New Results on Erasure Combinatorial Batch Codes

Phuc-Lu Le[*], Son Hoang Dau[†], Hy Dinh Ngo[*], Thuc D. Nguyen[*§]

[*]Faculty of Information Technology, Ho Chi Minh City University of Science, Vietnam
[†]The School of Computing Technologies, RMIT University, Australia
{lplu, ndthuc, ndhy}@fit.hcmus.edu.vn,   sonhoang.dau@rmit.edu.au

Abstract—We investigate in this work the problem of Erasure Combinatorial Batch Codes, in which n files are stored on m servers so that every set of $n - r$ servers allows a client to retrieve at most k distinct files by downloading at most t files from each server. Previous studies have solved this problem for the special case of $t = 1$ using Combinatorial Batch Codes. We tackle the general case $t \geq 1$ using a generalization of Hall's theorem. Additionally, we address a realistic scenario in which the retrieved files are consecutive according to some order and provide a simple and optimal solution for this case.

Index Terms—Combinatorial batch codes, erasures, multiset, non-adaptive group testing, consecutive files.

I. INTRODUCTION

Combinatorial Batch Codes (CBC) were defined by Paterson *et al.* in [1] as a combinatorial version of batch codes (introduced by Ishai *et al.* [2]). More specifically, a CBC allows ones to store n files on m servers (possibly with repetitions among servers) such that any k distinct items can be retrieved by downloading at most t items from each server. The goal is to minimize the total number N of items stored across all servers, given n, m, k and t. To address the practical scenario in distributed storage systems where servers may fail or maintenance frequently, CBC was also generalized to Erasure Combinatorial Batch Codes (ECBC) by J. Jung *et al.* [3], which allows up to r servers to fail. Several constructions of ECBC in which every item is stored in the same number of servers were developed in [3].

As far as we know, most of the previous works on CBC/ECBC assumed that $t = 1$, i.e., each server can be used at most $t = 1$ time. The problem has been approached using various mathematical structures, including dual systems, extremal hypergraphs, and transversal matroids (see, e.g., [4]–[6]). A number of works focused on solving CBC for specific cases, e.g. when k is small ($k = 3, 4, 5$) or when $n = m + 2$ [7]. In almost all cases, the main used method is Hall's theorem from graph theory. More specifically, by viewing a CBC/ECBC as a matching in a bipartite graph, one can apply Hall's condition to find the optimal total storage.

In this paper, using also Hall's theorem, we extend the previous studies to address the more general case of ECBC for any $r \geq 0$ and $t \geq 1$. Note that in practice, allowing the client to download more than one item from each server can improve the efficiency of data retrieval. Additionally, motivated by the application in video streaming where several consecutive video chunks are often downloaded at one time,

[§]Corresponding author

we consider and provide the optimal solution for a special version of ECBC in which the retrieved files are consecutively indexed (instead of being random). Although this version is rather simple to address, it opens up an entirely new research direction on batch codes where batches to be downloaded are not randomly selected but follow some special patterns. Along with the approaching CBC/ECBC problems, we focus on the efficient algorithm to retrieve a list of files from the given ECBC system. Finally, we briefly note the relationship between ECBC and Non-Adaptive Group Testing (NAGT).

II. PRELIMINARIES

A. Problem Definitions

We first recall the definition of an ECBC.

Definition 1. *An Erasure Combinatorial Batch Code denoted by* $\mathsf{ECBC}(n, m, k, t, r)$ *is a set system* $(X, \mathcal{S})$ *where* $X = \{1, 2, \ldots, n\}$ *and* $\mathcal{S}$ *is a collection of m-subsets of X, denoted by* $S_1, S_2, \ldots, S_m$, *such that for every subset* $X' \subset X$ *of size up to k and for every subset J of* $\{1, 2, \ldots, m\}$ *with* $|J| \geq m - r$, *there exists a subset* $C_j \subset S_j$, $j \in J$, *satisfying* $|C_j| \leq t$ *and* $X' = \bigcup_{j \in J} C_j$. *The goal is to construct an* $\mathsf{ECBC}(n, m, k, t, r)$ *that minimizes the total storage* $N \triangleq \sum_{j=1}^{m} |B_j|$.

In Definition 1, the set X can be considered as the list of indices of n files while $S_1, S_2, \ldots, S_m$ are the indices of files that stored on m server. Here, r refers to the maximum number of inactive servers simultaneously.

