On the Maximally Recoverable Property for Multi-Protection Group Codes

Minghua Chen, Cheng Huang, and Jin Li
Microsoft Research, One Microsoft Way, Redmond, WA, 98052
{minghua.chen, cheng.huang, jinl}@microsoft.com

***Abstract*—In this paper, we study the Maximally Recoverable (MR) property for *multi-protection group* (MPG) codes. MPG codes with MR property achieve the best erasure recoverability given configurations, where a configuration represents the structural relationship between data and parity symbols. We present construction and decoding algorithms for MPG codes with MR property. We show that both recoverability and minimum decoding overhead of any MPG code with MR property depend only on the configuration, where decoding overhead is defined as the additional number of symbols to access, in order to decode the lost data symbols.**

I. Introduction

We define an error correction and/or erasure resilient code as a *multi-protection group* (MPG) code, if its data symbols are separated into a number of protection groups, and one or more parity symbols are generated to exclusively protect the data symbols within each group.

Many popular error correction codes and/or erasure resilient codes widely used in communication and storage systems are MPG codes. For example, Product codes [3] are two-dimensional codes constructed by encoding a rectangular array of data symbols with one code along rows and with another code along columns. It uses one protection group per row and per column. Turbo codes [4] are Shannon limit approaching error correction codes in low signal-to-noise ratio (SNR) environment. It is parallel concatenations of two or more recursive systematic convolutional codes. Each component encoder of the Turbo codes uses a different protection group, which are usually reorganized by an interleaver that permutes the ordering of the data symbols in the protection group.

The low-density parity-check (LDPC) codes [5] can be considered as simplified MPG codes, in which each parity symbol leads to a separate small protection group whose members are the data symbols XORed to form the parity symbol. Generalized LDPC (GLDPC) relaxes the constraints, and allows more than one parity symbol and may use Galois Field operation. However, it still keeps a small protection group size and uses iterative in-group decoding strategy [6], [7]. In comparison, general MPG codes allow arbitrary protection group size, and may perform cross protection group decoding.

Our recently proposed Pyramid Codes [1] are also MPG codes. We have shown that MPG codes can help to reduce the I/O overhead, in terms of extra throughput that is needed to access data symbols in a distributed storage system, and at the same time satisfy the same data reliability and storage overhead requirement [1].

We define the way that protection groups are formed over MPG codes as a *configuration*. A rigorous mathematical definition of a configuration will be introduced in Section II. A configuration defines the structural relationship between data symbols and parity symbols. Research work in the construction of LDPC codes shows that configuration plays a big role in the error/erasure recoverability of MPG codes [8], [9].

Given a configuration, some interesting questions to answer are what is the best erasure recoverability an MPG code can achieve, how to construct such optimal codes, and how to perform decoding. Because MPG codes have been shown to be effective in distributed storage scenarios [1], we are concerned with the best erasure recoverability, though some results may be extended to error recoverability.

For example, Fig. 1 shows a configuration used by Product codes. 2 parity symbols are generated along the rows, and 2 are generated along the columns. If all 4 data symbols are lost, it is known that Product codes can not successfully recover them. One natural question to ask is that does there exist a code such that the 4 lost data symbols can be recovered using the four parities.

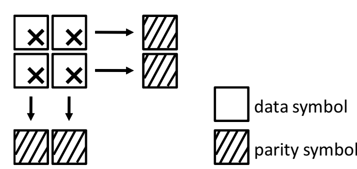

Fig. 1. Can two column parity symbols and two row parity symbols recover the four lost data symbols?

In this paper, we provide answers to the above questions by exploring the *Maximally Recoverable* (MR) property for MPG codes, which we initially studied in data storage scenarios [1]. MPG codes with MR property achieve the best erasure recoverability. For instance, for the 4 lost data symbols shown in Fig. 1, MPG codes with MR property can recover them using 2 column parity symbols and 2 row parity symbols. For simplicity, we refer to MPG codes with MR property as MR codes in the rest of the paper. The property is also studied in the scenario of network coding with links failure [2].

