Adaptive Threshold Read Algorithms in Multi-level Non-Volatile Memories

Evyatar Hemo and **Yuval Cassuto**
Department of Electrical Engineering, Technion – Israel Institute of Technology
evyatarhemo@gmail.com, ycassuto@ee.technion.ac.il

***Abstract*— For an array of memory cells that are read by threshold measurements, we ask the question of how to choose the measurements in the read sequence to minimize the number of measurements before the array is fully read. We propose and study analytically and experimentally various adaptive read algorithms, and provide corresponding lower bounds on the average number of measurements. We show that new two-dimensional read algorithms improve over the best one-dimensional ones. We further adapt the read algorithms to the case where the cell levels are not uniformly distributed, as motivated by partially-erased memory arrays.**

I. Introduction

Solid-state storage technologies constantly grow in their storage densities, and are becoming the most attractive media for many applications. One popular way in which improved density is achieved is by increasing the number of levels to which a cell can be programmed. Unfortunately, using more levels per cell is not a "magic formula" to squeeze more bits into the same hardware, but rather an action with significant ramifications on the read/write performance and reliability of the device.

There is a significant body of known work on how to represent data in multi-level flash memories, so as to endow the storage device with different goodness features. These features include error correction, rewrite capability, and write optimization. The works in that area are so numerous and diverse, that we avoid the daunting task of listing a fair set of references. An important collection of works on flash data representation is offered by Jiang and Bruck in [1], while Peleato et al. specifically focus on minimizing read time on NVM in [2]. One direction that is less explored theoretically is the optimization of *reading* information from multi-level memory arrays – in particular, memory arrays that are read by threshold measurements applied to a group of cells in parallel. We choose this read problem as the topic of study reported in this paper. The motivation to consider the read problem comes from a concern that with threshold reads, continued growth in the number of levels will introduce a significant toll on the read performance. The driving idea of this work is that read algorithms that only optimize read time for the worst-case (across information contents) are suboptimal, since they fail to benefit from information contents that are "easier" to read.

To understand the problem at hand, suppose we have n memory cells with q discrete levels $\{0, \ldots, q-1\}$. The cells are read by applying a sequence of threshold measurements, each applied to the n cells in parallel, and returns n binary values of whether the cell levels are above or below the threshold. This type of measurement is natural for a variety of cell technologies used today, and likely those of the future. The motivation to read multiple cells in parallel using the same threshold comes from the assumption that switching and stabilizing reference threshold levels is a time-consuming task. Under this model of parallel threshold read, we develop a framework aimed at minimizing the number of threshold measurements required to read a memory array in its entirety. The algorithmic part of the framework includes algorithms that *adaptively* choose measurement thresholds so as to minimize the length of the read sequence. The analytic part of the framework derives the average performance of the algorithms, and provides lower bounds on the average numbers of measurements required by any algorithm.

As an example, in the case of $n = 4$ and $q = 8$ we intuitively feel that reading cell levels $(0, 1, 2, 1)$ is easier than reading the more spread levels $(0, 2, 4, 6)$. Indeed, when the 4 cells are read in parallel, multiple cells at the same level can benefit from a given threshold measurement, and fewer measurements will be needed. This intuitive feel is pursued in this paper, both algorithmically: minimizing the number of measurements, and analytically: calculating and bounding the average number of required measurements. The primary regime of operation to benefit from this work is when n is not much larger than q. This setup is the most interesting for the adaptive read problem, since the savings potential (over trivial read algorithms) is significant. This fact limits the applicability of the framework to current NAND flash technology, which uses an especially high level of parallelism. Nevertheless, it is plausible that future non-volatile memory technologies will work in a lower parallelism regime, due to technology limitations or cost issues. We also believe that threshold reading is a fundamental and interesting problem in general, worthy of the detailed study that follows.

In Section II the scope is on one-dimensional (1D) read algorithms, where the set of measured cells is fixed to the full block. In Section III we move to two-dimensional (2D) algorithms, where the measured cells can be chosen from the array with certain degrees of freedom. The results show that adaptive choice of threshold measurements can improve read performance over fixed predetermined read sequences. 2D algorithms are further shown to be superior over 1D algorithms, including over the 1D lower bound that any 1D algorithm must satisfy. In Section IV we analyze scenarios in which the values of the memory cells are not uniformly

Part of the results of this paper appeared in the proceedings of the 2013 International Symposium on Information Theory [3].

distributed. In particular, we study the case where one level has a higher probability of incidence compared to the remaining levels. This case is motivated by memory arrays where some portion of the cells are in "erased" state, thereby biasing the level distribution away from the uniform case. Adapting the read algorithms to a known non-uniform distribution is shown to significantly improve the read performance. In terms of prior work, the studied problem is related to the problem of adaptive sorting algorithms, but we have not found a simple way to directly apply existing knowledge to the threshold-read problem.

Unless stated otherwise, all log functions are base 2.

A. Storage capacity vs. reading speed in multi-level memories

Before delving into a detailed study, it would be beneficial to elaborate a little further on the context of the contributions. As already stated, increasing the number of levels in a cell has the obvious advantage of increasing the storage capacity. This is shown in Fig. 1 with the dashed line showing a capacity growth linear in $\log q$. Assuming readout by threshold-measurements, increasing the number of levels also increases the read time, since an elementary read algorithm requires $q-1$ threshold measurements to be applied to read the cell. The effect of this on the read speed is depicted by the solid curve of Fig. 1. The read speed is given as the ratio between the number of read bits and the number of measurements (by the elementary read algorithm). The objective of this work is to propose read algorithms that offer higher read speeds than those of Fig. 1, by cutting the measurement counts below those of the elementary read algorithm.

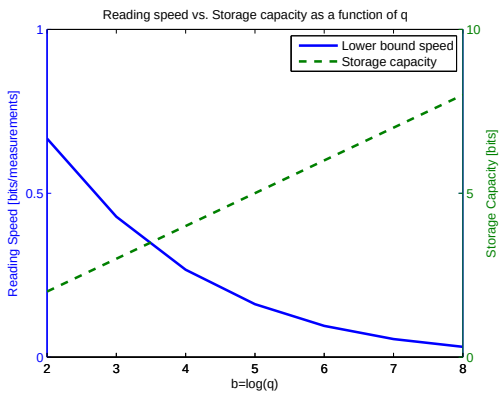

Figure 1. Storage capacity vs. reading speed tradeoff: storage capacity (dashed), and the read speed in bits/measurement (solid).

The above illustration of the capacity vs. speed tradeoff is a simplified one, because the read-speed performance will strongly depend on n, the number of measured cells.

B. Parallel threshold read model

We first give some formal definitions for the threshold-read model. Let the state of the storage cell be represented as a discrete **cell level** c, taken from the integer set $\{0,\ldots,q-1\}$.

Definition 1. *A* **threshold** τ *is an integer from the set* $\{1,\ldots,q-1\}$. *Given a threshold* τ, *a cell is said to be* **active** *with respect to* τ *if its cell level satisfies* $c \geqslant \tau$. *In the complementary case of* $c < \tau$, *the cell is said to be* **inactive** *with respect to* τ.

