Iterative Programming of Noisy Memory Cells

Michal Horovitz*, **Eitan Yaakobi**†, **Eyal En Gad**‡, and **Jehoshua Bruck**§

* Dept. of Computer Science, Tel-Hai College, and The Galilee Research Institute - Migal, Upper Galilee, Israel

†Dept. of Computer Science, Technion—Israel Inst. of Tech., Haifa 32000, Israel

‡ Micron Technology, Milipitas, CA 95035, USA

§ Dept. of Electrical Engineering, California Inst. of Tech., Pasadena, CA 91125, USA

Emails: **horovitzmic@telhai.ac.il, yaakobi@cs.technion.ac.il, eengad@micron.com, bruck@caltech.edu**

***Abstract*—This paper studies a model, which was first presented by Bunte and Lapidoth, that mimics the programming operation of memory cells. Under this paradigm it is assumed that the cells are programmed sequentially and individually. The programming process is modeled as transmission over a channel, while 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 for maximizing 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 the BSC, the BEC, or the Z channel, where we focus on a specific broad family of programming schemes. 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.**

I. Introduction

Many of the current and future memories, such as DRAM, SRAM, phase-change memories, STT-MRAM, flash memories, consist of memory cells. 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. The process of changing the cell state, called *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 become 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 figure of merits. Namely, we consider a model in which the cells are programmed sequentially, one after the other [2], [3]. Assume n binary cells are programmed. The cell programming process is modeled as transmission over some *discrete memoryless channel* (*DMC*), for example the *binary symmetric channel* (*BSC*), the *binary erasure channel* (*BEC*), 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 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 is required.

Previous works considered programming schemes mostly for flash memory cells. In [11], 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 [12], an algorithm was shown for optimizing the expected cell programming precision, when the programming noise follows a random distribution. In [22], algorithms for parallel programming of flash memory cells were studied which were then extended in [17]. Other works studied the programming schemes with continuous-alphabet channels, see [6], [14], [15], [20] and references therein.

Our point of departure in this paper is the programming model which was first presented in [2], [3] by Bunte and Lapidoth for discrete alphabet memory channels (DMC). In particular, in [2] the case of symmetric channels with focus on the BSC was studied. We extend the results from [2] and study the problem for the BEC and the Z channel, where the last is applicable in particular for flash memories. 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 [2], we note that we propose a slightly different formulation to the problem and model, which we found to be more suitable to the cases we solve in this paper that are motivated by common programming techniques. The proposed programming schemes in the paper are rather basic but at the same time are simple to implement, which increases their practicality.

Yet another model studied in this work is motivated by DNA-based storage systems [1], [5], [9]. DNA synthesis is the process of artificially creating DNA molecules such that arbitrary single stranded DNA sequences of length few hundreds bases can be generated chemically. When synthesizing DNA strands, the bases are added one after the other to form the long sequence [13]. 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, it is possible to add another different base which indicates a synthesis failure in this location.

The rest of the paper is organized as follows. In Section II, we formally present the definitions and the problem. We study the BSC and the BEC in Section III and the Z channel in Section IV. We generalize in Section V 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 is studied in Section VI. Due to lack of space, most of the proofs and several details are omitted and can be found in the extended version of this paper in [10].

II. Definitions and Basic Properties

Let W be a discrete memoryless channel (DMC). We model the process of programming a cell as 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 a *programming scheme*, is an algorithm which states the rules to program the n cells. Its *average delay* over a channel W is defined to be the ratio between the expected number of programming attempts and the number of cells, 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. For $t \geq 0$, the strategy of the programming scheme PS_t is to program the cell until its programming *succeeds*, that is, it stores its correct value or the number of attempts is t. Hence, 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 PS_{-1} denotes the programming scheme where the cell is programmed until it stores the correct value. For notational purposes, we denote this programming scheme by PS_{-1} instead of PS_∞.