Next, we recall the well-known Hall's Marriage theorem, which plays an essential role in the constructions of CBC/ECBC.

B. Hall's Marriage Theorem

Hall's theorem is stated as follows: for positive integers m and n, assume that $A_1, A_2, \ldots, A_n$ are n subsets of $X = \{1, 2, \ldots, m\}$. If for every subset I of $\{1, 2, \ldots, n\}$, the condition $\left| \bigcup_{i \in I} A_i \right| \geq |I|$ holds, then there exist n distinct elements from these n subsets, one from each set. We refer to the aforementioned condition as the Hall's condition. The theorem states that in a simple undirected bipartite graph $G = (X, Y, E)$ with two disjoint sets of vertices X, Y and E as the edges connecting vertices between them, there exists a matching from X to Y if and only if, for any subset X' of X with size k, $|V(X')| \geq |X'|$, where $V(X')$ represents the set of vertices in Y that are adjacent to some vertex in X'.

In other words, Hall's condition can be applied to a subset X' of X with size k to find a matching from X' to Y. This is the other form of Hall's theorem which is compatible with solving the CBC-ECBC problems and was mentioned by Bujtas and Tuza in [8].

The way to apply Hall's condition is straightforward: by defining the equivalence between the original problem and Hall's condition, and then establishing the dual condition, we can find the optimal solution for some cases of m and n. The dual condition can be understood as the requirement that 'the files must be stored on separate servers, so as to prevent servers from storing too many files'. These concepts form the basis for constructing the proof of the general case where $t \geq 1$ and $r \geq 0$ that will be discussed in more detail in the next section.

C. Existing Results for CBC and ECBC

The incidence matrix $A = (a_{ij})_{m \times n}$ of a $\mathsf{CBC}(n, m, k, t)$ or $\mathsf{ECBC}(n, m, k, t, r)$ is defined as: $a_{ij} = 1$ if and only if server i stores file j, and $a_{ij} = 0$ otherwise. Let $F_j \triangleq \{i : a_{ij} = 1\}$ be the set of rows that has a '1' in column j and $S_i = \{j : a_{ij} = 1\}$ be the set of columns that has a '1' in row i. In other words, F_j consists of the indices of servers that store file j, $j = 1, \ldots, n$ and S_i consists of the indices of files that are stored in server i, $i = 1, \ldots, m$.

For some special parameter ranges, the minimum total store of a $\mathsf{CBC}(n, m, k, t)$ can be determined.

Theorem 1 (Paterson, Stinson, and Wei [1]). *Let $N(m, n, k)$ denote the minimum total storage of a $\mathsf{CBC}(n, m, k, t)$ when $t = 1$. Then the following statements hold.*

1) *If $n = m$ then $N(n, m, k) = n$.*
2) *If $m = k$ then $N(n, m, k) = kn - k(k-1)$.*
3) *If $n = m + 1$ then $N(n, m, k) = m + k$.*
4) *If $n \geq (k-1)\binom{m}{k-1}$ then*

$$N(n, m, k) = kn - (k-1)\binom{m}{k-1}.$$

5) *If $\binom{m}{k-2} \leq n \leq (k-1)\binom{m}{k-1}$ then*

$$N(n, m, k) = n(k-1) - \left\lfloor \frac{(k-1)\binom{m}{k-1} - n}{m - k + 1} \right\rfloor.$$

Paterson, Stinson, and Wei [1] were the first to study the property of the incidence matrix of a CBC ($t = 1$) using the Hall's condition.

Theorem 2 (Paterson, Stinson, and Wei [1]). *The $m \times n$ binary matrix A represents a $CBC(n, m, k)$ if and only if one of the following conditions holds.*

(1) *For every $c \in \{1, 2, \ldots, k\}$, and for every subset $J \subset \{1, 2, \ldots, n\}$ of c elements, it holds that $\left| \bigcup_{j \in J} F_j \right| \geq c$. In other words, this condition requires that every set of c files must be collectively stored on at least c servers.*

(2) *For every $d \in \{0, 1, \ldots, k-1\}$, and every subset $I \subset \{1, 2, \ldots, m\}$ of size d, it holds that $\left| \bigcup_{i \in I} S_i \right| \leq d$. In*

other words, this condition requires that every set of d servers store at most d files.