By presenting a construction algorithm, we show that MR codes exist under *arbitrary* configurations. We also present an optimal decoding algorithm achieving minimum decoding overhead. Such optimal decoding is not explored in [2].

To the best of our knowledge, our proposed generalized Pyramid Code [1], constructed by an algorithm similar to the one presented in Section III, and the network codes constructed by Theorem 11 in [2] are the only known non-MDS (Maximum Distance Separable) codes that have MR property.

We also show that, both the erasure recoverability and the minimum decoding overhead of MR codes depend *only* on the configuration. As such, configuration is the only parameter to tune the performance of MR codes.

II. Problem Formulation

Consider an (n, k) MPG erasure resilient code. Let $\mathcal{D} = \{d_1, d_2, \cdots, d_k\}$ be the set of data symbols to protect. We cover the entire set $\mathcal{D}$ by a number of subsets $S_1, S_2, \cdots, S_L$ where $S_l \subseteq \mathcal{D}, l = 1, \cdots, L$. Each S_l is defined as a protection group. Protection groups may intersect, overlap, or contain one another to provide different degree of protection to the data symbols.

Let $U_l = \{t_1^l, \ldots, t_{u_l}^l\}$ be the protection group of parity symbols generated using *only* the data symbols in S_l. Let $u_l = |U_l|$ be the size of set U_l, satisfying $\sum_{l=1}^{L} u_l = n-k$.

Let $\Omega = \{(S_1, U_1), (S_2, U_2), \ldots, (S_L, U_L)\}$ be a *configuration*, representing the structural relations between the data symbols and the parity symbols. Let $V_l = S_l \cup U_l, l = 1, \ldots, L$. We define *atom sets* for Ω as follows:

$$\begin{aligned} & S_i \backslash \cup_{j \neq i} S_j, && 1 \leq i \leq L \\ & (S_{i_1} \cap S_{i_2}) \backslash \cup_{j \neq i_1, i_2} S_j, && 1 \leq i_1, i_2 \leq L, i_1 \neq i_2 \\ & \cdots && \\ & \cap_{1 \leq m \leq M} S_{i_m} \backslash \cup_{j \neq i_m, 1 \leq m \leq M} S_j, && 1 \leq i_1, \ldots, i_M \leq L, \\ & && i_{m1} \neq i_{m2}, \\ & && 1 \leq m_1 \neq m_2 \leq M, \\ & && M \leq L \end{aligned}$$

There are altogether $2^L - 1$ atom sets, which are denoted as $A_1, \ldots, A_H$; some of them might be empty sets. Unlike protection groups, the atom sets are disjoint from each other and form a partition of the data set $\mathcal{D}$.

An illustrative example is shown in Fig. 2. 11 data symbols are covered by two protection groups: 8 in S_1, and 9 in S_2. The data symbols in S_1 are protected by 3 parity symbols in U_1, and the data symbols in S_2 are protected by 4 parity symbols in U_2. From atom sets' point of view, the 6 data symbols in A_3 are protected by all 7 parity symbols in $U_1 \cup U_2$; the 2 data symbols in A_1 are protected by the 3 parity symbols in U_1; the 3 data symbols in A_2 are only protected by the 4 parity symbols in U_2.

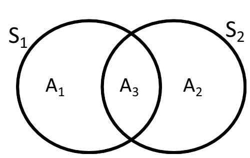

Fig. 2. There are three atom sets for two groups S_1, S_2, with $|A_1| = 2, |A_2| = 3$, $|A_3| = 6$, $u_1 = 3$ and $u_2 = 4$.

We use $\Lambda(A_h) = \{\cup_j U_j | A_h \subseteq S_j, 1 \leq j \leq L\}$ to denote the set of all parity symbols that can be used to recover erasures in A_h.

Let G be an $n \times k$ generator matrix for any systematic erasure resilient code over Ω. Each data and parity symbol maps to one row in G, classified as data row and parity row, respectively. For the parity row corresponding to the parity symbol t_i^l, the row vector can only take nonzero values in entries corresponding to the data symbols in S_l.