Note that this definition of threshold is equivalent to the standard notion of threshold being a real number in the interval $(\tau-1,\tau)$. For simplicity, we choose to make τ an integer, and define active cells as $\geqslant \tau$.

Definition 2. *A* **measurement** *is an operator acting on a set of cells* $\mathcal{S}$ *by applying a threshold of* τ, *and obtaining* $|\mathcal{S}|$ *binary values reflecting the activity of each cell in* $\mathcal{S}$ *with respect to* τ. *We denote the measurement as a vector* $M_\tau(\mathcal{S}) = (m_1,\ldots,m_{|\mathcal{S}|})$, *where* $m_i \in \{0,1\}$. $m_i = 1$ *represents an active cell with respect to* τ, *and* $m_i = 0$ *represents an inactive cell.*

The accumulated information on the cell levels after some sequence of measurements is represented by *uncertainty windows*:

Definition 3. *The* **uncertainty window** *of a cell is given as a pair of integers* $[L,U]$, *if it is known that* $L \leqslant c \leqslant U$.

It is straightforward to observe that for any threshold-measurement sequence, the level of every cell is known up to a set of consecutive integer values given by $L \leqslant c \leqslant U$. When $L = U$, the cell level c is completely known (no uncertainty).

II. Read Algorithms

In a standard storage setup involving non-volatile memories, we want to determine the cell levels of a block $\mathcal{N}$ of n memory cells by applying measurements to the cell block until all levels become known. The sequence of measurements applied to the cell block is allowed to be *adaptive*, i.e., the next measurement is selected based on the accumulated state of the cells measured thus far. In particular, the measurement sequence will stop once all cell levels have been determined, which may happen well before reaching the number of measurements needed for the worst-case cell-level combination. We assume throughout the section that all measurements act on the full block of cells, hence $\mathcal{S} = \mathcal{N}$. This assumption will be lifted in subsequent sections.

A. Sequential scan

The simplest way to determine the cell levels of all cells is by a sequential scan – applying measurements starting from $\tau = 1$ upward, and stopping when all cell levels are fully determined. In the worst case, all $q-1$ possible τ values will be used, but an earlier stop is possible if none of the n cells in the block currently stores the upper levels. Early stopping of sequential scan is the simplest form of adaptivity employed in read algorithms, improving the average read time over the non-adaptive scan of all $q-1$ thresholds.

We now give an expression for the expected number of measurements, assuming the cell levels are uniformly and independently distributed over the set $\{0,\ldots,q-1\}$.

Proposition 1. *The expected number of measurements needed for sequential-scan read, assuming uniform level distribution, is given by*

$$T(n,q) = (q-1) - \sum_{k=1}^{q-2} \left(\frac{k}{q}\right)^n. \tag{1}$$

Proof: If all the cells in the block have levels in $\{0, ..., k-1\}$, for some $k \leqslant q-1$, then k measurements are clearly sufficient. This is because the k-th measurement, having $\tau = k$, gives $M_k(\mathcal{N}) = (0, \ldots, 0)$ (all cells inactive), making all higher-threshold measurements redundant. Therefore, for every $k \leqslant q-2$, $\Pr[\#meas. \leqslant k] = (k/q)^n$, and for $k = q-1$, we trivially get $\Pr[\#meas. \leqslant q-1] = 1$. For $k \leqslant q-2$, if all levels fall in $\{0, ..., k-1\}$ but not in $\{0, ..., k-2\}$, then k measurements are both sufficient and necessary, and the number of *saved* measurements is $q-1-k$. The expected number of saved measurements is thus given by

$$E[q-1-\#meas.] = \sum_{k=1}^{q-2}(q-1-k)\left[\left(\frac{k}{q}\right)^n - \left(\frac{k-1}{q}\right)^n\right]. \quad (2)$$

Splitting the sum and shifting the summation indices of the second sum, (2) becomes

$$\sum_{k=1}^{q-2}(q-1-k)\left(\frac{k}{q}\right)^n - \sum_{k=1}^{q-3}(q-2-k)\left(\frac{k}{q}\right)^n = \sum_{k=1}^{q-2}\left(\frac{k}{q}\right)^n. \quad (3)$$

Subtracting the right-hand side of (3) from $q-1$ we get the expected number of measurements in the claim (1). ■

From the expression for $T(n,q)$ in (1), it is observed that as n grows, the expected number of measurements tends to the worst case of $q-1$. This stems from the fact that as n grows, it becomes improbable that no cell stores the upper levels.

B. Binary search

A potentially better read-algorithm than sequential scan is the binary search. For reading a single cell level ($n = 1$), it is clear that the binary search, requiring $\log q$ measurements, is optimal. The binary search is the basis of a widely used analog to digital converter (ADC) family called *successive approximation register (SAR)*[4]-[7]. To use the binary search for multiple cells ($n > 1$), the following simple extension of the algorithm is needed. Recall the cell uncertainty window from Definition 3 denoted by $[L, U]$. Initially, each cell has an uncertainty window of $[L = 0, U = q-1]$. After the first binary-search measurement $M_{q/2}(\mathcal{N})$ (we assume for simplicity that q is a power of two), the cells that are active will have an uncertainty window of $[q/2, q-1]$; those that are inactive will have $[0, q/2-1]$. The n-cell binary-search algorithm proceeds by successively cutting by half level intervals that overlap with the uncertainty window of at least one cell in $\mathcal{N}$. It stops when all uncertainty windows are of size 1 ($L = U$). We give a formal specification of the algorithm in recursive presentation. In the sequel we denote by L_i and U_i the upper and lower limits, respectively, of the uncertainty window of the i-th cell in $\mathcal{N}$. Complete measurement of all the cells in $\mathcal{N}$ is achieved by calling BinarySearch$(\mathcal{N}, 0, q-1)$.

Algorithm 1. *BinarySearch*$(\mathcal{N}, L, U)$
if $L = U$ *return*
$\tau = (L+U+1)/2$
$\boldsymbol{m} = M_\tau(\mathcal{N})$
// update uncertainty windows
For all i *with* $m_i = 0$, *set* $U_i = \min(U_i, \tau-1)$
For all i *with* $m_i = 1$, *set* $L_i = \max(L_i, \tau)$
// recursive calls
if $\exists i : m_i = 0, U_i \geqslant L$ *then BinarySearch*$(\mathcal{N}, L, \tau-1)$
if $\exists i : m_i = 1, L_i \leqslant U$ *then BinarySearch*$(\mathcal{N}, \tau, U)$

A recursive BinarySearch call in Algorithm 1 is invoked (if and) only if there is a cell in the corresponding sub-interval. If there is no cell i whose uncertainty window overlaps with $[L, \tau-1]$, then the call BinarySearch$(\mathcal{N}, L, \tau-1)$ is skipped. Similarly, if there is no cell i whose uncertainty window overlaps with $[\tau, U]$, then the call BinarySearch$(\mathcal{N}, \tau, U)$ is skipped. These skipped intervals result in saved measurements. In the extreme case of $n = 1$, all measurements return (degenerate, size 1) all-0 or all-1 vectors, and it is always the case that only one sub-interval is chosen in the recursion. We now turn to analyze the expected number of measurements applied by the BinarySearch algorithm.

