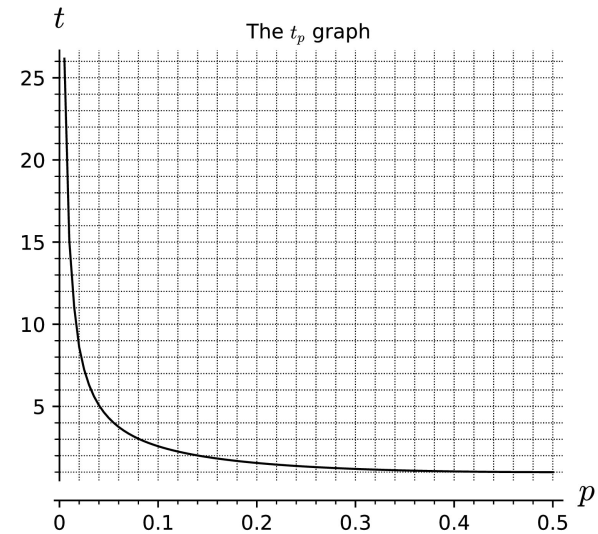

Fig. 2. The t_p graph.

By applying the same technique as in the proof of Corollary 6 with using Lemma 22 and Theorem 23 we solve Problem 1 for the new model.

Corollary 24: Denote by $\widehat{D} = \min\{\mathcal{D}(p, T), D\}$. Then, the solution for Problem 21 is $F_3(\text{BSC}(p), D, T) = \frac{\widehat{D}}{\mathcal{D}(p,T)} \cdot \max\{1 - h(p^t), 1 - p^{t-1}\}$, which is obtained by $PS_{T,1}(D)$ or $PS_{T,0}(D)$.

For each p we denote by t_p the smallest value of t, such that $h(p^t) \geq p^{t-1}$ (t_p may be a noninteger). Since $h(px) \geq ph(x)$ for $0 \leq x, p \leq 1/2$, we conclude that for each $t \geq t_p$ it holds that $h(p^t) \geq p^{t-1}$. Thus, for a given p, $t \geq t_p$ if and only if $\mathcal{C}(\text{BSC}(p), t, 0) = 1 - p^{t-1} \geq 1 - h(p^t) = \mathcal{C}(\text{BSC}(p), t, 1)$. Let T be the last attempt to program. If $T \geq t_p$ then in the T-th attempt all the failed cells will be programmed with question marks. Otherwise, they will be reprogrammed with their value (without verification). In Fig. 2 the t_p values are presented in a graph, where the horizontal axis is p and the vertical axis is t. The graph line is $f(p) = t_p$.

VIII. Conclusion

In this paper we studied a model which describes the process of cell programming in memories. We focused on the case where the programming is modeled by a discrete-input memoryless symmetric channel or by the Z channel, and accordingly, we designed programming schemes that maximize the number of information bits that can be reliably stored in the memory, while the average and maximum numbers of times to program a cell are constrained. While this work established several interesting observations on the programming strategies in memories and transmission schemes, there are still several questions that remain open. In particular, the generalization of this model to multilevel cells, and to a setup in which the cells are programmed in parallel.

In addition, the paper contains results regarding channels which have monotonic non-decreasing normalized capacity, which means that normalized capacity increases as the number of iterations increases. An interesting study is to find necessary conditions on channels for having the monotonic non-increasing property. Additionally, generalizing the results in Sections VI and VII for some types of channels by using the technique that was proposed in Section IV is an interesting direction for future work.

APPENDIX A

In this part we present the omitted proofs in the paper.

Corollary 25: 12 $F_1(Z(p), D, T) = \min\{\mathcal{D}(Z(p), T), D\} \cdot \overline{\mathcal{C}}(Z(p), t^*(T)) = \mathcal{C}(Z(p), PS_{t^*(T)}(Z(p), D))$ *obtained by* $PS_{t^*(T)}(Z(p), D)$ *with parameter* $\alpha^*(p, t^*(T))$.