Given an erasure pattern e, the rows in G corresponding to the lost data and parity symbols are crossed out. All k data symbols can be recovered if and only if the remaining sub-matrix, denoted by $G'(e)$, has rank k. The remaining parity rows can be assigned to the position of lost data rows to reconstruct a rank k matrix. It is clear that one parity row can only be assigned to one lost data row. Remaining data rows can be interpreted as assigned to themselves. An example of such assignment is shown in Fig. 3.

For $G'(e)$ to have rank k, it is necessary that there exists an assignment from the remaining parity rows to the lost data rows, such that all the k diagonal entries of $G'(e)$ are nonzero [10]. We define this as a size k assignment in $G'(e)$. $G'(e)$ has rank k implies a size k assignment from the remaining data and parity symbols to the original k data symbols. For readers familiar with Tanner graph, this necessary condition is equivalent to the one based on full size matching in Tanner graph [1][1].

$$G = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline * & * & 0 & 0 & * & * \\ * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \end{array}\right] \xrightarrow{\text{erasure}} \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ - & - & - & - & - & - \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ - & - & - & - & - & - \\ \hline - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline * & * & 0 & 0 & * & * \\ * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \end{array}\right]$$

$$\xrightarrow{\text{assignment}} \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & * & * \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ * & * & * & * & * & * \\ \hline 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Fig. 3. Assigning three parity rows to three lost data rows in $G'(e)$.

Observing that the parity symbols assigned to the lost data symbols in A_h must belong to $\Lambda(A_h)$, we have the following lemma on the recoverability of erasure patterns.

Lemma 1: Under a configuration Ω, a necessary condition for an erasure pattern e to be recoverable is that there exists an *atomic assignment* from $\Lambda(A_h)$ to A_h, such that the number of lost data symbols in A_h is equal to the number of assigned parity symbols.

We show this condition is also sufficient in the next Section. Define the set containing all recoverable e under configuration Ω as $E(\Omega)$.

Assume that for e, there are l_i lost data symbols in A_i for $i \in \{1, \ldots, I\}$, and p_j remaining parity symbols in U_j for $j \in \{1, \ldots, J\}$, protecting $A_1, \ldots, A_I$. We represent an atomic assignment by a matrix in which the columns correspond to $A_1, \ldots, A_I$, and the rows correspond to $U_1, \ldots, U_J$, with zero entries in the (i, j) positions where

[1] Observing the limited space, we will not go over the condition using the Tanner graph description. Interested readers can find more details of the topic in [1].

U_i does not protect A_j. Our goal is to assign nonnegative integers to each nonzero (i,j) entry, such that the sum of the i-th column is equal to l_i, and the sum of the j-th row is no more than p_j. An example of an atomic assignment is shown in Fig. 4 for the configuration shown in Fig. 2 with $l_1 = l_2 = 2$, $l_3 = 3$, $p_1 = 3$ and $p_2 = 4$. Shown in [1], this assignment problem can be translated

	$A_1(l_1 = 2)$	$A_2(l_2 = 2)$	$A_3(l_3 = 3)$
$U_1(p_1 = 3)$	?(2)	0	?(1)
$U_2(p_2 = 4)$	0	?(2)	?(2)

Fig. 4. Assigning nonnegative integers to ? positions such that the sum of column i is equal to l_i, and the sum of row j is no more than p_j.

to finding maximum size matching in a Tanner graph, and the Edmonds-Karp algorithm can be applied to find the assignment with $O((\sum_i l_i)^2(\sum_j p_j))$ complexity.

III. Maximally Recoverable Property

Definition 1: A systematic erasure resilient code is said to have Maximally Recoverable (MR) property under a configuration Ω, if it can recover any $e \in E(\Omega)$.

MR codes can be interpreted as a series of Maximum Distance Separable (MDS) codes protecting atom sets $A_h, 1 \leq h \leq H$. The number of parity symbols protecting each atom set is dynamically assigned according to an atomic assignment. Consequently, each atom set can recover erasures up to the amount of parity assigned to it.