Proposition 2. *Let $F(n, \log q)$ be the expected number of measurements needed for binary-search read, assuming uniform level distribution. Then $F(n, \log q)$ can be calculated by the recursive formula*

$$F(n,l) = \sum_{i=0}^{n}\frac{\binom{n}{i}}{2^n}\left[1 + F(i, l-1) + F(n-i, l-1)\right], \quad (4)$$

where $F(n,l) = 0$ if either $n = 0$ or $l = 0$.

An explicit analytic expression for $F(n,l)$ is given by

$$F(n,l) = \sum_{k=0}^{l-1} 2^k\left[1 - \left(1 - \frac{1}{2^k}\right)^n\right]. \quad (5)$$

Proof: Proving the recursive formula in (4) is straightforward. Each entry to BinarySearch in Algorithm 1 applies one measurement, and calls BinarySearch again recursively to measure the i inactive cells at the lower sub-interval, and the $n-i$ active cells at the upper sub-interval. Note that when either i or $n-i$ are zero, one recursive call is skipped, which is captured by setting $F(0,l) = 0$. The probability of an $i, n-i$ split between active and inactive cells is determined to be $\binom{n}{i}/2^n$ by the uniform distribution.

The closed-form expression in (5) is now proved by induction on l. For the induction base we verify that (5) gives $F(n,0) = 0$, as required by the initial conditions. We now assume that (5) is correct for $l-1$, and prove it for l. First we simplify (4) using symmetry to get

$$F(n,l) = 1 + \sum_{i=0}^{n}\frac{\binom{n}{i}}{2^{n-1}}F(i, l-1). \quad (6)$$

As the induction hypothesis we substitute (5), with argument $l-1$, in the right-hand side of (6)

$$F(n,l) = 1 + \sum_{i=0}^{n}\frac{\binom{n}{i}}{2^{n-1}}\left[\sum_{k=0}^{l-2}2^k\left[1 - \left(1 - \frac{1}{2^k}\right)^i\right]\right].$$

Exchanging summation order we get

$$F(n,l) = 1 + \frac{1}{2^{n-1}}\sum_{k=0}^{l-2}2^k\sum_{i=0}^{n}\binom{n}{i}\left[1 - \left(1 - \frac{1}{2^k}\right)^i\right]. \quad (7)$$

The inner sum can now be written as

$$\sum_{i=0}^{n}\binom{n}{i}\left[1-\left(1-\frac{1}{2^k}\right)^i\right]=2^n-\left(2-\frac{1}{2^k}\right)^n=2^n\cdot\left[1-\left(1-\frac{1}{2^{k+1}}\right)^n\right]. \tag{8}$$

Substituting (8) back in (7) now gives

$$F(n,l)=1+\sum_{k=0}^{l-2}2^{k+1}\left[1-\left(1-\frac{1}{2^{k+1}}\right)^n\right]=\sum_{k=0}^{l-1}2^k\left[1-\left(1-\frac{1}{2^k}\right)^n\right],$$

proving the induction step for (5). The last equality is obtained by shifting the summation index and inserting the 1 into the sum.

■

C. Lower bound on the average number of measurements

For the purpose of evaluating the efficiency of the simple read algorithms of Sections II-A and II-B, we now derive a lower bound on the average number of measurements. Given a size-n block $\mathcal{N}$ of q-ary cells, we wish to find a lower bound $LB(n,q)$ defined in the following.

Definition 4. *$LB(n,q)$ is called a* **lower bound** *if any read algorithm for n q-ary cells requires on average at least $LB(n,q)$ measurements.*

To obtain such a bound, the key observation we make is that given an assignment of levels to the n cells $(c_1,\ldots,c_n)$, any read algorithm must apply a measurement in every threshold level that appears as one of the c_i, and also in every threshold level that is immediately above one of the c_i. If at least one of these two measurements is missing, then the corresponding cell remains with an uncertainty window of size at least two. More formally, we have the following definitions.

Definition 5. *Given a vector of cell levels $\boldsymbol{c}=(c_1,\ldots,c_n)$, with $c_i\in\{0,\ldots,q-1\}$, define the* **incidence set** *as the set $\mathcal{I}(\boldsymbol{c})=\{s\in\{1,\ldots,q-1\}|\exists i,\ c_i=s\}$. The* **shifted incidence set** *is defined as $\mathcal{I}^*(\boldsymbol{c})=\{s\in\{1,\ldots,q-1\}|\exists i,\ c_i+1=s\}$.*

Therefore, we have the following.

Proposition 3. *For a given cell-level vector $\boldsymbol{c}$, a lower bound on the number of measurements is given by $|\mathcal{I}(\boldsymbol{c})\cup\mathcal{I}^*(\boldsymbol{c})|$.*

Observe that $|\mathcal{I}(\boldsymbol{c})\cup\mathcal{I}^*(\boldsymbol{c})|\leqslant\min(2n,q)$.

Example 1. *For the following $q=8$, $n=4$ cell-level vector $\boldsymbol{c}=(2,2,4,5)$, we have $\mathcal{I}(\boldsymbol{c})=\{2,4,5\}$ and $\mathcal{I}^*(\boldsymbol{c})=\{3,5,6\}$. Since $\mathcal{I}(\boldsymbol{c})\cup\mathcal{I}^*(\boldsymbol{c})=\{2,3,4,5,6\}$, the lower bound is 5.*

In order to obtain an analytic lower bound for the *average* number of measurements, we need to find the expectation of $|\mathcal{I}(\boldsymbol{c})\cup\mathcal{I}^*(\boldsymbol{c})|$ over uniformly distributed vectors $\boldsymbol{c}\in\{0,\ldots,q-1\}^n$. As seen in Proposition 3, the lower bound for a particular $\boldsymbol{c}$ depends on both the size of $\mathcal{I}(\boldsymbol{c})$ (how many levels appear), and the overlap between $\mathcal{I}(\boldsymbol{c})$ and $\mathcal{I}^*(\boldsymbol{c})$ (how many levels in the union serve as both incident and shifted). It can be seen that

$$|\mathcal{I}(\boldsymbol{c})\cup\mathcal{I}^*(\boldsymbol{c})|=|\mathcal{I}(\boldsymbol{c})|+L(\boldsymbol{c}),$$

where $L(\boldsymbol{c})$ is the number of runs of consecutive levels in $\mathcal{I}(\boldsymbol{c})$. In Example 1, for $\boldsymbol{c}=(2,2,4,5)$ we have $L(\boldsymbol{c})=2$, since the set $\mathcal{I}(\boldsymbol{c})=\{2,4,5\}$ can be split into two runs: $\{2\}$ and $\{4,5\}$. The number of runs captures the number of necessary measurements, because in each run only the last level contributes an element to $\mathcal{I}^*$ not already appearing in $\mathcal{I}$.