In this paper we focus on programming schemes that consist of combinations of several schemes from $\{PS_t\}_{t\geq -1}$. Let $PS\left((\beta_1,t_1),(\beta_2,t_2),\dots,(\beta_\ell,t_\ell)\right)$ 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}[1]. Formally, for $T \geq 0$, the maximum number of attempts to program a cell, we define the following set of programming schemes.

$$\mathcal{P}_T = \Bigg\{ PS\left((\beta_1,t_1),(\beta_2,t_2),\dots,(\beta_\ell,t_\ell)\right) : \\ 0 \leq t_1,\dots,t_\ell \leq T, 0 < \beta_1,\dots,\beta_\ell \leq 1, \sum_{i=1}^{\ell} \beta_i = 1 \Bigg\}, \quad (1)$$

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

Remark 1. *We focus here on this specific family of iterative programming schemes. A higher rate may be achieved by resorting to programming schemes outside this family, for example, schemes which allow accumulating delay, randomized strategies, or defining more general stopping rules [2]. However, we find this family of programming schemes more practical as it mimics common techniques for programming cells in flash and other memories.*

For asymmetric channels the average delay of PS_t may depend also on 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 this section we refer only for symmetric channels[2], while these concepts will be defined similarly in Section IV 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. When a cell is programmed according to a programming scheme PS_t, we can model this process as a 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 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 error probability. We denote the capacity of the channel W_t by $\mathcal{C}(W,t)$. Note that for every channel W, it holds that $\mathcal{C}(W,0)=0$ and $\mathcal{D}(W,0)=0$.

For $T \geq -1$, it can be readily verified that for a programming scheme $PS = PS\left((\beta_1,t_1),\dots,(\beta_\ell,t_\ell)\right) \in \mathcal{P}_T$ over a symmetric channel W, the average delay, denoted by $\mathcal{D}(W,PS)$, is given by $\mathcal{D}(W,PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{D}(W,t_i)$. Similarly, the *capacity*[3] of the the channel W using programming scheme PS is denoted by $\mathcal{C}(W,PS)$ and is defined to be $\mathcal{C}(W,PS) = \sum_{i=1}^{\ell} \beta_i \mathcal{C}(W,t_i)$.

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 delay, that is, the average number of attempts to program a cell, is at most some prescribed value D, and the number of attempts to program a cell is at most T, i.e., the maximum delay is at most T. The case of $T=-1$ corresponds to having no constraint on the 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 $\mathcal{D}(W,PS) \leq D$. In particular, given W ,D, and T, find the value of $F_1(W,D,T) \triangleq \max_{PS\in\mathcal{P}_T:\mathcal{D}(W,PS)\leq D}\{\mathcal{C}(W,PS)\}$.*

Assume that for a given symmetric channel W, an average delay D, and a programming scheme $PS = PS\left((\beta_1,t_1),\dots,(\beta_\ell,t_\ell)\right) \in \mathcal{P}_T$, it holds 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 $\frac{D}{\mathcal{D}(W,PS)}$ fraction of the cells with the programming scheme PS, and the remaining cells are not programmed. Hence, we define the programming scheme $PS(W,D)$ as follows:

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

where $\beta = \frac{D}{\mathcal{D}(W,PS)}$. It can be readily verified that $\mathcal{D}(W,PS(W,D)) = \min\{\mathcal{D}(W,PS),D\}$ and $\mathcal{C}(W,PS(W,D)) = \min\left\{1,\frac{D}{\mathcal{D}(W,PS)}\right\}\cdot\mathcal{C}(W,PS)$. Note that $\mathcal{D}(W,PS)=0$ if and only if $PS=PS((1,0))$, and then we define $\mathcal{C}(W,PS(W,D)) = \mathcal{C}(W,PS)$ which is equal to zero 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.

Proposition 2 presents a strong connection between the normalized capacity and the capacity of a channel W using a programming scheme PS under a constraint D.

Proposition 2. *For a symmetric channel W, an average delay D, and a programming scheme PS, the following holds*

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

For clarity and readability, Table I summarizes the notations used throughout the paper.

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 presented in [2], however we state them here in order to compare with the BEC and since these results and techniques will be used in Section V for the case of programming with different error probabilities, and in Section VI for a new model. Additionally,

[1]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.

[2]We refer to the definition of symmetric DMCs as defined in [7] which are also called *partitioned symmetric* [8], 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.

[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.

TABLE I: Summary of notations.