For instance, in the atomic assignment shown in Fig. 4, A_1 is assigned 2 parity symbols and can be considered as under the protection of a $(4,2)$ MDS code. As such, the 2 erasures in A_1 can be recovered. Similarly, A_2 and A_3 are assigned with 2 and 3 parity symbols, and can recover 2 and 3 erasures, respectively.

A natural question to ask is whether MR codes always exist under an arbitrary configuration Ω. The following theorem explores the answer.

Theorem 1: MR codes for any configuration Ω can be constructed with $O((n-k)k^3\binom{n-1}{k-1})$ complexity, if the size of Galois Field is larger than $\binom{n-1}{k-1}$.

Proof: For MR codes, its generator matrix G satisfies that for any $e \in E(\Omega)$, $G'(e)$ has rank k. We inductively construct one such MR generator matrix G for Ω.

Let $g_i, i = 1, \ldots, n$ be the i-th row of G. Let G_i be a matrix containing the first i rows of G, and the corresponding configuration be Ω_i. Since MR codes are a systematic code, G_k is merely a $k \times k$ identity matrix.

Suppose G_i is available, we construct one more row g_{i+1} to form G_{i+1}. Without loss of generality, let g_{i+1} correspond to a parity symbol $t_m^l \in U_l \subset V_l$.

For all $e \in E(\Omega_{i+1})$, $G_i'(e\backslash\{t_m^l\})$ must have rank $k-1$ or k. Otherwise, $G_{i+1}'(e)$ can not have rank k and e can not belong to $E(\Omega_{i+1})$. If $G_i'(e\backslash\{t_m^l\})$ has rank k, the value of t_m^l doesn't matter. Let us consider all $e \in E(\Omega_{i+1})$ that $G_i'(e\backslash\{t_m^l\})$ has rank $k-1$. These are the erasure patterns that t_m^l help in the recovery. Our goal is to choose g_{i+1} so that $G_{i+1}'(e)$ has rank k for all such erasure patterns. $G_{i+1}'(e)$ is given by:

$$G_{i+1}'(e) = \begin{pmatrix} G_i'(e\backslash\{t_m^l\}) \\ g_{i+1} \end{pmatrix}. \quad (1)$$

Let the null space of $G_i'(e\backslash\{t_m^l\})$ be $N(G_i'(e\backslash\{t_m^l\}))$. Because $G_i'(e\backslash\{t_m^l\})$ is of rank $k-1$, the null space is a nonzero row vector that is perpendicular to all row vectors in $G_i'(e\backslash\{t_m^l\})$. For $G_{i+1}'(e)$ to have rank k such that e is recoverable, it is sufficient to select g_{i+1} that is not orthogonal to $N(G_i'(e\backslash\{t_m^l\}))$, i.e. $\langle g_{i+1}, N(G_i'(e\backslash\{t_m^l\}))\rangle \neq 0$.

For each $e \in E(\Omega_{i+1})$ that does not contain t_m^l and $G_i'(e\backslash\{t_m^l\})$ has rank $k-1$, we compute all $N(G_i'(e\backslash\{t_m^l\}))$ and form a matrix out of the results by using each $N(G_i'(e\backslash\{t_m^l\}))$ as a row vector. Clearly, this matrix has a finite number of rows, and this number is bounded by $\binom{i}{k-1}$. Computing each $N(G_i'(e\backslash\{t_m^l\}))$ can be done with $O(k^3)$ complexity.