To calculate the distribution of $L(\boldsymbol{c})$, we first regard the set $\mathcal{I}(\boldsymbol{c})$ as a length-q binary indicator vector whose i-th entry is 1 if $i\in\mathcal{I}(\boldsymbol{c})$. We then define the combinatorial object $D(q,l,L)$ to be the number of length-q binary vectors that have l ones falling into exactly L runs of consecutive coordinates. To handle the special extreme cases of levels 0 and $q-1$, we further refine $D(q,l,L)$ to $D_0(q,l,L)$, $D_1(q,l,L)$ and $D_2(q,l,L)$, where $D_j(q,l,L)$ is the number of (q,l,L) vectors that have a one on j of the locations 0 and $q-1$. This refinement is needed because the elements 0 and $q-1$ are special in that a run that contains them necessitates one fewer measurement than a run that does not touch the edges. Based on [8], we now write the closed-form expressions for $D_j(q,l,L)$.

$$D_0(q,l,L)=\binom{l-1}{L-1}\binom{q-l-1}{L},\ D_1(q,l,L)=2\binom{l-1}{L-1}\binom{q-l-1}{L-1},$$
$$D_2(q,l,L)=\binom{l-1}{L-1}\binom{q-l-1}{L-2}+\Delta[l=q;L=1].$$

The function $\Delta[]$ is 1 when all of its arguments are true and 0 otherwise. The extra term of $\Delta[l=q;L=1]$ covers the case where all q locations are ones, where we have a single run of ones that touches both edges. We are now ready to present a lower bound on the average number of measurements.

Theorem 4. *A lower bound on the average number of measurements given uniformly distributed cell levels is given by*

$$LB(n,q)=\frac{1}{q^n}\sum_{k=1}^{n}k!\cdot S(n,k)\cdot\sum_{L=1}^{k}\sum_{j=0}^{2}D_j(q,k,L)\cdot(k+L-j),$$

where $S(n,k)$ is the Stirling number of the second kind [9].

Proof: Every vector $\boldsymbol{c}\in\{0,\ldots,q-1\}^n$ can be uniquely obtained by choosing a size-k set $\mathcal{I}(\boldsymbol{c})$, and then applying a surjection from the n-set of coordinates to the k-set $\mathcal{I}(\boldsymbol{c})$. It is well known [10] that the number of surjections from an n-set to a k-set equals $k!S(n,k)$. The number of necessary measurements depends only on k and the number of runs in the set $\mathcal{I}(\boldsymbol{c})$. The two inner sums count all choices of size-k sets $\mathcal{I}(\boldsymbol{c})$, and weight each choice with its corresponding number of necessary measurements $k+L-j$. The number of necessary measurements does not depend on the particular surjection applied to $\boldsymbol{c}$, and hence the numbers of surjections appear at the outer sum. The overall sum, normalized by the number of vectors q^n, gives the expected number of necessary measurements given the uniform distribution on $\boldsymbol{c}$. It is clear that any read algorithm will have at least the number $k+L-j$ of necessary measurements on every input $\boldsymbol{c}$, and therefore on average must apply no less than $LB(n,q)$ measurements. ■

To better understand the proof of Theorem 4, we give an example of a mapping between vectors $\boldsymbol{c}\in\{0,\ldots,q-1\}^n$ and incidence sets $\mathcal{I}$.

Example 2. *Suppose $n=3$, $q=8$, and we have the incidence set $\mathcal{I}=\{2,5\}$ (with size $k=2$). The vectors $\boldsymbol{c}$ that map to*

$\mathcal{I}$ *are the* 6 *vectors* $(2,2,5)$, $(2,5,2)$, $(5,2,2)$, $(5,5,2)$, $(5,2,5)$, $(2,5,5)$. *Substituting* $n = 3$, $k = 2$ *in* $k!S(n,k)$ *indeed gives* 6. *All these* $\mathbf{c}$ *vectors will have the same number of necessary measurements, which depends solely on* $\mathcal{I}$.

1) Asymptotics of $LB(n,q)$*:* As q becomes large compared with n, the probability that two cells will store the same level or a pair of adjacent levels approaches 0. In addition, the extreme levels 0 or $q-1$ will also have a low probability of incidence. Therefore, in the limit of large q we have

$$\lim_{q\to\infty} LB(n,q) = 2n. \tag{9}$$

Even in this extreme case of no runs longer than 1, the lower bound does not exclude read algorithms with a constant (in q) number of measurements. This fact motivates a further study of read algorithms that adapt to the instantaneous contents of the memory cells. Such adaptation will allow reducing the average measurement counts well below the $q-1$ measurements required by a non-adaptive reader.

D. Performance comparison

To summarize the section, we take the analytic expressions for the average measurement counts of two read algorithms: sequential scan ($T(n,q)$ in Section II-A) and binary search ($F(n,\log q)$ in Section II-B), and plot their values in comparison with the lower bound ($LB(n,q)$ in Section II-C). The case of fixed $n = 4$ is shown in Fig. 2. The average numbers of measurements are plotted as a function of $\log(q)$.

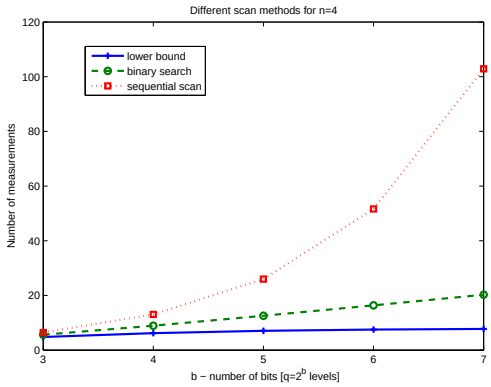

Figure 2. Analytic average measurement counts for $n = 4$: sequential scan (squares), binary search (circles), and the lower bound (crosses).

It is observed in Fig. 2 that the read complexity of sequential search grows linearly with q. In contrast, binary search grows more gracefully. The growth of binary search is roughly linear in $\log(q)$, generalizing the slope-1 linear growth of the standard $n = 1$ binary search. The lower bound grows even slower than binary search, converging to $2n = 8$ as predicted by (9).

Another interesting case to examine is when n and q grow together, while maintaining a fixed ratio. Fig. 3 shows the results for $n = q/2$. Here the three curves follow a similar shape, but with widening gaps as q grows. The fixed-ratio case is important because it makes n small enough to have "easy" read instances that improve the average counts (when

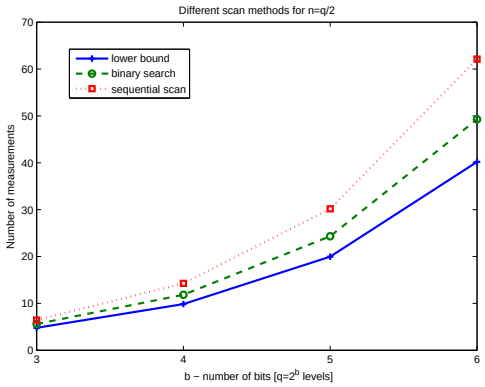

Figure 3. Analytic average measurement counts for $n = q/2$.

n is very large, with high probability all q levels will be used, and sequential scan is optimal), but n is also large enough to motivate advanced read algorithms (when n is very small, e.g. $n = 1$, the binary search is likely close to optimal). Our objective in the remainder of the paper is to improve over binary search using more advanced algorithms.