Notation	Description
PS_t	Programming scheme with at most t attempts
$\mathcal{P}_T$	The set of programming schemes with maximum delay T
$\mathcal{D}(W, PS)$ $\mathcal{D}(W, t)$ $\mathcal{D}(p, t)$	The average delay of PS over the channel W The average delay of PS_t over the channel W The average delay of PS_t over the $BSC(p)$ or the $BEC(p)$
$\mathcal{C}(W, PS)$ $\mathcal{C}(W, t)$	The capacity of channel W using PS The capacity of channel W using 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
$\overline{\mathcal{C}}(W, PS)$ $\overline{\mathcal{C}}(W, t)$	The normalized capacity of channel W using PS The normalized capacity of channel W using PS_t
$BSC(p)$ $BEC(p)$ $Z(p)$	The binary symmetric channel with crossover probability p, $0 \le p \le 0.5$ The binary erasure channel with erasure probability p, $0 \le p \le 1$ The Z channel with error probability p, $0 \le p \le 1$
$Z(p, \alpha)$	The Z channel with error probability p and probability α for occurrence of 1, $0 \le p,\ \alpha \le 1$
$PS_{t,q}$	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 $PS_{t,q}$

the translation between the notations by Bunte and Lapidoth [2] 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.

According to well known results on the capacity of the BSC and the BEC, the following is proved. In this paper $h(x)$ is the binary entropy function where $0 \le x \le 1$.

Lemma 3. *For the programming scheme PS_t, $t \ge -1$, and error probability p for the BSC and the BEC, the following properties hold:*

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

The next theorem compares between the normalized capacity when using PS_t and PS_{t+1} for each $t \ge 1$. This result is used next in Corollary 5 which establishes the solution to Problem 1.

Theorem 4. *For all $t \ge 1$ the following properties hold:*

1) *$\overline{\mathcal{C}}(BSC(p), PS_t) \le \overline{\mathcal{C}}(BSC(p), PS_{t+1})$,*
2) *$\overline{\mathcal{C}}(BEC(p), PS_t) = 1 - p$.*

Note that if $D \ge \frac{1}{1-p}$ 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 [2]).

Corollary 5. *For $T \ge -1$, denote $D' = \min\{\mathcal{D}(p, T), D\}$. The solution to Problem 1 for the BSC and the BEC is as follows.*

1) *If $T \ge 0$ then*
   a) *$F_1(BSC(p), D, T) = D' \cdot \frac{(1-p)(1-h(p^T))}{1-p^T}$ and this value is obtained by the programming scheme $PS_T(BSC(p), D)$.*
   b) *$F_1(BEC(p), D, T) = D' \cdot (1-p)$ and this value is obtained by the programming scheme $PS_t(BEC(p), D)$ for any t such that $0 \le t \le T$ and $\mathcal{D}(W, t) \ge D'$.*
2) *$F_1(BSC(p), D, -1) = F_1(BEC(p), D, -1) = D' \cdot (1 - p)$ and this value is obtained by the programming scheme $PS_{-1}(BEC(p), D)$ for the BSC, and by the programming scheme $PS_t(BEC(p), D)$ for any t such that $\mathcal{D}(W, t) \ge D'$ for the BEC.*

Remark 6. *The results in Lemma 3 and in Corollary 5 regarding the BSC were presented in Proposition 3 and in Proposition 5 in [2], respectively. We note that in Proposition 3 in [2], ϵ, ζ are equivalent to $p, D - 1$ in our notations, respectively. Furthermore, the gap in the solution from [2] and our result stems from the fact that we let cells to be not programmed at all, while in [2] 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. The Z Channel

In this section we study programming schemes for $Z(p)$, the Z channel with error probability p, i.e., 0 is flipped to 1 with probability p, where $0 \le p \le 1$. The capacity of $Z(p)$ was well studied in the literature; see e.g. [19], [21]. We denote by $Z(p, \alpha)$ the Z channel where α is the probability for occurrence of 1 and p is the crossover $0 \to 1$ probability. The capacity of $Z(p, \alpha)$ was shown for example in [19], [21], and is denoted here by $\mathcal{C}(Z(p^t, \alpha))$.

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 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 programmed words. 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 and the programming scheme PS_t is applied. The following lemma is readily proved.