Proof: Let $PS = PS((\beta_1, t_1), \ldots, (\beta_\ell, t_\ell)) \in \mathcal{P}_T$ be a programming scheme which meets the constraint D with the parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_t)$. Thus, we have

$$\begin{aligned} \mathcal{C}\left(Z(p, \boldsymbol{\alpha}), PS\right) &= \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{C}\left(Z(p, \alpha_i), t_i\right) \\ &\underset{(1)}{=} \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}\left(Z(p, \alpha_i), t_i\right) \cdot \overline{\mathcal{C}}\left(Z(p, \alpha_i), t_i\right) \\ &\underset{(2)}{\leq} \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}\left(Z(p, \alpha_i), t_i\right) \cdot \overline{\mathcal{C}}\left(Z(p), t^*(T)\right) \\ &= \overline{\mathcal{C}}\left(Z(p), t^*(T)\right) \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}\left(Z(p, \alpha_i), t_i\right) \\ &\underset{(3)}{\leq} \overline{\mathcal{C}}\left(Z(p), t^*(T)\right) \cdot D \end{aligned}$$

where (1) is by the definition of the normalized capacity, (2) is by $t^*(T)$ and $\overline{\mathcal{C}}\left(Z(p), t^*(T)\right)$ definitions, and (3) is since PS meets the average delay constraint D with parameter $\boldsymbol{\alpha}$. □

ACKNOWLEDGMENT

The authors would like to thank Amos Lapidoth for helpful discussions. They are also indebted to the two anonymous reviewers as well as the Associate Editor Prof. Ramji Venkataramanan for their important comments.

REFERENCES

[1] M. Anantha, B. Bose, and L. G. Tallini, "ARQ protocols and unidirectional codes," *IEEE Trans. Comput.*, vol. 56, no. 4, pp. 433–443, Apr. 2007.

[2] M. Blawat *et al.*, "Forward error correction for DNA data storage," in *Proc. Int. Conf. Comput. Sci.*, vol. 80, 2016, pp. 1011–1022.

[3] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," in *Proc. 21st Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, Atlanta, GA, USA, Apr. 2016, pp. 637–649.

[4] C. Bunte and A. Lapidoth, "On the storage capacity of rewritable memories," in *Proc. IEEE 26th Conv. Electr. Electron. Eng. Isr.*, Nov. 2010, pp. 402–405.

[5] C. Bunte and A. Lapidoth, "Computing the capacity of rewritable memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 2512–2516.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2012.

[7] S. Elmougy, L. Pezza, L. G. Tallini, A. Al-Dhelaan, and B. Bose, "Diversity combining type I-hybrid ARQ protocol over *m*-ary asymmetric Varshamov channels," *Comput. Electr. Eng.*, vol. 77, pp. 389–397, Jul. 2019.

[8] Y. Erlich and D. Zielinski, "DNA Fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, 2017.

[9] M. Franceschini, L. Lastras-Montano, T. Mittelholzer, and M. Sharma, "The role of feedback in rewritable storage channels [lecture notes]," *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 190–194, Nov. 2009.

[10] R. Gallager, *Information Theory and Reliable Communication*. Hoboken, NJ, USA: Wiley, 1968.

[11] J. Gibson, *Information Theory and Rate Distortion Theory for Communications and Compression*. San Mateo, CA, USA: Morgan Kaufmann, 2013.

[12] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, pp. 77–80, Jan. 2013.

[13] M. Horovitz, E. Yaakobi, E. E. Gad, and J. Bruck, "Iterative programming of noisy memory cells," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Gotland, Sweden, Aug. 2019, pp. 1–5.

[14] A. Jiang and J. Bruck, "On the capacity of flash memories," in *Proc. Int. Symp. Inf. Theory Appl.*, Auckland, New Zealand, Dec. 2008, pp. 94–99.

[15] A. Jiang and H. Li, "Optimized cell programming for flash memories," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process.*, Victoria, BC, Canada, Aug. 2009, pp. 914–919.

[16] S. Kosuri and G. Church, "Large-scale de novo DNA synthesis: Technologies and applications," *Nature Methods*, vol. 11, no. 5, pp. 499–507, May 2014.

[17] T. Klove, P. Oprisan, and B. Bose, "Diversity combining for the Z-channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1174–1178, Mar. 2005.

[18] L. A. Lastras-Montano, M. M. Franceschini, and T. Mittelholzer, "The capacity of the uniform noise rewritable channel with average cost," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun. 2010, pp. 201–205.

[19] L. A. Lastras-Montano, M. Franceschini, T. Mittelholzer, and M. Sharma, "Rewritable storage channels," in *Proc. Int. Symp. Inf. Theory Its Appl.*, Toronto, ON, Canada, Dec. 20, pp. 7–10.

[20] L. A. Lastras-Montaño, M. M. Franceschini, T. Mittelholzer, and M. Sharma, "On the capacity of memoryless rewritable storage channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3178–3195, Jun. 2004.

[21] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. London, U.K.: Pearson-Hall, 2004.

[22] L. A. Lastras-Montano, T. Mittelholzer, and M. M. Franceschini, "Superposition coding in rewritable channels," in *Proc. Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Jan. 2010, pp. 1–9.