III. Two-Dimensional Read Algorithms

In order to improve on the average number of measurements, we will move to a more realistic storage setup where the cells are organized as a two-dimensional (2D) array, and each measurement is applied to a subset of the memory cells. In the 2D problem, the memory block $\mathcal{N}$ is an array of q-ary cells with dimensions $m \times n$, hence $|\mathcal{N}| = mn$. The set of cells $\mathcal{S}$ to which a measurement $M_\tau(\mathcal{S})$ is applied is no longer the full set $\mathcal{N}$, but a set of size n. Setting the measurement set size to n allows comparisons between the proposed 2D algorithms and using the 1D algorithms of Section II row-by-row. We start the discussion of 2D read algorithms with a motivating example.

Example 3. *Suppose we read the following* 2×2, $q = 8$ *array row-by-row.*

1	*2*
0	*3*

Then from Proposition 3 *we know that we need for the top row at least* 3 *measurements:* $\mathcal{I}((1,2)) \cup \mathcal{I}^*((1,2)) = \{1,2,3\}$, *and for the bottom row at least* 3 *measurements as well:* $\mathcal{I}((0,3)) \cup \mathcal{I}^*((0,3)) = \{1,3,4\}$. *In total we need* 6 *measurements.*

Alternatively, if we can choose whether to measure a row **or** *a column, it is possible to reduce the number of measurements to* 4. *We first measure the top row with* $\tau = 2$, *then we measure the left column with* $\tau = 1$. *At this point the* 1 *and* 0 *in the left column are fully known, and the* 2 *on the top right is known to be* $\geqslant 2$. *Now we measure the right column with* $\tau = 3$ *and* $\tau = 4$, *after which the* 2 *and* 3 *on the right are also known.*

By showing a gap in the lower bound on the number of measurements, Example 3 shows that a 2D algorithm can *in principle* improve over a 1D algorithm. However, it is still not clear how to reduce the number of measurements

algorithmically using 2D flexibility. In the remainder of the section we address the algorithmic 2D read problem. We first define the general optimization problem formally.

Problem 5. *Given a $m \times n$ memory array $\mathcal{N}$, find a minimal-length sequence of measurements $M_{\tau_1}(\mathcal{S}_1), M_{\tau_2}(\mathcal{S}_2), \ldots$ that reads all the cells in $\mathcal{N}$. The sets $\mathcal{S}_j$ are of size n, and their selection is governed by some prescribed access model. The measurement sequence is adaptive, i.e., the choice of $\mathcal{S}_j$ and τ_j may depend on the outcomes of preceding measurements.*

A. ANDF algorithm

In a $m \times n$ array $\mathcal{N}$, suppose we adopt the most flexible access model whereby at each step j the set $\mathcal{S}_j$ can be any n cells out of $\mathcal{N}$. For this access model we propose the *any n degrees of freedom (ANDF)* algorithm to adaptively select a sequence of measurements. The core of the ANDF algorithm is a criterion to choose the n cells and the threshold τ_j, based on the current uncertainty state of the array cells. The objective of the implemented criterion is to read $\mathcal{N}$ at full with a short measurement sequence.

The criterion that we choose for the ANDF algorithm is minimization of the *expected uncertainty* after the measurement. The idea behind this criterion (defined shortly), is to greedily choose the measurement that makes the largest step toward eliminating the uncertainty in the array.

Definition 6. *The* **uncertainty** Ω *of a cell with uncertainty window $[L, U]$ is defined as $\Omega = \log(U - L + 1)$.*

The uncertainty of a cell ranges from $\log q$ initially (before any measurement) to 0 when $L = U$ (level fully determined). Let $[L_i, U_i]$ be the uncertainty window of cell i, with its uncertainty value denoted Ω_i. It is easy to see that if $L_i < \tau \leqslant U_i$, then after the measurement the uncertainty window of cell i will be $[L_i, \tau - 1]$ if $c_i < \tau$ and $[\tau, U_i]$ if $c_i \geqslant \tau$. Hence we can find the expected uncertainty of cell i after a measurement with threshold τ to be

$$H_i(\tau) = \Pr(c_i < \tau)\log(\tau - L_i) + \Pr(c_i \geqslant \tau)\log(U_i - \tau + 1) = \frac{\tau - L_i}{U_i - L_i + 1}\log(\tau - L_i) + \frac{U_i - \tau + 1}{U_i - L_i + 1}\log(U_i - \tau + 1), \quad (10)$$

where (10) follows by a simple substitution of the uniform distribution into the probabilities above. Another way to write $H_i(\tau)$ is now given using the binary entropy function

$$H_i(\tau) = \Omega_i - h\left(\frac{\tau - L_i}{U_i - L_i + 1}\right),$$

where $h(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. Our objective is to maximize the *uncertainty reduction* by the selected measurement, hence the criterion for choosing $\mathcal{S}$ and a corresponding τ by ANDF is set to be

$$\underset{\mathcal{S},\tau}{\operatorname{argmax}} \sum_{j\in\mathcal{S}} \Omega_j - H_j(\tau) = \underset{\mathcal{S},\tau}{\operatorname{argmax}} \sum_{j\in\mathcal{S}} h\left(\frac{\tau - L_j}{U_j - L_j + 1}\right). \quad (11)$$

Note that this is a generalization of the 1D binary search algorithm, for which $\mathcal{S} = \mathcal{N}$ is fixed and τ is chosen as the mid point between L_i and U_i to reduce the uncertainty by 1 bit.

An efficient implementation of the selection criterion (11) is given in the formal specification of the ANDF algorithm in Algorithm 2.

Algorithm 2. *ANDF($\mathcal{N}$)*

```
while ∃i : L_i ≠ U_i
    for k = 1 to q − 1 // all thresholds
        for i = 1 to mn // all cells in N
            UncRed(i, k) = Ω_i − H_i(k)
        end
        S^k = top_n(UncRed(:, k)) // top n cells
    end
    k* = argmax_k Σ_{j∈S^k} UncRed(j, k)
    m = M_{k*}(S^{k*})
    // update uncertainty windows
    For all j with m_j = 0, set U_j = min(U_j, k* − 1)
    For all j with m_j = 1, set L_j = max(L_j, k*)
end
```

The matrix *UncRed* holds the values of the expected uncertainty reduction for each cell i and threshold k. The function top_n picks the n cells with the largest $\Omega_i - H_i(k)$ for a given k. Finally, the optimal threshold $k^\star$ is the k whose top_n cells have the largest sum of uncertainty reductions. Altogether, in each iteration of the while loop the algorithm finds the optimal threshold according to the criterion (11). The complexity of each selection iteration is roughly mnq, since the top_n selection can be done in complexity $O(n\log(mn))$ using a heap data structure, which is lower than the mn operations in the cell loop.