Bujtas and Tuza [8] extended the aforementioned approach to the CBC problem with an arbitrary t.

Theorem 3 (Bujtas and Tuza [8]). *The $m \times n$ binary matrix A represents a $\mathsf{CBC}(n, m, k, t)$ if and only if one of the following conditions holds.*

(1) *For every $c \in \{1, 2, \ldots, k\}$, and every subset $J \subset \{1, 2, \ldots, n\}$ of size c, it holds that $\left| \bigcup_{j \in J} F_j \right| \geq \lceil \frac{c}{t} \rceil$. In other words, the condition requires that any set of c files must be collectively stored on at least $\lceil \frac{c}{t} \rceil$ servers.*

(2) *For every $d \in \{0, 1, \ldots, \lceil \frac{k}{t} \rceil - 1\}$, and every subset $I \subset \{1, 2, \ldots, m\}$ of size d, it holds that $\left| \bigcup_{i \in I} S_i \right| \leq dt$. In other words, this condition requires that any set of d servers together store at most dt files.*

Jung, Mummert, Niese, and Schroeder [3] extended the previous results to the ECBC problem with $t = 1$, via an extension of Hall's theorem.

Theorem 4 (Jung, Mummert, Niese & Schroeder [3]). *The $m \times n$ binary matrix A represents an $\mathsf{ECBC}(n, m, k, t, r)$ with $t = 1$ if and only if one of the following conditions holds.*

(1) *For every $c \in \{1, 2, \ldots, k\}$, and every subset $J \subset \{1, 2, \ldots, n\}$ of size c, it holds that $\left| \bigcup_{j \in J} F_j \right| \geq r + c$. In other words, every set of $c \leq k$ files collectively must be stored on at least $r + c$ servers.*

(2) *For every $d \in \{r, r+1, \ldots, r+k-1\}$, and for every subset $I \subset \{1, 2, \ldots, m\}$ of size d, it holds that $\left| \bigcup_{i \in I} S_i \right| \leq d - r$. In other words, every set of d servers store at most $d - r$ files.*

With the background and definitions established in this section, we are now ready to present our main results.

III. MAIN RESULTS

In this section, we establish the generalization of Hall's condition for ECBC for any $r \geq 0$, $t \geq 1$, and demonstrate its application in determining the minimum total storage $N(n, m, k, r, t)$ of an ECBC problem in several cases.

A. The Main Theorem for ECBC with General r and t

Theorem 5. *A binary matrix A of size $m \times n$ represents an $\mathsf{ECBC}(n, m, k, t, r)$ if and only if one of the following conditions holds.*

(1) *For every $c \in \{1, 2, \ldots, k\}$, and for every subset $J \subset \{1, 2, \ldots, n\}$ of size c, it holds that*

$$\left| \bigcup_{j \in J} F_j \right| \geq \left\lceil \frac{c}{t} \right\rceil + r.$$

In other words, the condition requires that any set of $c \leq k$ files must be collectively stored on at least $\lceil \frac{c}{t} \rceil + r$ servers.

(2) *For every $d \in \{r, r+1, \ldots, r+\Delta-1\}$, and for every subset $I \subset \{1, 2, \ldots, m\}$ of size d, it holds that*

$$\left| \bigcup_{i \in I} S_i \right| \leq t(d - r).$$

In other words, the condition requires that any set of d servers store at most $t(d-r)$ files. Here, $\Delta \triangleq \lceil \frac{k}{t} \rceil$.

Proof. We divide the proof into two steps as follows.

First step. We show that a binary matrix A represents an $\mathsf{ECBC}(n, m, k, t, r)$ if and only if (1) holds.

First, let us assume that the matrix A represents an $\mathsf{ECBC}(n, m, k, t, r)$. Regardless of r unavailable servers, any $c \leq k$ files can be retrieved from the remaining servers by downloading at most t files from each. These files must be stored on at least $r + \lceil \frac{c}{t} \rceil$ servers originally, for otherwise, there would be at most $\lceil \frac{c}{t} \rceil - 1$ available servers storing any of these files, and downloading t from each would not be enough to recover c files.