Lemma 7. *For the programming scheme PS_t and the channel $Z(p, \alpha)$, the following properties hold:*

1) *For all $t \ge 0$, $\mathcal{C}(Z(p, \alpha), t) = \mathcal{C}(Z(p^t, \alpha))$.*
2) *$\mathcal{C}(Z(p, \alpha), -1) = h(\alpha)$.*
3) *For all $t \ge 1$, $\mathcal{D}(Z(p, \alpha), t) = \frac{(1-\alpha)(1-p^t)}{1-p} + \alpha$.*
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 $0 \le \alpha_i \le 1$, for all $1 \le i \le \ell$. 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 $Z(p)$ 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 1 - Z channel. *Given the Z 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 $\mathcal{D}(Z(p, \boldsymbol{\alpha}), PS) \le D$. In particular, given $Z(p)$, D, and T, find the value of*
$$F_1(Z(p), D, T) \triangleq \max_{PS \in \mathcal{P}_T, \boldsymbol{\alpha} : \mathcal{D}(Z(p,\boldsymbol{\alpha}), PS) \le D} \{\mathcal{C}(Z(p, \boldsymbol{\alpha}), PS)\}.$$

In order to solve Problem 1 for the Z channel, we use the normalized capacity of $Z(p, \alpha)$ using a programming scheme PS_t, denoted by $\overline{\mathcal{C}}(Z(p, \alpha), t)$, which is defined as in Equation (3). Given p and t, the maximum normalized capacity using PS_t is $\overline{\mathcal{C}}(Z(p), t) = \max_{0 \le \alpha \le 1} \left\{\overline{\mathcal{C}}(Z(p, \alpha), t)\right\}$, and we denote by $\alpha^*(p, t)$ any value of α which achieves this capacity. That is, $\overline{\mathcal{C}}(Z(p), t) = \overline{\mathcal{C}}_t(Z(p, \alpha^*(p, t))) = \max_{0 \le \alpha \le 1} \left\{\overline{\mathcal{C}}(Z(p, \alpha), t)\right\}$, 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). Given T, a constraint on the maximum delay, we define $t^*(T) = \arg\max_{0 \le t \le T} \{\overline{\mathcal{C}}(Z(p), t)\}$ for $T \ge 0$ and $t^*(-1) = \arg\max_{-1 \le t} \{\overline{\mathcal{C}}(Z(p), t)\}$. Thus, we can conclude the following corollary which is proved in a similar technique to Corollary 5.

Corollary 8. $F_1(Z(p),D,T) = \min\{\mathcal{D}(Z(p),T),D\} \cdot \overline{\mathcal{C}}(Z(p),PS_{t^*(T)}) = \mathcal{C}(Z(p),PS_{t^*(T)}(Z(p),D))$ *and this value is obtained by* $PS_{t^*(T)}(Z(p),D)$ *with parameter* $\alpha^*(p,t^*(T))$.

An explicit solution for the Z channel can be obtained by finding the value of $t^*(T)$ and $\alpha^*(p,t^*(T))$. 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, $F_1(Z(p),D,T) = \min\{D,\mathcal{D}(Z(p),T)\}\cdot\overline{\mathcal{C}}(Z(p),T)$.

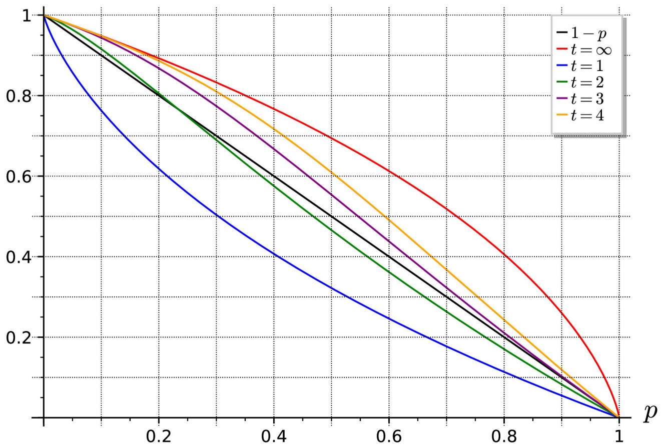

Fig. 1: The normalized capacity of $Z(p)$ using PS_t for some values of t, comparing to $1-p$, the maximum normalized capacity for the $BSC(p)$ and the $BEC(p)$.

V. Different Error Probabilities

In this section we generalize the programming model for the BSC and the BEC. 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 [16], [18], 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.

Let $\mathbf{P} = (p_1,p_2,\dots) = (p_i)_{i=1}^{\infty}$ be a probabilities sequence, 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 i-th trial, the programming is modeled as transmission over the channel $W(p_i)$. 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 $0 \leq p_i \leq 0.5$ for all $i \geq 1$, while for the BEC, $0 \leq p_i \leq 1$.