[23] H.-T. Lue *et al.*, "Study of incremental step pulse programming (ISPP) and STI edge effect of BE-SONOS NAND flash," in *Proc. IEEE Int. Rel. Phys. Symp.*, May 2008, pp. 693–694.

[24] T. Mittelholzer, M. Franceschini, L. A. Lastras-Montano, I. M. Elfadel, and M. Sharma, "Rewritable channels with data-dependent noise," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–6.

[25] T. Mittelholzer, L. A. Lastras-Montano, M. Sharma, and M. Franceschini, "Rewritable storage channels with limited number of rewrite iterations," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun 2002, pp. 973–977.

[26] P. Oprisan and B. Bose, "Z-channel feedback error control," in *Proc. IEEE Inform. Theory Workshop*, Bangalore, India, Oct. 2002, p. 210.

[27] M. Qin, A. A. Jiang, and P. H. Siegel, "Parallel programming of rank modulation," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 719–723.

[28] M. Qin, E. Yaakobi, and P. H. Siegel, "Optimized cell programming for flash memories with quantizers," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2780–2795, May 2014.

[29] P. Sindhu, "Retransmission error control with memory," *IEEE Trans. Commun.*, vol. C-25, no. 5, pp. 473–479, May 1977.

[30] K.-D. Suh *et al.*, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.

[31] L. G. Tallini, S. Al-Bassam, and B. Bose, "Feedback codes achieving the capacity of the Z-channel," *IEEE Trans. Inform. Theory*, vol. 54, no. 3, pp. 1357–1362, Mar. 2008.

[32] L. G. Tallini, S. Al-Bassam, and B. Bose, "Correction to 'feedback codes achieving the capacity of the Z-channel,"' *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4348–4350, Sep. 2009.

[33] L. G. Tallini, S. Elmougy, and B. Bose, "Analysis of plain and diversity combining gybrid ARQ protocols over the $m(\geq 2)$-ary asymmetric channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5550–5558, Dec. 2006.

[34] R. Venkataramanan, S. Tatikonda, L. Lastras-Monta, and M. Franceschini, "Rewritable storage channels with hidden state," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 815–824, May 2014.

[35] S. Verdu, "Channel capacity," in *The Electrical Engineering Handbook*. Boca Raton, FL, USA: CRC Press, 1997, ch. 73.5, pp. 1671–1678.

[36] E. Yaakobi, A. Jiang, P. H. Siegel, A. Vardy, and J. K. Wolf, "On the parallel programming of flash memory cells," in *Proc. IEEE Inf. Theory Workshop*, Dublin, Ireland, Aug. 2010, pp. 1–5.

[37] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 1, pp. 1–6, Dec. 2017.

Michal Horovitz (Member, IEEE) received the B.Sc. degree from the Department of Mathematics and the Department of Computer Science, The Open University of Israel, Ra'anana, Israel, in 2009, and the Ph.D. degree from the Computer Science Department, Technion—Israel Institute of Technology, Haifa, Israel, in 2017. She is currently a Lecturer with the Department of Computer Science, Tel-Hai College, Israel. She is also a Researcher with the Galilee Research Institute—Migal, Upper Galilee, Israel. Her research interests include coding theory with applications to non-volatile memories and DNA storage, information theory, combinatorics, machine learning, and optimization.

Eitan Yaakobi (Senior Member, IEEE) received the B.A. degree in computer science and mathematics and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology, and the Center for Memory and Recording Research, University of California, San Diego. He is currently an Associate Professor at the Computer Science Department, Technion—Israel Institute of Technology. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010–2011.

Eyal En Gad received the B.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology in 2008 and the Ph.D. degree in electrical engineering from the California Institute of Technology in 2015. From 2015 to 2016, he was a Post-Doctoral Scholar at the University of South California. He joined Micron Technology in 2016, working on error-correcting coding schemes and algorithms for flash memories.

Jehoshua Bruck (Life Fellow, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion—Israel Institute of Technology in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1989.

He is currently the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and systems and the theory of computation in nature. His industrial and entrepreneurial experiences include working with IBM Research, where he has participated in the design and implementation of the first IBM parallel computer, cofounding and serving as the Chairman for Rainfinity (acquired in 2005 by EMC), that created the first virtualization solution for network attached storage; cofounding and serving as the Chairman for XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system; and cofounding and serving as the Chairman for MemVerge, a start-up company that is pioneering big memory computing.

Dr. Bruck was a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award, and the IBM Outstanding Technical Achievement Award.