Let $f_1, \cdots, f_{u_l}$ be u_l row vectors that correspond to data symbols in U_l. Let $f_j, j = u_l+1, \cdots, J$ be the projection of j-th row vector onto the sub space $span(f_1, \cdots, f_{u_l})$, i.e., all coefficients other than those of the data symbols in U_l is set to be zero. Since g_{i+1} corresponds to a parity symbol $t_m^l \in U_l \subset V_l$, it is apparent that $g_{i+1} \in span(f_1, \cdots, f_{u_l})$. g_{i+1} needs to satisfy $\langle g_{i+1}, f_j\rangle \neq 0$ for $j = 1, \ldots, J$. Let $\varepsilon = [\varepsilon_1, \cdots, \varepsilon_{u_l}]^T$ and

$$g_{i+1} = \varepsilon_1 f_1 + \cdots + \varepsilon_{u_l} f_{u_l}, \quad (2)$$

then $\langle g_{i+1}, f_j\rangle = \sum_{m=1}^{u_l} \varepsilon_m \langle f_m, f_j\rangle = \sum_{m=1}^{u_l} \varepsilon_m f_{j,m}$, where $f_{j,m}$ is simply the m-th column coefficient of f_j. Writing the dot products in a $J \times u_l$ matrix form, we have

$$\begin{pmatrix} f_{1,1} & \cdots & f_{1,u_l} \\ \vdots & \ddots & \vdots \\ f_{J,1} & \cdots & f_{J,u_l} \end{pmatrix} \varepsilon = \begin{pmatrix} I_{u_l} \\ f_{u_l+1} \\ \cdots \\ f_J \end{pmatrix} \varepsilon \triangleq \begin{pmatrix} I_{u_l} \\ F \end{pmatrix} \varepsilon. \quad (3)$$

ε should be chosen to be nonzero and satisfying $F\varepsilon$ is nonzero in every row. Suppose the code is generated in $GF(q)$, each row constraint defines a plane to avoid in space $GF(q^{u_l})$, with the plane has q^{u_l-1} elements in $GF(q^{u_l})$. Since ε has $(q-1)^{u_l}$ nonzero choice, it is clear that if $q > J - u_l + 1$, then ε can have at least one satisfying choice. As J is bounded by $\binom{n-1}{k-1}$ and $u_l \geq 1$, a sufficient condition is then $q > \binom{n-1}{k-1}$. For example, if $n = 20$ and $k = 16$, then q needs to be no less than $4845 \approx 2^{12.4}$.

If such a ε exists, the following procedure can be applied to search for such ε. First we randomly select a nonzero ε. We calculate (3) and are done if there is no zero entry in all the rows. Otherwise, we tune $\varepsilon_1, \cdots, \varepsilon_{u_l}$ one by one. We first tune ε_1, for all null vectors f_j with $f_{j,1}$ not equal to zero, we calculate a value to prevent ε_1 from being as $(\sum_{m=2}^{u_l} \varepsilon_m f_{j,m})/f_{j,1}$. We then choose an arbitrary values of ε_1 that is not in the set of the values. We are guaranteed to find at least one surviving value in $GF(q)$ if $q > \binom{n-1}{k-1} + 1$. After we adjust ε_1, only those vectors f_j with $f_{j,1} = 0$ are not considered and there could still be zero entries in Eqn. (3). If so, we move on to ε_2. For each f_j with $f_{j,2} \neq 0$, we again calculate a value to prevent

ε_2 from being as $(\sum_{m=1,3,\cdots,u_l} \varepsilon_m f_{j,m})/f_{j,2}$, and choose from the remaining values for ε_2. After tuning ε_2, only those vectors f_j with $f_{j,1} = f_{j,2} = 0$ could result in zero entries in Eqn. (3). Thus, the tuning process reduces the number of rows in (3) with value zero. We repeat the step till we come to ε_{u_l}, or all rows in Eqn. (3) are nonzero. The resulting ε is the desired one. The worst case complexity to compute ε is $O(J\,k^2)$.

Following the induction, we can construct the generator matrix G, with complexity $O((n-k)k^3\binom{n-1}{k-1})$. ■

Two observations can be made based on Theorem 1. First, every erasure pattern $e \in E(\Omega)$ is in fact recoverable, by MR codes under Ω. Hence, the necessary condition for erasure patterns to be recovered under Ω, shown in Lemma 1, is also sufficient. Second, given an erasure pattern, existence of an atomic assignment is determined by the configuration. As such, the erasure recoverability of MR codes depends only on the configuration.