For $t \geq -1$ 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 $W(\mathbf{P})$.

When a cell is programmed according to a programming scheme PS_t, we can model this process as transmission over the set of channel sequence $W(p_i)_{i=1}^{t}$, and an error occurs if and only if there is an error in each of the t channels. We denote this as a new channel $W_t(\mathbf{P})$, and the capacity of this channel is denoted by $\mathcal{C}(W(\mathbf{P}),t)$. Define $Q_i = \Pi_{j=1}^{i} p_j$ for $i \geq 1$. Then, for example, for the BSC we get $BSC_t(\mathbf{P}) = BSC(Q_t)$, and the capacity of this channel for $t \geq 1$ is $\mathcal{C}(BSC(\mathbf{P}),t) = \mathcal{C}(BSC(Q_t)) = 1-h(Q_t)$.

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 $PS \in \mathcal{P}_T$, $PS = PS((\alpha_1,t_1),\dots,(\alpha_\ell,t_\ell))$, over the channel sequence $W(\mathbf{P})$ is given by $\mathcal{D}(W(\mathbf{P}),PS) = \sum_{i=1}^{t}\alpha_i\mathcal{D}(W(\mathbf{P}),t_i)$, and the definition of the capacity is extended as follows $\mathcal{C}(W(\mathbf{P}),PS) = \sum_{i=1}^{t}\alpha_i\mathcal{C}(W(\mathbf{P}),t_i)$.

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

Problem 1 - Different probabilities. *Given a probabilities sequence* $\mathbf{P}$ *with a channel* $W \in \{BSC, BEC\}$*, 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* $\mathcal{D}(W(\mathbf{P}),PS) \leq D$*. In particular, find the value of* $F_2(W(\mathbf{P}),D,T) \triangleq \max_{PS\in\mathcal{P}_T:\mathcal{D}(W(\mathbf{P}),PS)\leq D}\{\mathcal{C}(W(\mathbf{P}),PS)\}$.

For $\mathbf{P} = (p_1,p_2,\dots)$ and $Q_i = \Pi_{j=1}^{i}p_j$, define $Y_t \triangleq \sum_{i=1}^{t-1}Q_i$ for $t \geq 1$ ($Y_1 = 0$), 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 9. *For the programming scheme* PS_t*, and* $\mathbf{P} = (p_1,p_2,\dots)$*, the following properties hold:*

1) *For* $t \geq 1$, $\mathcal{C}(BSC(\mathbf{P}),t) = 1-h(Q_t)$,
2) *For* $t \geq 1$, $\mathcal{C}(BEC(\mathbf{P}),t) = 1-Q_t$,
3) $\mathcal{C}(BSC(\mathbf{P}),-1) = \mathcal{C}(BEC(\mathbf{P}),-1) = 1$,
4) *For* $t \neq 0$,
$$\mathcal{D}(\mathbf{P},t) \triangleq \mathcal{D}(BSC(\mathbf{P}),t) = \mathcal{D}(BEC(\mathbf{P}),t) = 1+Y_t,$$
5) $\mathcal{D}(\mathbf{P},0) \triangleq \mathcal{D}(BSC(\mathbf{P}),0) = \mathcal{D}(BEC(\mathbf{P}),0) = 0$,

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 2.

Proposition 10. *Given a channel sequence* $W' = W(\mathbf{P})$*, an average delay* D*, and a programming scheme* PS*, it holds that* $\mathcal{C}(W',PS(W',D)) = \min\{D,\mathcal{D}(W',PS)\}\cdot\overline{\mathcal{C}}(W',PS)$.

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

Theorem 11. *For* $t \geq 1$ *and* $\mathbf{P}$*, the following holds:*

1) $\overline{\mathcal{C}}(BSC(\mathbf{P}),PS_t) \leq \overline{\mathcal{C}}(BSC(\mathbf{P}),PS_{t+1})$ *for all* $\mathbf{P}$ *such that* i, $0 \leq p_i \leq 0.5$,
2) $\overline{\mathcal{C}}(BEC(\mathbf{P}),PS_t) \leq \overline{\mathcal{C}}(BEC(\mathbf{P}),PS_{t+1})$ *if and only if* $p_{t+1} \leq \frac{Y_{t+1}}{Y_t+1}$.

