On the Equivalence Between the B-Code Constructions and Perfect One-Factorizations

Mingqiang Li and Jiwu Shu
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Email: lmq06@mails.tsinghua.edu.cn, shujw@tsinghua.edu.cn

***Abstract*—The B-Code is a class of MDS array code with optimal properties for RAID 6. Xu et al. proposed an open problem in 1999: Are the B-Code constructions strongly equivalent to perfect one-factorizations of a graph? In this paper, we show that the constructions of the B-Code of length l are equivalent to perfect one-factorizations of a l-regular graph on $l+r$ vertices, where l is an integer not smaller than 4, and r is equal to 1 for an odd l or 2 for an even l.**

I. Introduction

Fault tolerance is an important concern in the design of storage systems [1]. As today's storage systems grow in size and complexity, they are increasingly confronted with disk failures [2], [3] together with latent sector errors [4]. RAID (Redundant Array of Inexpensive Disks) [5] is a promising technique to cope with this challenge. Among all levels of RAID, RAID 5 is widely used in modern storage systems to recover one disk failure. However, in a RAID 5 subsystem, another disk failure or a latent sector error that occurs on another disk can lead to permanent data loss. Under this background, RAID 6 that can tolerate two disk failures is expected by storage system designers.

RAID 6 is designed based on a Maximum-Distance Separable (MDS) code of distance 3. Here, a MDS code attains the Singleton bound [6] and thus has optimal *storage efficiency* (i.e. the ratio of user data to the total of user data plus redundancy data). There are two categories of erasure codes that can be used in RAID 6: Reed-Solomon codes [7], [8] and array codes [9]. The encoding and decoding processes of Reed-Solomon codes involve complex finite-field operations, while the encoding and decoding processes of array codes use only simple XOR (Exclusive OR) operations. Thus, array codes are more efficient than Reed-Solomon codes in terms of computation complexity and are preferred in the design of RAID 6.

One crucial parameter of array codes for RAID 6 is the *update complexity*, which is defined as the average number of parity bits affected by a change of a single information bit in array codes. The update complexity of a code can significantly affect the write performance of the corresponding RAID 6, especially for small writes. It is well-known that the lower bound of the update complexity of an array code of distance 3 is 2. For MDS array codes of distance 3 with separate information and parity columns (such as EVENODD [10], RDP Codes [11], and Libration Codes [12]), it was proved in [13] that their update complexity is always strictly larger than 2. Then, a natural question is whether the update complexity of 2 is achievable for MDS array codes of distance 3. A positive answer to this question was given by several previous work [14]–[17]. The MDS array codes of distance 3 proposed in these work, including ZZS Codes [14], X-Code [15], B-Code [16], and P-Code [17], combine information and parity bits within columns in order to achieve optimal update complexity, i.e. 2 for distance 3. Among these array codes, the B-Code [16] is the representative one, which has optimal *length* (i.e. the number of columns), twice of that of X-Code [15] with the same column size. Here, it should be noted that both ZZS Codes [14] and P-Code [17] are included in the class of the B-Code. In current literature, however, the B-Code has been constructed only for some special families of lengths larger than 4, including $p-1$ and p in [14] and [17] and $2p-2$ and $2p-1$ in [16], where p is an odd prime. The B-Code constructions for other lengths are still unknown.

Definition 1 ([18]): A *one-factorization* of a graph is a partitioning of the set of its edges into subsets such that each subset is a graph of degree one. Here, each subset is called as a *one-factor*. A *perfect one-factorization* is a particular one-factorization in which the union of any pair of one-factors forms a Hamiltonian cycle.

(*Remark:* A Hamiltonian cycle is a cycle in an undirected graph, which visits each vertex exactly once and also returns to the starting vertex.)

Xu et al. proved in [16] that the constructions of the B-Code of length $2n+1$ are equivalent to perfect one-factorizations of a $(2n+1)$-regular graph on $2n+2$ vertices (i.e. a complete graph on $2n+2$ vertices), where n is an integer not smaller than 2. They also proposed an open problem in [16]: *Are the B-Code constructions strongly equivalent to perfect one-factorizations of a graph?* If a positive answer can be given to this open problem, the problem of the B-Code constructions in coding theory can then be completely converted into the well-known problem of perfect one-factorizations in graph theory, and many results on perfect one-factorizations can be used for the B-Code constructions.

In this paper, we will show in Section III that the construc-

This work was supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 60925006) and the National High Technology Research and Development Program of China (Grant No. 2009AA01A403).