IV. Decoding of Any MR Codes

Given an erasure pattern e, of which we assume it includes data symbols $d_1, \ldots, d_l$. We are interested in recovering $d_1, \ldots, d_r$ out of them, $r \leq l$. Decoding is to choose p symbols $c_1, \ldots, c_p$, which are either parity symbols generated based on certain data symbols d_1 to d_m ($r \leq m \leq k$) or merely these data symbols if available, and form a $p \times m$ decoding matrix to recover the r lost data symbols by performing Gaussian elimination.

Define a decoding choice as a set of these p symbols. Define decoding overhead as $p - r$, i.e. the difference between the number of symbols to access to decode r data symbols, and that of accessing them directly if they are not lost.

There could be multiple choices to recover the r data symbols. One of them is straightforward decoding. It is to choose r parity symbols that can recover these lost data symbols, then combine with the available data symbols to perform the decoding. For example, we assume for the MR code shown in Fig. 2, there are 1 erasures in A_3 and no erasure elsewhere. One straightforward decoding choice is to recover the erasure using the $(11, 8)$ MDS code protecting S_1. Since 8 symbols need to be accessed to recover one erasure, the decoding overhead is 7. Another straightforward decoding choice is to recover the erasure using the $(13, 9)$ MDS code protecting S_2. This results in a decoding overhead of 8. Obviously, the decoding matrix of straightforward decoding is square, i.e. $p = m$.

Moreover, A_3 can be thought as under the protection of $(13, 6)$ MDS code. It is thus possible the 7 parity symbols can be combined to recover the erasure in A_3, with the coefficients corresponding to all other data symbols, except the interested one, cancel each other. If so, the decoding overhead is 6, which is less than the best of any straightforward decoding choice, and the decoding matrix is not a square one.

Nevertheless, different decoding choice can have different decoding overhead. In a wide range of storage applications, this decoding overhead could mean extra traffic over a large scale network or between multiple local servers, which limits the number of parallel accesses the system can support, and hence is desired to be minimized. We define the decoding choice with minimum decoding overhead as the *optimal* decoding choice.

A natural question to ask is how to find the optimal decoding choice, and how the simple straightforward decoding perform. We start by describing an interesting yet important property of MR codes, then present an algorithm to find the optimal decoding choice.

Theorem 2: The $p \times m$ decoding matrix of the optimal decoding choice is necessarily a full rank square matrix.

Proof: Any optimal decoding matrix must have rank p, i.e. all p rows are linearly independent; otherwise decoding can be done using less than p symbols. We prove $p < m$ is not possible, by contradiction.

Assume an MR code has an optimal decoding matrix that has $p < m$. Since d_1 to d_r are decodable, it is sufficient and necessary that certain r rows in the decoding matrix can be reduced to be a rank r sub-matrix by Gaussian elimination, with zero entries corresponding to $d_{r+1}, \ldots, d_m$. Assume these rows to be the first r rows, corresponding to $c_1, \ldots, c_r$.

Consider the $(p-r) \times (m-r)$ sub-matrix, denoted by A, formed by removing the first r rows and columns of the decoding matrix. Clearly A has rank $p - r$, all $p - r$ rows in A are linearly independent, and there exists an assignment from the corresponding $p - r$ parity symbols to $p - r$ data symbols $d_{r+1}, \ldots, d_p$. Any of c_1 to c_r can be expressed as a linear combination of c_{r+1} to c_p, and d_1 to d_r. We now construct an erasure pattern $e \in E(\Omega)$, that the code can not recover. We consider two cases.

If one of c_1 to c_r has nonzero coefficient corresponding to one of d_{p+1} to d_m, assumed to be c_1 and d_m respectively, then there exists an assignment from c_1 to d_m. Hence for an erasure pattern e with $d_{r+1}, \ldots, d_q, d_m$ lost and $c_1, c_{r+1}, \ldots, c_p$ remaining, an atomic assignment exists, thus $e \in E(\Omega)$. However, since c_1 is linearly dependent on $c_{r+1}, \ldots, c_p$ and the remaining data symbols, $G'(e)$ has only rank $k-1$. Hence, e can not be recovered by the code, and the code can not have MR property.