By the previous theorem we conclude the following corollary, which its proof is similar to the proof of Corollary 5.

Corollary 12. *The solution for Problem 1 - Different probabilities is as follows:*

1) $F_2(BSC(\mathbf{P}),D,T) = \mathcal{C}(BSC(\mathbf{P}),PS_T(BSC(\mathbf{P}),D))$ *and it is obtained by* $PS_T(BSC(\mathbf{P}),D)$,
2) *if* $1 \geq p_1 \geq p_2 \geq p_3 \geq \cdots$ *then* $F_2(BEC(\mathbf{P}),D,T) = \mathcal{C}(BEC(\mathbf{P}),PS_T(BEC(\mathbf{P}),D))$ *and it is obtained by* $PS_T(BEC(\mathbf{P}),D)$

VI. 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 [13]. 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 $'?'$.

The average delay of programming a cell with $PS_{t,q}$ over the $BSC(p)$ does not depend on q, and hence equals to $\mathcal{D}(p,t)$. However the capacity is clearly influenced by the parameter q, and is denoted by $\mathcal{C}(BSC(p),t,q)$

Let p be the programming error probability. Then, the probability that a cell will be erroneous after $t-1$ programming attempts is p^{t-1}. Therefore, programming with $PS_{t,q}$ over $BSC(p)$ can be represented by a channel which combines the BSC and the BEC, where its capacity is well known [4, Porblem 7.13].

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 $0 \le q_i \le 1$, for all $1 \le i \le \ell$. Then, we define $\mathcal{C}(BSC(p),PS,\boldsymbol{q}) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{C}(BSC(p),t_i,q_i)$. That is, $\mathcal{C}(BSC(p),PS,\boldsymbol{q})$ is the capacity of $BSC(p)$ using the programming scheme PS with the parameter $\boldsymbol{q}$. Similarly, we define the average delay of the programming scheme PS for $BSC(p)$ using the parameter $\boldsymbol{q}$ as $\mathcal{D}(BSC(p),PS,\boldsymbol{q}) = \sum_{i=1}^{\ell} \beta_i \cdot \mathcal{D}(BSC(p),t_i,q_i)$. Note that $\mathcal{D}(BSC(p),PS,\boldsymbol{q}) = \mathcal{D}(BSC(p),PS)$ for all $\boldsymbol{q}$. For this model, Problem 1 will be formulated as follows.

Problem 1 - Combined channel. *Given a channel $BSC(p)$, 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 $\mathcal{C}(BSC(p),PS,\boldsymbol{q})$, under the constraint that $\mathcal{D}(BSC(p),PS,\boldsymbol{q}) \le D$. In particular, given $BSC(p)$, D, and T, find the value of*

$$F_3(BSC(p),D,T) \triangleq \max_{\mathcal{D}(BSC(p),PS,\boldsymbol{q}) \le D} \{\mathcal{C}(BSC(p),PS,\boldsymbol{q})\}.$$

For this generalization of the model we prove that the best programming scheme is $PS_{T,q}(BSC(p),D)$ for $q=1$ or $q=0$, where $PS_{T,q}(BSC(p),D)$ is defined in a similar way as in Equation (2). That is, the best programming scheme is 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.

Theorem 13. *Denote by $D' = \min\{\mathcal{D}(p,T),D\}$ then the solution for Problem 1 - Combined channel is $F_3(BSC(p),D,T) = \frac{D'}{\mathcal{D}(p,T)} \cdot \max\{1-h(p^t), 1-p^{t-1}\}$ obtained by $PS_{T,q}(BSC(p),D)$ for q=1 or q=0, respectively.*

This result can be intuitively explained as if $q=0$ then the last programming is just providing a complete verification for all the successful cells by substituting a question mark in all the erroneous cells. Thus, according to some threshold (p,t), we can either provide a complete verification for the already programmed cells ($q=0$) or try to reprogram again all the failed cells ($q=1$). The proof and details are available in the long version of this paper [10].