B. CRDF algorithm

The complete flexibility endowed to ANDF in choosing the n measured cells may not be practical in real memory systems. Therefore, we are also interested in studying read algorithms with a much more restricted access model, which may be more realistic to implement. We now describe such an algorithm for the special case where $\mathcal{N}$ is a square $n \times n$ array. In that case, we let the read algorithm choose a size n set $\mathcal{S}$ that is either a *row or a column* of the array. It is slightly more flexible than the 1D access model that only works on rows. We define the *columns and rows degrees of freedom (CRDF)* algorithm to be the algorithm that maximizes the uncertainty reduction among all choices of row/column and threshold. So CRDF may be regarded as a variant of ANDF whose selection sets are restricted to rows or columns.

Algorithm 3. *CRDF($\mathcal{N}$)*

```
while ∃i : L_i ≠ U_i
    for k = 1 to q − 1 // all thresholds
        for i = 1 to mn // all cells in N
            UncRed(i, k) = Ω_i − H_i(k)
        end
(*)     S^k = top_row_col(UncRed(:, k)) // top n cells in row/col
    end
    k* = argmax_k Σ_{j∈S^k} UncRed(j, k)
    m = M_{k*}(S^{k*})
    // update uncertainty windows
```

For all j with $m_j = 0$, set $U_j = \min(U_j, k^\star - 1)$
For all j with $m_j = 1$, set $L_j = \max(L_j, k^\star)$
end

Note that the only change between ANDF in Algorithm 2 to CRDF in Algorithm 3 is in line (*) where top_n was replaced by top_row_col, which finds the top n cells conditioned to be all in the same row or column.

C. Lower bound in the 2D case

In Section II-C we derived an analytic lower bound on the average number of measurements for the 1D case ($\mathcal{S} = \mathcal{N}$). We now want to derive a similar bound for the 2D case, where the size of $\mathcal{N}$ is $N = mn$, and each measurement set $\mathcal{S}$ is of size n. The lower bound will be general in the sense that it does not limit the flexibility of choosing $\mathcal{S}$ (hence in particular it also applies to ANDF). The main idea in the derivation is to repeat the analysis of incidence sets as in the 1D bound, but this time adding the constraint that a single measurement does not cover the entire population of cells in a given level, but only up to n of them.

Theorem 6. *Given the uniform level distribution, a lower bound on the average number of n-cell measurements required to read an array of N cells is given by*

$$LB2D\,(N,q,n) = LB\,(N,q) + \\ +(q-1)\cdot\sum_{d=n+1}^{N}\left\lfloor\frac{d-1}{n}\right\rfloor\cdot\binom{N}{d}\cdot\left(\frac{2}{q}\right)^{d}\cdot\left(\frac{q-2}{q}\right)^{N-d}, \qquad (12)$$

where $LB(\cdot,\cdot)$ is given in Theorem 4.

Proof: In the 1D lower bound (Theorem 4), a level c occupied by at least one cell necessitated measurements with thresholds c, $c+1$. Conversely, a single measurement τ sufficed to cover one of the two measurements for *all* cells that are at levels τ and $\tau-1$. Now in the 2D case, a measurement can only cover up to $n < N$ cells in the two adjacent levels τ and $\tau-1$. Therefore, when more than n cells store a pair of adjacent levels, additional measurements need to be applied beyond the 1D lower bound. The number of excess measurements is $\lfloor(d-1)/n\rfloor$, where d is the number of cells in the pair of adjacent levels. For a given pair of levels τ, $\tau-1$, the probability that d cells occupy them is given by the binomial distribution in the three rightmost multiplicative terms of (12). Finally, the multiplicative factor $q-1$ adds the excess measurements for all pairs of adjacent levels. Note that while the $q-1$ adjacent-level pairs are overlapping, this does not result in over counting, the reason being that a level c with high cell occupancy actually adds to the 2D lower bound twice: once for measurement $\tau = c$ and once for $\tau = c+1$. ■

D. Results

The motivating Example 3 showed that in principle increased measurement flexibility can speed up the read time of the array. We now want to evaluate this advantage quantitatively in an experimental setup. In particular, we want to quantify the advantage of the row/column CRDF algorithm over row-by-row 1D reading, as well as the gap between CRDF to the maximally flexible ANDF. The performance of the 2D algorithms is now evaluated by simulations. A 4×4 memory array with uniformly distributed level assignment was read by both the CRDF and ANDF algorithms 1000 times, providing the results presented in Fig. 4. The average number of measurements, normalized by $n = 4$, of CRDF and ANDF are marked as a function of $\log q$. The normalization by n allows to compare the results with the 1D binary search and lower bound, also marked on the same plot. We first observe

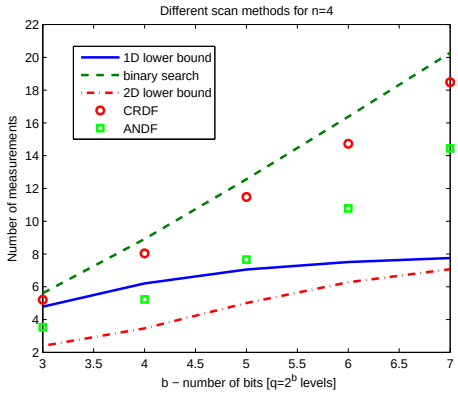

Figure 4. Simulated average measurement counts for 4×4 arrays (normalized by n): 1D lower bound (solid), binary search (dashed), 2D lower bound (semi-dashed), CRDF (circle markers), and the ANDF (square markers).

that CRDF improves over the 1D binary search for all q, with a growing gap. ANDF is clearly superior to CRDF, indicating the benefits of increased measurement flexibility. Note that as previously observed in Example 3, the 1D lower bound (solid line) does not apply to the 2D algorithms, since the necessary measurements within a row can be shared by multiple rows, e.g. in a column measurement. Indeed ANDF outperforms the 1D lower bound for low q.

The results for a fixed ratio $n = q/2$ are shown in Fig. 5.

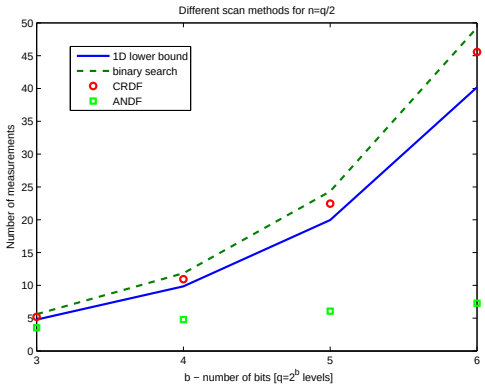

Figure 5. Simulated average measurement counts (normalized by n) for $n = q/2$.

As we can see, for $n = q/2$ the ANDF significantly outperforms both CRDF and the lower bound, which is a motivation to implement more flexible measurement-set selection.

IV. Non-uniform Distribution of Memory Levels

Up until now it was assumed that the level c_i of each memory cell is uniformly distributed in $\{0,\ldots,q-1\}$. However, it may be the case that the level distribution is not uniform, and that we have some a priori knowledge of this distribution. A plausible scenario is when the 0 level occurs more frequently than other levels in an array that is only partially programmed. Another case is when some multi-write code is employed, changing the level distribution after each write (the lower levels are more frequent in the early writes, gradually drifting toward the upper levels with each write). For such non-uniform setups we propose to adapt the ANDF and CRDF algorithms from the previous section to accommodate non-uniform level distributions.