Next, assume that the binary matrix A meets condition (1) and moreover, there is a collection of unavailable servers S with $|S| \leq r$. For every set $J \subset \{1, 2, \ldots, n\}$ of size c, i.e., c files, the number of servers that store at least one of these files is

$$\left| \bigcup_{j \in J} (F_j \backslash S) \right| = \left| \left(\bigcup_{j \in J} F_j \right) \backslash S \right| \geq \left| \bigcup_{j \in J} F_j \right| - r$$

$$\geq \left\lceil \frac{c}{t} \right\rceil + r - r = \left\lceil \frac{c}{t} \right\rceil.$$

We now create an $(mt) \times n$ matrix A' from A by replicating each row of A t times. In other words, we replicate each server t times, where each copy stores the same set of files as the original one. Then for the same set $J \subset \{1, 2, \ldots, n\}$ of size $c \leq k$, the number of servers (including the duplicated ones) storing at least one file is

$$t \left| \bigcup_{j \in J} (F_j \backslash S) \right| \geq \left\lceil t \cdot \frac{c}{t} \right\rceil = c,$$

which means that the Hall's condition is satisfied for the sets F'_j, $j = 1, \ldots, n$, defined for the matrix A'. By applying Hall's theorem, there is a matching between the c files and a set of available servers (original and copies). Since each of original (available) servers appears at most t times in this set, it appears in the matching at most t times. This means that there are at most t files downloaded from each available servers while retrieving c files. Thus, the matrix A represents an $\mathsf{ECBC}(n, m, k, t, r)$.

Second step. We aim to prove the equivalence between the two conditions (1) and (2).

First, suppose that A does not satisfy condition (1). Then there exists $J \subset \{1, 2, \ldots, n\}$ of size $c \leq k$ such that $\bigcup_{j \in J} F_j$

contains $\lceil \frac{c}{t} \rceil + r - 1$ elements at most. Take $I \triangleq \bigcup_{j \in J} F_j \subset \{1, 2, \ldots, m\}$. Then we have

$$|I| \leq \left\lceil \frac{c}{t} \right\rceil + r - 1 \leq \Delta + r - 1.$$

Note that for each F_i, $1 \leq i \leq n$, we must have $|F_i| \geq r+1$, as otherwise there will be a case when all servers containing the ith file are unavailable, which results in the permanent loss of this file. This implies that $|I| \geq r + 1$. Therefore,

$$\left| \bigcup_{i \in I} S_i \right| = c > (c + tr - 1) - tr \geq t(|I| - r),$$

implying that the condition (2) of Theorem 5 does not hold.

Supposing that A does not satisfy condition (2) of Theorem 5. Then there exists $d \in \{r, \ldots, r+\Delta-1\}$ and subset $I \subset \{1, 2, \ldots, m\}$ of size d in which $Y = \bigcup_{i \in I} S_i$ has more than $t(d-r)$ elements. Take $J \subset Y$ and $|J| = t(d-r) + 1$. It is not difficult to check that

$$t(d-r) + 1 \leq t(\Delta - 1) + 1 < k.$$

Hence, $|J| < k$ and

$$\left| \bigcup_{j \in J} F_j \right| = d < (d - r + 1) + r = \left\lceil \frac{t(d-r)+1}{t} \right\rceil + r,$$

which implies that the condition (1) does not hold.

By combining what we have proved in the two steps, the theorem follows. $\qquad \square$

We now proceed to explore the application of Theorem 5 in determining the optimal total storage for $\mathsf{ECBC}(n, m, k, t, r)$.

B. ECBC with Optimal Total Storage

In is section, we use Theorem 5 to obtain ECBC with optimal total storage in some specific cases. The previous researches have shown that in almost all cases, the minimum total storage for an $\mathsf{ECBC}(n, m, k, t, r)$ where $r = 0$ and $t = 1$ can be represented as a function of its parameters. By replacing $k \rightarrow k + r$ for the case of $t = 1, r > 0$ or $k \rightarrow \lceil \frac{k}{t} \rceil$ for $t > 1, r = 0$, we can obtain the corresponding answers. Thus, for the general case of $t \geq 1, r \geq 0$, it is possible to replace $k \rightarrow \lceil \frac{k}{t} \rceil + r$ and apply the known results to find new bounds, the rest of work is construction. Let $N(n, m, k, t, r)$ denote the minimum total storage achievable by an $\mathsf{ECBC}(n, m, k, t, r)$.

