A-29

Cluttered Orderings for two-dimensional Parity Codes

Tomoko Adachi† Meinard Müller† Hiroaki Uehara† Masakazu Jimbo†

Keywords: RAID, disk arrays, erasure codes, cluttered orderings, mathematical modeling.

1. Introduction

The desire to speed up secondary storage systems has lead to the development of *disk arrays* which achieve performance through disk parallelism. While performance improves with increasing numbers of disks the chance of data loss coming from *catastrophic failures*, such as head crashes and failures of the disk controller electronics, also increases. To avoid high rates of data loss in large disk arrays one includes redundant information stored on additional disks – also called *check disks* – which allows the reconstruction of the original data – stored on the so-called *information disks* – even in the presence of disk failures. These disk array architectures are known as *redundant arrays of independent disks* (RAID) [3].

2. Erasure Codes

Erasure-correcting capability of disk arrays can be achieved by employing *erasure codes*. Let $k, c \in \mathbb{N}$ and let GF(2) denote the field with two elements. Then an erasure code is defined by a linear injection $\gamma : \mathrm{GF}(2)^k \to \mathrm{GF}(2)^{k+c}$ such that an information $x \in \mathrm{GF}(2)^k$ appears unchanged in the first k bits – the so-called *information bits* – of the corresponding code vector $\gamma(x)$. The remaining c bits are referred to as *check bits* which can be computed as the parity of subsets of information bits. Each such code can be defined in terms of a $c \times (k+c)$-*parity check matrix*, $H = [C|I]$, where I denotes the $c \times c$ identity matrix and C is a $c \times k$ matrix. The codewords in the code are the vectors $y \in \mathrm{GF}(2)^{k+c}$ satisfying the equation $Hy = 0$. Note, that the first k columns of H correspond to the information bits and the last c columns to the check bits (see also Fig. 1).

An unreadable bit of a code vector is called an *erasure*. It is a well known fact that a code can correct a set of t erasures iff the corresponding t columns of H are linearly independent considered as vectors over GF(2) [3]. An erasure code which can correct any t erasures will be abbreviated as t-EC. In view of the RAID-application there are the following two important metrics in erasure codes. One metric is the *update penalty*, which is the number of check disks whose contents must be changed if an information disk is changed. In terms of the matrix H it can be defined as the maximum over the weights of the columns of H. It follows easily that the update penalty of an t-EC is at least t. Another metric is the *check bit overhead*, which is the ratio c/k of the number of check bits to information bits. Good erasure codes have high erasure correcting capabilities, whereas the update penalty as well as the check bit overhead should be low. (See [3] for further details.)

†Mathematics, Keio University

3. Two-dimensional Parity Codes

In the rest of the paper we will investigate some specific erasure codes. Let $k = \ell^2$ for some $\ell \in \mathbb{N}$, so that the k information bits can be arranged in a two dimensional array. Associate to each row and each column a check bit containing the parity of that row or column, i.e., $c = 2\ell$. The so defined code is called 2-*dimensional parity code* which is easily seen to be a 2-EC with an optimal update penalty 2. In [3] is proved that this code can even correct most of the 3-erasures as well and has – with respect to this even higher erasure-correcting capability – optimal check disk overhead among all such codes. As illustration we present the case $\ell = 2$. For example, information disk 1 is associated to the check disks a and c.

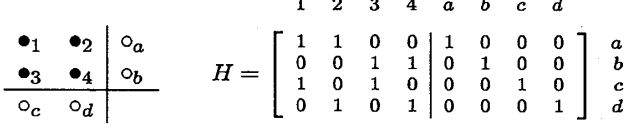

Fig. 1: 2-dim. parity code and its parity check matrix.

A 2-dimensional parity code can be modeled by the complete bipartite graph $K_{\ell,\ell} = (U, V, E)$ in the following way. The point set of $K_{\ell,\ell}$ is partitioned into the two sets U and V both having cardinality ℓ. Assign the points of U to the ℓ check bits corresponding to the rows and the points of V to the ℓ check bits corresponding to the columns. By definition, in $K_{\ell,\ell}$ any point of U is connected with any point of V by exactly on edge constituting the edge set E, i.e., $|E| = \ell^2$.

Now, any of the $k = \ell^2$ information bits can be mapped to exactly one edge of E determined by the row and column of the corresponding check bits assigned to the points of the edge. This graph-theoretical modeling of the parity codes will be helpful for our following investigations.