Now we consider the case where all c_1 to c_r have zero coefficients corresponding to d_{p+1} to d_m. Without loss of generality, assume $c_1, \ldots, c_r$ have zero coefficients corresponding to and only to $d_q, \ldots, d_m$ ($r + 2 \leq q \leq p + 1$). We further consider two sub-cases:

1) One of c_{r+1} to c_{q-1} has one nonzero coefficient corresponding to one of d_q to d_m, assuming to be c_{r+1} and d_m, respectively. By case setting, one of $c_1, \ldots, c_r$ has nonzero coefficient corresponding to d_{r+1}, assumed to be c_1. There exists an atomic assignment from $c_1, c_{r+1}, \ldots, c_p$ to $d_{r+1}, \ldots, d_p, d_m$, respectively. Following the same procedure as in the previous case, we can construct an $e \in E(\Omega)$ but the code fails to recover. Hence the code can not have MR property.

2) All c_{r+1} to c_{q-1} also have zero coefficients corresponding to all $d_q, \ldots, d_m$. Therefore, c_{r+1} to c_{q-1} is sufficient to reduce c_1 to c_r to a full rank $r \times r$ sub-matrix. If $q < p$

then c_q to c_p are not used in the reduction of c_1 to c_r, and they should not be in the optimal decoding matrix. If $q = p + 1$, then the decoding matrix contains zero-value columns, violating the assumption that it is not reducible.

In all cases, the assumption $p < m$ does not stand. Therefore, the optimal decoding matrix of MR codes must be square and have full rank. ■

There are several observations to make based on Theorem 2 for the optimal decoding. First, for every optimal $p \times p$ decoding matrix with l total lost data symbols involved, there must exist a size l assignment from parity symbols to the lost data symbols. Hence, this optimal $p-r$ decoding overhead can also be achieved by decoding using this l parity symbols and $p - l$ remaining data symbols. This indicates that any minimum decoding overhead can be achieved by straightforward decoding.

Second, if the optimal decoding matrix contain parity symbols from $\Lambda(A_h)$, then all data symbols in A_h must all involve in the decoding, i.e. they correspond to $|A_h|$ columns in the matrix. If no parity symbol in $\Lambda(A_h)$ is involved, then so do the data symbols in A_h. This implies that first either all data symbols in A_h can be recovered, or none of them can be recovered; second, p can only be the sum of the sizes of atom sets. This is key to design our algorithm on searching the optimal decoding choice, which will be presented shortly.

Third, the minimum decoding overhead depends only on Ω. This is because any decoding matrix of MR codes has full rank if and only if there exists an atomic assignment, which is determined by configuration.

Now we derive the algorithm for searching the optimal decoding choice. Let a partial configuration of Ω be $\Omega' \subseteq \Omega$; let $|\Omega'|$ be the number of data symbols in Ω'. Let $e_{\Omega'}$ be the projection of e onto Ω', containing all erasures in S_l or U_l, where $(S_l, U_l) \in \Omega'$. Let $e_0 \subseteq e$ be the set of data symbols we want to decode.

A sufficient and necessary condition for e_0 to be straightforwardly decoded is $e_0 \subseteq e_{\Omega'} \in E(\Omega')$ for some Ω'. This is because any decoding matrix for straightforwardly decoding e_0 is a full-rank square matrix with $|\Omega'|$ columns for certain Ω'. $e_{\Omega'}$ must also be decodable, since all data symbols in Ω' are recovered. Therefore, searching the optimal choice for decoding e_0 is equivalent to solve the following problem to find the Ω' with minimum $|\Omega'|$, which is denoted by Ω_0:

$$\begin{aligned} \Omega_0 \quad = \quad & \arg\min_{\Omega' \subseteq \Omega} |\Omega'| \\ & \text{s.t. } e_0 \subseteq e_{\Omega'} \in E(\Omega'). \end{aligned} \tag{4}$$

The minimum decoding overhead is merely $|\Omega_0|$.