Theorem 6. *If $m = \lceil \frac{k}{t} \rceil + r$ and $n \geq tm$ then $N(n, m, k, t, r) = m(n - tm + r + 1)$.*

Proof. By applying the condition (2) of Theorem 5, we can see that each set of size $d = m - 1$ servers contains at most $t(m - 1 - r)$ files. As a result, the rest contains at least $n - t(m - 1 - r)$ files. Hence, by counting the number of storage on overall servers, we have

$$N \geq m(n - t(m - 1 - r)) = mn - tm(m - 1 - r)$$
$$= tm(r + 1) + m(n - tm).$$

TABLE I: Construction for ECBC when $n = 17, m = 5, k = 10, t = 3, r = 1$
(the empty cells contain 0)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
S_1	1	1	1										1	1	1	1	1
S_2	1	1	1	1	1	1										1	1
S_3			1	1	1	1	1	1	1							1	1
S_4							1	1	1	1	1	1				1	1
S_5										1	1	1	1	1	1	1	1

Construction: each file from indices $1, 2, \ldots, tm$ is stored on $r + 1$ servers: files $1 \to t$ on servers $1 \to r + 1$, files $t + 1 \to 2t$ on servers $2 \to r + 2$, and so on; the remaining $n - tm$ files are store on all servers. One can verify that in this case, $N(n, m, k, t, r) = tm(r + 1) + m(n - tm)$ as needed and this also satisfies the ECBC condition. □

Example. For $k = 10, t = 3, r = 1$ then $m = 5$ and $n = 17$ then $N_{\min} = 17 \cdot 5 - 3 \cdot 5 \cdot 3 = 40$. The construction is showed in Table I.

The bound for another case can be obtained as follows (the similar idea for construction can be found in in [1]):

Theorem 7. *Let* $h \triangleq \lceil \frac{k}{t} \rceil$. *If* $n \geq (h - 1)\binom{m}{r+h-1}$ *and* $m \geq h + r$ *then we have*

$$N(n, m, k, t, r) = nh - (h - 1)\binom{m}{r + h - 1}.$$

C. Application to the Multiset CBC with $r > 0$

In [9], Zhang, Yaakobi, and Silberstein studied the multiset CBC where each file can appear multiple times in the retrieval list, i.e each $F_j \in \{1, 2, \ldots, n\}$ is a multiset with elements having multiplicity up to p. Now we consider the erasure multiset ECBC in the presence of server failures. Based on Theorem 5, we can state the Hall's conditions for this problem following the study in [9].

Theorem 8. *Denote* $\delta = \lceil k/p \rceil$. *The sets* F_j, $j \in \{1, 2, \ldots, n\}$ *represent a multiset ECBC with parameters* m, n, k, r, t, p *if and only if one of the following conditions is satisfied.*

(1) *For every* $c \in \{1, 2, \ldots, \delta\}$ *and for every subset* $J \subset \{1, 2, \ldots, n\}$ *of size* c, *it holds that*

$$\left| \bigcup_{j \in J} F_j \right| \geq \left\lceil \frac{\min\{cp, k\}}{t} \right\rceil + r.$$

(2) *For every* $d \in \{r, r+1, \ldots, r + \lceil \frac{p\delta}{t} \rceil - 1\}$, *and for every subset* $I \subset \{1, 2, \ldots, m\}$, *it holds that*

$$\left| \bigcup_{i \in I} S_i \right| \leq t(d - r).$$

D. Method to Retrieve Files from the Given ECBC Matrix

Consider an ECBC matrix A with parameters n, m, k, t, r as defined. The goal is to find the list of servers containing the k files to query, given a list of r unavailable servers. To solve the problem, we use a bipartite graph $G = (V, E)$ and apply a maximum matching algorithm such as Hopcroft-Karp

which run in the time complexity as $O(\sqrt{V}E)$. The procedure consists of two steps:

1) Construct a sub-matrix from the ECBC matrix that has columns corresponding to the k files to retrieve and rows corresponding to servers (excluding unavailable ones) that each has at least one of these files, meaning each row has at least one number 1. The size of this sub-matrix is at most $(m - r) \times k$.

2) Apply the Hopcroft–Karp algorithm to find the matching in the following two cases:

   - If $t = 1$, find the maximum matching between columns and rows (ECBC property of the matrix ensures this will contain k pairs).
   - If $t > 1$, create t copies of each row and then apply the algorithm as if $t = 1$. The chosen edges do not share a common vertex which can ensure that each server is accessed at most t times. From these edges, one can find the list of needed servers and the corresponding files on those servers.