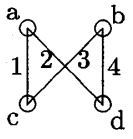

Fig. 2: Code as graph.

4. Cluttered Orderings

In a RAID system disk writes are expensive operations and should therefore be minimized. When writing on a single information disk one also has to recompute the parity information and change the contents of all check disks involved. This overhead is expressed by the update penalty as defined in Section 2. In many applications there are writes on a small fraction of consecutive disks – say d disks – where d is small in comparison to k, the number of information disks. In this case a write can be implemented as an efficient *read-modify-write* which can be described as follows [2]. First the d information disks are read followed by all of their associated check disks. In the case when check disks overlap, the physical read only takes place once. All of the new parity is computed and then this new parity and the new information is written back to the disks. Once again, the shared check disks are only physically written once. Therefore, to minimize the number of operations when writing

to d consecutive information disks one has to minimize the number of check disks – say f – associated to the d information disks. In other words, the order of the information disks – or the order of the corresponding columns of the parity check matrix – plays a crucial role for the efficiency of the RAID system. In view of the interpretation of 2-dimensional parity codes as bipartite graphs, we will use a graph-theoretical approach to model this problem similar to [2].

Let $G = (V, E)$ be a graph with $n = |V|$ vertices and edge set $E = \{e_0, e_1, \cdots, e_{m-1}\}$. Let $d \leq m$ be a positive integer, called a *window* of G, and π a permutation on $\{0, 1, \cdots, m-1\}$, called an *edge ordering* of G. Then, given a graph G with edge ordering π and window d, we define $V_i^{\pi,d}$ to be the set of vertices which are connected by an edge of $\{e_{\pi(i)}, e_{\pi(i+1)}, \cdots, e_{\pi(i+d-1)}\}$, $0 \leq i \leq m-1$, where indices are considered modulo m. The cost of accessing a subgraph of d consecutive edges is measured by the number of its vertices. An upper bound of this cost is given by the *d-maximum access cost* of G defined as $\max_i |V_i^{\pi,d}|$. An ordering π is a *(d, f)-cluttered ordering*, if it has d-maximum access cost equal to f. In view of the RAID-application one is interested in minimizing the parameter f. The following example shows a $(3, 4)$-cluttered ordering of $K_{3,3}$.

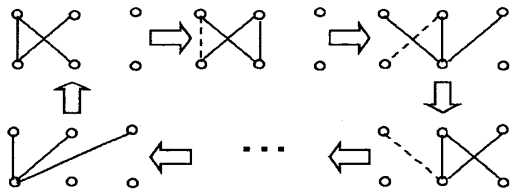

Fig. 3: $(3, 4)-$ cluttered ordering of $K_{3,3}$.

5. Cluttered Orderings for 2-dim. Parity Codes

Cohen, Colbourn and Froncek [2] investigated the full 2-code and describe a cyclic construction for a cluttered ordering of the complete graph corresponding to the full 2-code. In the following, we continue our investigation of the 2-dimensional parity code and describe how to construct a (d, f)-cluttered ordering of the complete bipartite graph $K_{\ell,\ell}$ with some small f. There are two fundamental concepts involved. Firstly, we introduce the notion of a *wrapped ρ-labelling* for a bipartite graph. Secondly, we define the concept of an *f-movement*.

Let $H = (U, V, E)$ be a bipartite graph and $\ell = |E|$. A pair of mappings (ρ_U, ρ_V), $\rho_U : U \to \mathbb{Z}_\ell$, $\rho_V : V \to \mathbb{Z}_\ell$, is said to be a *wrapped ρ-labelling* of H, if there are non-empty subsets $X, Y \subset U$, $X', Y' \subset V$ and an $i \in \mathbb{Z}_\ell$ coprime to ℓ satisfying condition (i) and either condition (ii) or condition (ii)':

(i) Each element of $\mathbb{Z}_\ell$ occurs exactly once in $\{\rho_U(x) - \rho_V(y) \pmod{\ell} \mid x \in U, y \in V, (x, y) \in E\}$

(ii) $\rho_U(X') = \rho_U(X) + i$ and $\rho_V(Y') = \rho_V(Y) + i \pmod{\ell}$.

(ii)' $\rho_U(X') = \rho_V(Y) + i$ and $\rho_V(Y') = \rho_U(X) + i \pmod{\ell}$.