978-1-4244-7892-7/10/$26.00 ©2010 IEEE

tions of the B-Code of length $2n$ are equivalent to perfect one-factorizations of a $2n$-regular graph on $2n+2$ vertices. Thus, we have the following result:

Theorem 1: The constructions of the B-Code of length l are equivalent to perfect one-factorizations of a l-regular graph on $l+r$ vertices, where l is an integer not smaller than 4, and r is equal to 1 for an odd l or 2 for an even l.

Before giving a proof on the equivalence, we first provide a brief introduction to the B-Code and its graph description in the next section.

II. The B-Code and Its Graph Description

The algebraic definition of the B-Code is given as follows:

Definition 2 (B-Code [16]): Let $\mathbf{H}_l = (H_1\ H_2\ \cdots\ H_l)$ be a binary matrix, where $l = 2n$ or $l = 2n+1$, and $H_k = (h_{i,j})_{2n\times n}$ is a binary submatrix of size $2n \times n$, for $1 \le i \le 2n$, $1 \le j \le n$, and $1 \le k \le l$. Suppose $\mathbf{H}_l$ meets the following three conditions:

1) $h_{i,ni} = 1$ $(1 \le i \le 2n)$;
2) the *weight* (i.e. the number of 1's) of each row of $\mathbf{H}_l$ is $l-1$; and
3) for any m and k (where $1 \le m, k \le l$ and $m \ne k$), the square matrix $(H_m\ H_k)$ is nonsingular.

If a code's parity-check matrix is $\mathbf{H}_l$, the code is then called as the B-Code, denoted by B_l.

Take B_4 and B_5 for example. Two parity-check matrices for B_4 and B_5 are as follows:

$$\mathbf{H}_4 = \left(\begin{array}{cc|cc|cc|cc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right), \qquad (1)$$

$$\mathbf{H}_5 = \left(\begin{array}{cc|cc|cc|cc|cc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}\right). \qquad (2)$$

In the following discussion, we will use the above two examples of B_4 and B_5.

It was shown in [16] that the B-Code is a class of array code of distance 3, which has the following optimal properties:

1) it is Maximum-Distance Separable (MDS);
2) its update complexity is 2, which is the minimum update complexity that MDS codes of distance 3 can have; and
3) it achieves the maximum length that MDS codes with optimal update complexity can have.

In structure, the B-Code is an array code of size $n \times l$, i.e. with n rows and l columns, where $l = 2n$ or $l = 2n+1$. It was proved in [14] that this size has optimal length. For B_{2n}, the first $n-1$ rows are information rows, and the last row is a parity row. In other words, the bits in the first $n-1$ rows are information bits, while those in the last row are parity bits. The structure of B_{2n+1} is derived from that of B_{2n} by adding one more information column as the last column. Figure 1 illustrates the structures of B_{2n} and B_{2n+1}.

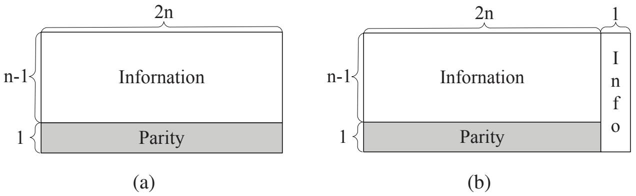

Fig. 1. Structures of (a) B_{2n} and (b) B_{2n+1} [16].

$d_{2,3}$	$d_{3,4}$	$d_{4,1}$	$d_{1,2}$
p_1	p_2	p_3	p_4

(a)

$d_{2,3}$	$d_{3,4}$	$d_{4,1}$	$d_{1,2}$	$d_{1,3}$
p_1	p_2	p_3	p_4	$d_{2,4}$

(b)

Fig. 2. Array representations of (a) B_4 and (b) B_5.

In the B-Code, because of its optimal update property, each information bit contributes to the calculation of (or is protected by) exactly 2 parity bits contained in other columns. Moreover, any two information bits do not contribute to the calculation of the same pair of parity bits. These characteristics of the B-Code are illustrated in Figure 2, which gives the array representations of B_4 and B_5. In this figure, the $d_{i,j}$'s (where $1 \le i, j \le 4$ and $i \ne j$) are the information bits, while the p_k's (where $1 \le k \le 4$) are the parity bits. They are related by

$$p_k = \sum_{k \in \{i,j\}} d_{i,j}, \qquad (3)$$

where '$\sum$' denotes the binary sum operation.