Hence we can apply this procedure in the realistic systems which use the CBC/ECBC ideas.

IV. BATCH CODES WITH CONSECUTIVE ITEMS

In this section, we discuss a special version for the ECBC problem in which the retrieved files have consecutive indices. For examples, databases storing purchasing history, weather forecasts, school reports, medical records, can often be arranged in a linear order chronologically. A set of consecutive files provides relevant data for a specific period of time. Although a quite simple solution exists for this problem, the setting in which only specially structured subsets of files (instead of arbitrary subsets like in the traditional batch code) are retrieved is novel and has potential applications in practice. The only other work in this direction that we are aware of is [10, App. F], in which the database items are placed at the nodes of a binary tree and only batches of items lying along a root-to-leaf path are retrieved, which correspond to Merkle proofs in a Merkle tree [11].

A. Solution for General ECBC

If $t = 1$, we need the constraints $n \geq k$ and $m \geq k + r$. If r servers are unavailable and there are fewer than $k + r$ servers, it is not possible to retrieve k files. Moreover, the lower bound on the number of servers storing each file is $r + 1$, which ensures all r servers storing the same file are active. Therefore, we get the necessary condition $N \geq (r + 1)n$.

TABLE II: Construction for ECBC when $n = 9, k = 3, m = 5, r = 2$ in the consecutive version

	1	2	3	4	5	6	7	8	9
S_1	1			1			1		
S_2		1			1			1	
S_3			1			1			1
S_4	1	1	1	1	1	1	1	1	1
S_5	1	1	1	1	1	1	1	1	1

We demonstrate that the number N is the minimum for all values of m and n. We first construct the ECBC matrix:

(1) We store the files with the indices congruent to i modulo k on the servers with indices $i \in \{0, 1, 2, \ldots, k - 1\}$. This way, each file is stored once, resulting in a total storage count of n.

(2) For the servers with indices $i \in \{k, k+1, \ldots, k+r-1\}$, we store all files. So the total storage count to rn.

Hence, the total number of storage is $N_{\min} = n + rn = (r + 1)n$. It is not difficult to check that this construction satisfies all the given ECBC conditions:

- If r servers in group (2) are unavailable, we still have servers in group (1) to retrieve the files. Note that we only consider k consecutive files, two files with indices that have a difference greater than k will not be retrieved simultaneously. This is the main idea of the construction.

- If some servers in group (1) is unavailable, we can replace it with any server in group (2), since each server in that group stores all files.

For the case where $t > 1$, the above result remains tight, since the bound $(r + 1)n$ holds for all values of t. It just enables us to reduce the number of servers from k to $\left\lceil \frac{k}{t} \right\rceil$.

So, in the general ECBC problem, the solution for consecutive files version is $N_{\min} = (r + 1)n$.

Example. For $n = 9, m = 5, k = 3, r = 2$ then $N_{\min} = (r + 1)n = 27$. The construction is showed on the Table II.

In this case, by the periodically distribution of the files among servers, one can modify the given matrix A when some of values m, n or r change as follows: if the number n of files increases then add new columns to the matrix such that for each of them, the last r cells are all filled by 1, an extra number 1 will be filled base on the modulo k; if the number m of servers increases, it does not matter since $k + r$ servers are enough for the system; and if the upper bound r increases then the number of servers must increases too, each of the new servers will store all files there. When number k or t changes, it is required to edit quite a lot to get the new optimal system.

B. The Relations between CBC/ECBC and NAGT

NAGT is a technique that aims to minimize the number of tests required to identify some particular defected items among all items. The items are grouped and tested together to improve the accuracy. Tests are performed independently, can be done in parallel, and are represented by a measurement matrix.

In [12], Jia *et al.* studied the connections between ECBC and NAGT through the incident binary matrices, focusing on disjunct and separable matrices. In [13], [14], NAGT with consecutive positives has been studied, where positive items are consecutive in a linear order. This idea inspired the ECBC problems, where the files are also linearly ordered and the retrieval files are consecutive. In this special aspect, the optimal value of storage in the ECBC problem can easily be obtained as the previous part.

Furthermore, NAGT is a well-studied problem with numerous theories and research results have been done. By discovering new relationships between two problems ECBC and NAGT, we can potentially apply ideas from one to another. Here, we compare some other aspects of these two problems.