For each p we denote by t_p the smallest value of t, such that $h(p^t) \ge p^{t-1}$ (t_p may not be an integer). Since $h(px) \ge ph(x)$ for $0 \le x,p \le 0.5$, we conclude that for each $t \ge t_p$ it holds that $h(p^t) \ge p^{t-1}$. Thus, given p, it holds that $t \ge t_p$ if and only if $\mathcal{C}(BSC(p),t,0) = 1-p^{t-1} \ge 1-h(p^t) = \mathcal{C}(BSC(p),t,1)$. Let T be the last attempt to program. If $T \ge t_p$ then in the T-th attempt all the failed cells will be programmed with question marks. Otherwise, they will be reprogrammed again without verification. In Fig. 2 the values of t_p are presented in a graph, where the x-axis is p and the y-axis is t. The graph line is $f(p) = t_p$.

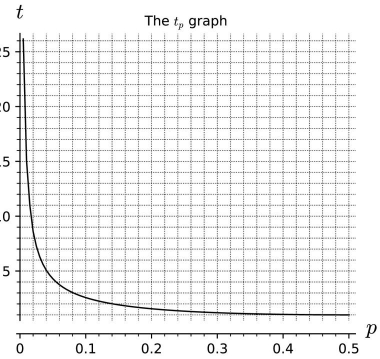

Fig. 2: The t_p graph.

Acknowledgment

The authors would like to thank Amos Lapidoth for helpful discussions. They are also indebted to the anonymous reviewers for their important comments.

References

[1] J. Bornholt et al., "A DNA-based archival storage system," *ASPLOS*, pp. 637–649, Atlanta, GA, Apr. 2016.

[2] C. Bunte and A. Lapidoth, "On the storage capacity of rewritable memories," *IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pp. 402–405, 2010.

[3] C. Bunte and A. Lapidoth, "Computing the capacity of rewritable memories," *Proc. Int. Symp. on Inform. Theory*, pp. 2512–2516, St. Petersburg, Jul. 2011.

[4] T.M. Cover and J.A. Thomas, "Elements of Information Theory," 1st Edition. New York: Wiley-Interscience, 1991.

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

[6] M. Franceschini, L. Lastras-Montano, T. Mittelholzer, and M. Sharma, "The role of feedback in rewritable storage channels [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 26, pp. 190–194, 2009.

[7] R. G. Gallager, "Information Theory and Reliable Communication," J. Wiley, NY, 1968.

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

[9] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[10] M. Horovitz, E. Yaakobi, E. En Gad, and J. Bruck, "Iterative Programming of Noisy Memory Cells," *arXiv:1901.03084*, 2019.

[11] A. Jiang and J. Bruck, "On the capacity of flash memories," *Proc. Int. Symp. on Inform. Theory and Its Applications*, pp. 94–99, 2008.

[12] A. Jiang and H. Li, "Optimized cell programming for flash memories," *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 914–919, 2009.

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

[14] L. Lastras-Montano, M. Franceschini, T. Mittelholzer, and M. Sharma, "Rewritable storage channels," *Proc. IEEE Int. Symp. of Inform. Theory*, pp. 7–10, Toronto, Jul. 2008.

[15] L. Lastras-Montano, M. Franceschini, T. Mittelholzer, and M. Sharma, "On the Capacity of Memoryless Rewritable Storage Channels," *IEEE Trans. Inform. Theory,* vol. 60, no. 6, pp. 3178–3195, Jun. 2014.

[16] H. T. Lue et al., "Study of incremental step pulse programming (ISPP) and STI edge effect of BE-SONOS NAND flash," *Proc. IEEE Int. Symp. on Reliability Physics*, vol. 30, no. 11, pp. 693–694, May 2008.

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

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

[19] L.G. Tallini, S. Al-Bassam, and B. Bose, "On the capacity and codes for the Z-channel," *Proc. IEEE Int. Symp. on Inform. Theory*, pp. 422, Lausanne, Switzerland, 2002.

[20] R. Venkataramanan, S. Tatikonda, L. Lastras-Montano, and M. Franceschini, "Rewritable storage channels with hidden state," *IEEE Journal on Selected Areas in Comm.*, Vol. 32, No. 5, pp. 815–824, May 2014.

[21] S. Verdu, "Channel capacity," Ch. 73.5 in the *Electrical Engineering Handbook, IEEE and CRC Press*, pp. 1671–1678, 1997.

[22] E. Yaakobi, A. Jiang, P.H. Siegel, A. Vardy, and J.K. Wolf, "On the parallel programming of flash memory cells," *Proc. IEEE Inf. Theory Workshop*, Dublin, Ireland, Aug.-Sep. 2010.