If one takes ℓ copies of H, say $H(i)$, $0 \leq i < \ell$, provides each of them with a labelling $\rho(i) := \rho + i \cdot \ell$, and identifies vertices with the same label, then it is not hard to show that one gets a partition of the complete bipartite graph $K_{\ell,\ell}$ into ℓ isomorphic copies of H. We refer to [1] for further details.

Next, we introduce the concept of an *f-movement*. Let G be an bipartite graph partitioned into two isomorphic bipartite subgraphs $H = (U, V, E)$ and $H' = (U', V', E')$ with $|U| = |U'|$, $|V| = |V'|$, $E = \{e_0, e_1, \cdots, e_{\ell-1}\}$, and $E' = \{e'_0, e'_1, \cdots, e'_{\ell-1}\}$. Given two permutations π_- and π_+ of the index set $\{0, 1, \ldots, \ell-1\}$, one can define the following $\ell+1$ bipartite subgraphs of G. Let $H_0 := H$ and inductively $H_i := (U_i, V_i, E_i), 1 \leq i \leq \ell$, $E_i := (E_{i-1} \setminus \{e_{\pi_-(i-1)}\}) \cup \{e'_{\pi_+(i-1)}\}$. Let U_i be the set of vertices in $U \cup U'$ contained in some edge of E_i and, similarly, V_i the set of vertices in $V \cup V'$ contained in some edge of E_i. It is easy to see that $H_\ell = H'$. Then the pair of permutations (π_-, π_+) is called an *f-movement* from H to H' if $\max_{0 \leq i \leq \ell}(|U_i| + |V_i|) = f$, where $f := |U| + |V|$.

Now, the next theorem gives some sufficient conditions for the existence of a (d, f)-cluttered ordering of $K_{\ell,\ell}$. We refer to [1] for a proof.

Theorem 5.1. *If there is a wrapped ρ-labelling of a bipartite graph $H = (U, V, E)$ with $d = \ell = |E|$ and $f = |U| + |V|$ and if there is an f-movement from $H(0)$ to $H(1)$ as defined above, then there exists a (d, f)-cluttered ordering.*

We conclude this paper with some specific construction. For $h_1, h_2 \in \mathbb{N}$, we define the bipartite graph $H(h_1, h_2) = (U, V, E)$ by $U = \{u_i : i = 1, 2, \ldots, h_1 + h_2\}$, $V = \{v_i : i = 1, 2, \ldots, h_2 + h_1\}$ and $E = \{\{u_i, v_j\} : 1 \leq i \leq h_1, 1 \leq j \leq h_2\} \cup \{\{u_i, v_{h_2+j}\} : 1 \leq j \leq i \leq h_1\} \cup \{\{u_{h_1+i}, v_j\} : 1 \leq i \leq j \leq h_2\}$. In [1] we show that there is a suitable f-movement, $f = 2(h_1 + h_2)$, such that one gets the following theorem.

Theorem 5.2. *If there is a wrapped ρ-labelling for $H(h_1, h_2)$, then there exists a (d, f)-cluttered ordering for the complete bipartite graph $K_{\ell,\ell}$, where $d = (h_1 + h_2)(h_1 + h_2 + 1)/2$ and $f = 2(h_1 + h_2)$.*

In [1] we give some examples for wrapped ρ-labellings for small values h_1 and h_2 leading to (d, f)-cluttered orderings by Theorem 5.2. Finding suitable wrapped ρ-labellings leading to small f is an ongoing research project of the authors.

6. Conclusion

Starting with the RAID application we have introduced erasure codes. Of special interest are the two-dimensional parity codes which can be modeled by the complete bipartite graph. Minimizing disk operations when writing to consecutive disks lead to the concept of cluttered orderings. Using some appropriate mathematical modeling – the ρ-labellings and f-movements – we gave some sufficient conditions for the existence of such orderings and concluded with some example. This paper gives some nice example for the interaction between real applications and mathematics.

7. References

[1] T. Adachi and M. Jimbo, Constructions of a cluttered ordering for the complete bipartite graph, in preparation.

[2] M. Cohen, C. Colbourn and D. Froncek (2001), Cluttered orderings for the complete graph, *COCOON 2001*, LNCS **2108**, Springer, 420-431.

[3] L. Hellerstein, G. Gibson, R. Karp, R. Katz and D. Patterson (1994), Coding techniques for handling failures in large disk arrays, *Algorithmica*, **12**, 182-208.