- Objectives: ECBC aims to store n files in m servers, while NAGT aims to test N items with T tests.
- Requested items: ECBC requires k files (a subset of the original files), while NAGT requires D defect items (a subset of the original items).
- The constrains: in ECBC, it is required $k \leq \min\{m, n\}$; in NAGT, it is required $D \leq \min\{N, T\}$.
- Classification of storage objects: ECBC classifies servers into two groups - those that contain at least one required file and others. In NAGT, tests are classified into positive class and negative class.
- Erasure aspects: in ECBC, there are at most r unavailable servers while in NAGT, there are R inhibitors that can join some tests and change the result of tests.

Despite some similar aspects of these two problems, the main difference lies in the finding of subsets. ECBC finds a subset of servers based on the storage of k files, while NAGT finds a subset of items based on the test results.

V. CONCLUSIONS

In this work, we extended the previous results on Erasure Combinatorial Batch Codes to accommodate the most general case when $r \geq 0$ and $t \geq 1$, using an generalization of Hall's theorem. We also explored the relationship between ECBC and NAGT problems, and found an optimal solution for a special case when retrieved files are arranged in specific consecutive linear sequence. This provides new insights into the problem and a new approach to solve it.

In the future, we will consider the way to slightly modify the given ECBC matrix A in case that some of values n, m, k, t, r change (since the modification of whole matrix requires plenty of computation). We also aim to explore the practical applications of ECBC in real-time data storage for IoT devices and multiple servers. Our research also has significant potential in industries requiring reliable data storage, such as biomedical and agriculture, thus we would like to further investigate the ECBC concept and its potential use in these industries.

VI. Acknowledgments

References

[1] M. B. Paterson, D. R. Stinson, and R. Wei, "Combinatorial batch codes," *Advances in Mathematics of Communications*, vol. 3, no. 1, p. 13, 2009.

[2] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 262–271, 2004.

[3] J. Jung, C. Mummert, E. Niese, and M. Schroeder, "On erasure combinatorial batch codes," *Advances in Mathematics of Communications*, vol. 12, no. 1, p. 49, 2018.

[4] C. Bujtás and Z. Tuza, "Optimal combinatorial batch codes derived from dual systems," *Miskolc Mathematical Notes*, vol. 12, no. 1, pp. 11–23, 2011.

[5] N. Balachandran and S. Bhattacharya, "On an extremal hypergraph problem related to combinatorial batch codes," *Discrete Applied Mathematics*, vol. 162, pp. 373–380, 2014.

[6] R. A. Brualdi, K. P. Kiernan, S. A. Meyer, and M. W. Schroeder, "Combinatorial batch codes and transversal matroids," *Advances in Mathematics of Communications*, vol. 4, no. 3, p. 419, 2010.

[7] D. Jia and G. Zhang, "Some optimal combinatorial batch codes with k= 5," *Discrete Applied Mathematics*, vol. 262, pp. 127–137, 2019.

[8] C. Bujtás and Z. Tuza, "Relaxations of hall's condition: optimal batch codes with multiple queries," *Applicable Analysis and Discrete Mathematics*, pp. 72–81, 2012.

[9] H. Zhang, E. Yaakobi, and N. Silberstein, "Multiset combinatorial batch codes," *Designs, Codes and Cryptography*, vol. 86, no. 11, pp. 2645–2660, 2018.

[10] Q. Cao, R. Gagiano, D. Huynh, X. Yi, S. H. Dau, P. L. Le, Q.-H. Luu, E. Viterbo, Y.-C. Huang, J. Zhu, M. M. Jalalzai, and C. Feng, "Parallel private retrieval of Merkle proofs via tree colorings," 2023. Available at https://arxiv.org/abs/2205.05211.

[11] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT)*, pp. 369–378, Springer, 1987.

[12] D. Jia, S. Zhang, and G. Zhang, "Erasure combinatorial batch codes based on nonadaptive group testing," *Designs, Codes and Cryptography*, vol. 87, no. 7, pp. 1647–1656, 2019.

[13] C. J. Colbourn, "Group testing for consecutive positives," *Annals of Combinatorics*, vol. 3, no. 1, pp. 37–41, 1999.

[14] T. V. Bui, M. Cheraghchi, and T. D. Nguyen, "Improved algorithms for non-adaptive group testing with consecutive positives," *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 1961–1966, 2021.