A. Expected uncertainty for non-uniform levels

We choose to pursue a simple non-uniform level distribution where level 0 is the most probable of all levels, while the remaining levels $\{1,\ldots,q-1\}$ have the same probability. This should be regarded mainly as an illustrative example, which can be generalized to other distributions. Given a cell with uncertainty window $[0,U]$, for some U, our objective is to derive the expected uncertainty after a measurement with threshold $\tau\in\{1,\ldots,U\}$. If the cell has uncertainty window $[L,U]$ with $L>0$, we revert to the uniform case given in (10). However, for uncertainty windows that contain the 0 level, we need to derive the expected uncertainty in light of the more probable 0 level.

Given an uncertainty window of $[0,U]$, the probability that the cell is at level 0 is

$$\Pr(c=0)=\frac{\alpha}{U+1}, \tag{13}$$

for some real number $\alpha\geqslant 1$, which depends on the level distribution (through the a priori probability of the 0 level). The value of α also depends on U, but for each U it can be easily calculated from the known level distribution. The remaining U levels have the same probability, which is given by

$$\Pr(c=k,k\neq 0)=\frac{U+1-\alpha}{U\,(U+1)}, \tag{14}$$

such that the probabilities sum to 1 within the $[0,U]$ window. Note that $\alpha=1$ corresponds to the uniform distribution.

The non-uniform distribution within $[0,U]$ means that the uncertainty Ω of this window is no longer $\log(U+1)$ as stated in Definition 6, but rather an expression that depends on α. To generalize the uncertainty to the non-uniform case, we redefine it as the *entropy of the level distribution* in $[0,U]$. This refined definition is consistent with Definition 6, since the entropy of the uniform distribution is indeed $\log(U+1)$. We now calculate the non-uniform uncertainty as a function of the distribution parameter α.

Proposition 7. *Given the distribution parameter α, the* **non-uniform uncertainty** *of a cell with uncertainty window $[0,U]$ is given by*

$$\Omega^*\left([0,U]\right)=\log\left(\frac{U+1}{\alpha}\right)+\frac{U+1-\alpha}{U+1}\log\left(\frac{\alpha U}{U+1-\alpha}\right). \tag{15}$$

Proof: The statement follows by calculating the standard distribution entropy with the probabilities in (13) and (14). ■

With an expression for the non-uniform uncertainty, we can now derive the expected uncertainty after a measurement with threshold τ.

Proposition 8. *The expected uncertainty of a cell with $[0,U]$ uncertainty window after applying a threshold τ is given by*

$$\begin{aligned}H(\tau)&=\frac{\alpha}{U+1}\log\left(1+\frac{(U+1-\alpha)(\tau-1)}{\alpha U}\right)\\&+(\tau-1)\frac{U+1-\alpha}{U(U+1)}\log\left(\tau-1+\frac{\alpha U}{U+1-\alpha}\right)\\&+\frac{(U+1-\alpha)\,(U-\tau+1)}{U\,(U+1)}\log(U-\tau+1).\end{aligned} \tag{16}$$

Proof: By definition

$$\begin{aligned}H(\tau)&=\Pr\left(c<\tau\right)\cdot H\left(\tau\mid c<\tau\right)\\&+\Pr\left(c\geqslant\tau\right)\cdot H\left(\tau\mid c\geqslant\tau\right),\end{aligned} \tag{17}$$

where $H(\tau|\cdot)$ denotes conditional uncertainty. The first and second terms of (17) will be denoted as $H_{\downarrow}(\tau)$ and $H_{\uparrow}(\tau)$, respectively.

In order to calculate $H_{\downarrow}(\tau)$ (corresponding to the lower sub-window, where the distribution is non-uniform), we first need to derive the expression for $Pr\left(c<\tau\right)$:

$$\Pr\left(c<\tau\right)=\frac{(U+1-\alpha)\,(\tau-1)+\alpha U}{U\,(U+1)}. \tag{18}$$

The left term in the numerator of (18) corresponds to non-0 levels and the right term to the 0 level. To derive the conditional uncertainty $H\left(\tau\mid c<\tau\right)$ we need to calculate

$$\Pr(c=0|c<\tau)=\frac{\alpha}{U+1}P_\tau^{-1}, \tag{19}$$

and

$$\Pr(c=k,k\neq 0|c<\tau)=\frac{U+1-\alpha}{U(U+1)}P_\tau^{-1}, \tag{20}$$

where

$$P_\tau\triangleq\Pr\left(c<\tau\right). \tag{21}$$

By definition of the expected uncertainty as the distribution entropy we now write using (19) and (20)

$$\begin{aligned}H\left(\tau\mid c<\tau\right)&=P_\tau^{-1}\frac{\alpha}{U+1}\log\left(\frac{P_\tau(U+1)}{\alpha}\right)\\&+\;P_\tau^{-1}(\tau-1)\frac{U+1-\alpha}{U(U+1)}\log\left(\frac{P_\tau U(U+1)}{U+1-\alpha}\right).\end{aligned} \tag{22}$$

Now multiplying (22) and (18), and using (21) we get

$$\begin{aligned}H_{\downarrow}(\tau)&=\frac{\alpha}{U+1}\log\left(\frac{P_\tau(U+1)}{\alpha}\right)\\&+\;(\tau-1)\frac{U+1-\alpha}{U(U+1)}\log\left(\frac{P_\tau U(U+1)}{U+1-\alpha}\right).\end{aligned} \tag{23}$$

Substituting P_τ from (18) and reorganizing, we get

$$H_\downarrow(\tau) = \frac{\alpha}{U+1}\log\left(1+\frac{(U+1-\alpha)(\tau-1)}{\alpha U}\right) + (\tau-1)\frac{U+1-\alpha}{U(U+1)}\log\left(\tau-1+\frac{\alpha U}{U+1-\alpha}\right). \quad (24)$$

Moving to the term $H_\uparrow(\tau)$ (corresponding to the upper sub-window, where the distribution is uniform), we again split to

$$\Pr(c \geqslant \tau) = \frac{(U+1-\alpha)(U-\tau+1)}{U(U+1)}, \quad (25)$$

and

$$H(\tau \mid c \geqslant \tau) = \Omega([\tau, U]) = \log(U-\tau+1). \quad (26)$$

Altogether we combine (25) and (26) to get

$$H_\uparrow(\tau) = \frac{(U+1-\alpha)(U-\tau+1)}{U(U+1)}\log(U-\tau+1). \quad (27)$$

Finally, we combine (24) and (27) to get the desired expected non-uniform uncertainty from

$$H(\tau) = H_\downarrow(\tau) + H_\uparrow(\tau).$$

■

Note that substituting $\alpha = 1$ into $H(\tau)$ above degenerates to the uniform case of (cf. (10))

$$H(\tau) = \frac{\tau}{U+1}\log(\tau) + \frac{U-\tau+1}{U+1}\log(U-\tau+1).$$

B. Non-uniform read algorithms

When the cell levels are not uniformly distributed, we wish to use our knowledge of the distribution to obtain faster read algorithms. All the previous read algorithms in the paper assumed the uniform distribution for their measurement selection criteria. Hence a natural way to improve the read time is by employing selection criteria that take into consideration the actual distribution.