Given an erasure pattern e, assume we want to recover $l_1, \ldots, l_r$ lost data symbols in $A_1, \ldots, A_r$ respectively. We describe one procedure to search for the optimal decoding choice in the following paragraph. The basic idea behind the algorithm is to start with Ω' with minimum possible $|\Omega'|$, and search over all possible Ω' by adding more and more protection groups into consideration. We prune out those Ω' whose size is larger than the minimum one we have observed during the search.

- For each assignment from $\Lambda(\cup_{i=1,\ldots,r} A_i)$ to $A_1, \ldots, A_r$, we construct $\Gamma_1 = \{A_1, \ldots, A_r\}$. We search all possible partial configurations that contain the assigned parity symbols and can decode e_0, through the following operations:
  1) Let $\Gamma_2 = \{A_h | \; A_h$ is protected by assigned parity symbols$\}$. If Γ_1 is equal to Γ_2, then decoding can be performed under current configuration, and one candidate of Ω_0 is founded. We update ω, the minimum decoding overhead seen so far, with $\sum_{A_i \in \Gamma_1} |A_i| - \sum_{i=1}^{r} l_i$.
  2) If $\Gamma_1 \subset \Gamma_2$ and $\omega > \sum_{A_i \in \Gamma_2} |A_i| - \sum_{i=1}^{r} l_i$, then we set $\Gamma_1 = \Gamma_2$, find all atomic assignments for atom sets in Γ_1, and go through steps 1) and 2) with the augmented Γ_1 recursively.
- After all possible atomic assignments are attempted, either we have found the minimum decoding overhead ω and the corresponding decoding choice, or the data symbols we want to recover is not decodable.

The above decoding algorithm works for any MR codes, as it explores only the configuration information.

V. Conclusions

In this paper, we establish the concept of MPG codes and study the MR property for MPG codes. MPG codes with MR property achieve the best erasure recoverability under the configuration. We show that configuration is the only parameter to tune both the recoverability and the minimum decoding overhead of MR codes. We also present construction and decoding algorithms for MPG codes with MR property.

Acknowledgement

We would like to thank Dr. Yunnan Wu for helpful discussions related to Theorem 2.

References

[1] C. Huang, M. Chen and J. Li, "Pyramid Codes: Flexible Schemes to Trade Space for Access Efficiency in Reliable Data Storage Systems", (to appear) *IEEE NCA 2007*, Cambridge, MA, Jul. 2007.

[2] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, L. M. G. M. Tolhiuzen, "Polynomial time algorithms for multicast network code construction", *IEEE Transactions on Information Theory*, 51 (6). June 2005, pp. 1973-1982

[3] S. Lin, and D. J. Costello, "Error Control Coding, Fundamentals and Applications", Prentice Hall Press, 2004.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes", *ICC 1993*, Geneva, Switzerland, May. 1993.

[5] R. G. Gallager, "Low-Density Parity-Check Codes". Cambridge, MA: M.I.T. Press, 1963.

[6] L. Lentmaier and K. Ziganggirov, "Iterative decoding of generalized low-density parity-check codes", *ISIT 1999*, Cambridge, MA, Aug. 1999.

[7] G. Yue, P. Li, and X. Wang, "Low-Rate Generalized Low-Density Parity-Check Codes with Hadamard Constraints", *ISIT 2005*, pp.1337-1381, Australia, 4-9 Sept. 2005.

[8] G. Richter, S. Stiglmayr, and M. Bossert, "Optimized Asymptotic Puncturing Distributions for Different LDPC Code Constructions", *ISIT 2006*, Seattle, WA, July, 2006

[9] S. Freundlich, D. Burshtein and S. Litsyn, " Approximately Lower Triangular Ensembles of LPDC Codes with Linear Encoding Complexity", *ISIT 2006*, Seattle, WA, July, 2006

[10] R.A. Horn, C.R. Johnson, "Matrix Analysis", Cambridge University Press.