For the single-parameter non-uniform level distribution considered in the previous sub-section, Proposition 8 gives a closed-form calculation of the expected uncertainty after applying a given threshold τ. It is thus straightforward to "plug in" this modified expected uncertainty to all the previous algorithms that select measurements by maximizing the total expected reduction in uncertainty. It turns out that using the non-uniform uncertainty in the selection criteria can improve the average read time significantly. A quantitative evaluation through simulations of this advantage is deferred to the next sub-section. Still in the analytic domain, we now want to obtain a closed form expression for the threshold τ that optimally reduces the uncertainty of a given cell. Such an expression can simplify the read algorithm by avoiding the need to exhaustively check all levels in the uncertainty window. The next proposition provides this optimal threshold as a function of the parameter α of the non-uniform distribution.

Proposition 9. *For a memory cell with uncertainty window* $[0, U]$*, and non-uniform level distribution, the threshold that maximizes the uncertainty reduction is given by*

$$\tau^\star = \frac{U(U-2\alpha+1)}{2(U-\alpha+1)} + 1. \quad (28)$$

Proof: Maximizing the uncertainty reduction is equivalent to minimizing the uncertainty $H(\tau)$. To find the τ that minimizes $H(\tau)$ we take the derivative of $H(\tau)$ from Proposition 8, and equate to zero. After some simple manipulations we get

$$\frac{dH(\tau)}{d\tau} = \frac{(U-\alpha+1)}{U(U+1)}\log\left(\frac{(U-\tau+1)(U-\alpha+1)}{(U-\alpha+1)(\tau-1)+\alpha U}\right). \quad (29)$$

Now equating to zero we get the τ given in (28). ■

Note that $\tau^\star$ may be a non-integer, in which case we take the two integers closest to it and take the one with the smaller uncertainty. In the uniform case ($\alpha = 1$), the optimal threshold in (28) degenerates to $\tau^\star = (U+1)/2$, as selected by the standard binary search.

C. Non-uniform distribution results

To evaluate the advantage of selection criteria matched to the level distribution, we conducted the following study using simulations. We randomly drew cell levels in a $n \times n$ array with a non-uniform distribution that fixes the probability to select level 0 to some predefined fraction. Then we wrote down the number of measurements required to read the full array when using the standard BinarySearch, ANDF and CRDF algorithms, and then with their counterparts that maximize the *non-uniform* uncertainty reduction. We repeated each experiment 1000 times and plotted the average numbers of measurements. The measurement counts were normalized by the number of rows n. The fraction of 0-level cells we chose is 50%.

In Fig. 6 the results are given for $q = 8$. The results are plotted for the standard binary search, CRDF and ANDF algorithms (solid curves), and for the respective algorithms that use the non-uniform criterion (dashed curves).

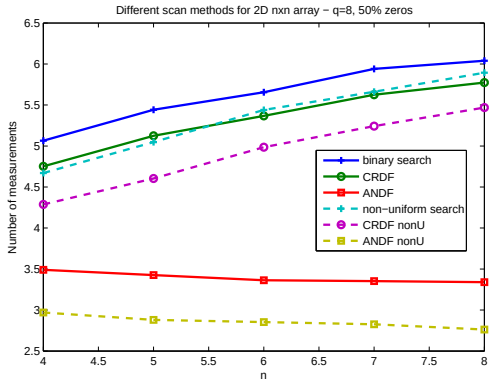

Figure 6. Simulated average measurement counts for $n \times n$ array with $q = 8$.

As can be seen, there is a significant improvement in the performance of all of the adapted algorithms. Similar results for $q = 32$ are presented in Fig. 7.

From the plots we learn that for higher values of q the improvement in performance gained by using dedicated non-uniform read algorithms is higher.

V. Conclusion

Adaptive threshold read algorithms can reduce the number of measurements required to read a memory array. In particular, 2D reading algorithms were shown to improve over pure

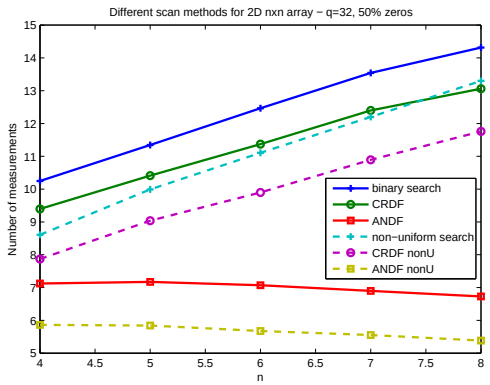

Figure 7. Simulated average measurement counts for $n \times n$ array with $q = 32$.

1D measurements. It was also shown that when the memory levels are not uniformly distributed, the read algorithms can benefit significantly by considering the distribution in the measurement selection. There are many directions to pursue as future research. Firstly, better theoretical understanding of the 2D read algorithms is an important open subject. The relationship between the measurement degree of freedom to the performance is of particular interest. Secondly, tighter lower bounds on the number of measurements seem likely to exist. Finally, it is worthwhile to study additional aspects of the storage vs. read-speed problem.

VI. Acknowledgment

This work was supported in part by the Marie Curie CIG grant and by the Technion Funds for Security Research.

References

[1] A. Jiang, and J. Bruck, "Data representation for flash memories," in *Data Storage*, In-Tech Publisher, 2010, pp.53-74.

[2] B. Peleato, et al., "Towards minimizing read time for NAND flash," in *IEEE Global Communications Conf.*, GLOBECOM, Dec. 2012, pp.3219-3224.

[3] E. Hemo, and Y. Cassuto, "Adaptive threshold read algorithms in multi-level non-volatile memories," in Proc. of IEEE Int. Symp. on Information Theory, Istanbul, Turkey, 2013.

[4] R.J. Baker, *CMOS: Circuit Design, Layout, and Simulation*, Hoboken, NJ:Wiley-IEEE Press, 2010.

[5] Ogawa, Tomohiko et al., "SAR ADC algorithm with redundancy and digital error correction," *IEICE Trans. on Fundamentals*, vol.E93-A, pp.415-423, 2010.

[6] Lin, Ying-Zu, et al., "An asynchronous binary-search ADC architecture with a reduced comparator count," *IEEE Trans. Circuits Syst. I, Reg. Papers*, pp.1829-1837, Aug. 2010.

[7] Mesgarani, Ali, and S. U. Ay. "A single channel 6-bit 900MS/s 2-bits per stage asynchronous binary search ADC," in *IEEE 54th International Midwest Symposium on Circuits and Systems*, MWSCAS, Aug. 2011, pp.1-4.

[8] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," *IEEE Trans. Inf. Theory*, Vol 57, No. 12, Dec. 2011.

[9] J. van Lint and R. Wilson, *A Course in Combinatorics*, Cambridge University Press, 2001.

[10] A. Mohr and T.D. Porter, "Applications of Chromatic Polynomials Involving Stirling Numbers," Department of Mathematics Southern Illinois University, 2008.