Because of the above characteristics of the B-Code, a graph approach is introduced in [16] to describe the B-Code. In the graph description of the B-Code, each parity bit is represented by a vertex, and each information bit that contributes to the calculation of 2 parity bits is represented by an edge that connects the two corresponding vertices. Then, B_{2n} can be described using a $(2n-2)$-regular graph on $2n$ vertices, and B_{2n+1} can be described using a $(2n-1)$-regular graph on $2n$ vertices (i.e. a complete graph on $2n$ vertices). In the cases of both B_{2n} and B_{2n+1}, we label the $2n$ vertices with integers from 1 to $2n$. For each column of B_{2n}, its $n-1$ information bits are represented by $n-1$ edges that are not adjacent to each other and are not incident to the vertex corresponding to the parity bit contained in the column. We label all these $n-1$ edges with the same integer that is labeled on the vertex corresponding to the parity bit contained in the column. This is the same for the first $2n$ columns of B_{2n+1}. For the last column of B_{2n+1}, its n information bits are represented by n edges, the union of which forms a one-factor of the graph on $2n$ vertices. We label all these n edges with ∞. Figure 3 shows the graph representations of B_4 and B_5.

Recall that the B-Code is a class of array code of distance 3, so it can recover the erasure of any two columns. This is guaranteed by the third condition of Definition 2. In the graph

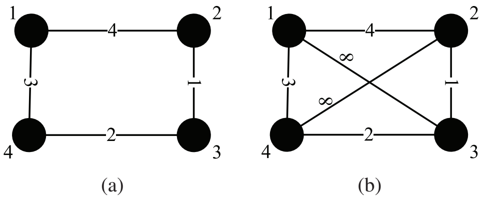

Fig. 3. Graph representations of (a) B_4 and (b) B_5.

description of the B-Code, this condition is equivalent to the following one:

Condition 1: For any pair of columns of the B-Code, the union (denoted by G^*) of the two corresponding subgraphs satisfies:

1) G^* does not contain a cycle; and
2) if the two columns both contain a parity bit, G^* then does not contain a path whose terminal vertices are the two vertices corresponding to the two parity bits contained in the two columns.

The above condition is explained by contradiction as follows.

We consider the first opposite case where G^* contains a cycle of length r. In such a cycle, suppose the r edges are $e_1, e_2, \cdots, e_r$. As we know, in the corresponding square matrix mentioned in the third condition of Definition 2, the column vector corresponding to each edge is a vector of weight 2, whose two 1's are in the two rows corresponding to the two vertices of the edge. Then, in the square matrix, the binary sum of the r column vectors corresponding to $e_1, e_2, \cdots, e_r$ is a zero vertical vector, which conflicts with the nonsingular property of the square matrix. Thus, G^* should not contain a cycle.

If the two columns both contain a parity bit, we then consider the second opposite case where G^* contains a path of length r' whose terminal vertices are the two vertices corresponding to the two parity bits contained in the two columns. In such a path, suppose the two terminal vertices are ${v'}_1$ and ${v'}_2$, and the r' edges are ${e'}_1, {e'}_2, \cdots, {e'}_{r'}$. As we know, in the corresponding square matrix mentioned in the third condition of Definition 2, the column vector corresponding to the terminal vertex ${v'}_1$ (or ${v'}_2$) is a vector of weight 1, whose only 1 is in the row corresponding to the vertex ${v'}_1$ (or ${v'}_2$). Then, in the square matrix, the binary sum of the $r'+2$ column vectors corresponding to the two terminal vertices ${v'}_1$ and ${v'}_2$ and the r' edges ${e'}_1, {e'}_2, \cdots, {e'}_{r'}$ is a zero vertical vector, which conflicts with the nonsingular property of the square matrix. Thus, G^* should not contain a path whose terminal vertices are the two vertices corresponding to the two parity bits contained in the two columns.

III. Proof on the Equivalence

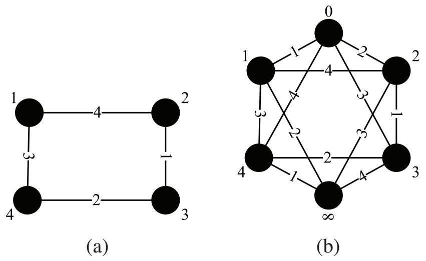

Fig. 4. Constructing (a) B_4 from (b) a perfect one-factorization of a 4-regular graph on 6 vertices.

Xu et al. [16] have proved that the B_{2n+1} constructions are equivalent to perfect one-factorizations of a $(2n+1)$-regular graph on $2n+2$ vertices (i.e. a complete graph on $2n+2$ vertices). Thus, we only need to consider the B_{2n} constructions in this section.

We will show that the B_{2n} constructions are equivalent to perfect one-factorizations of a $2n$-regular graph on $2n+2$ vertices by the following two algorithms.

Algorithm 1: Constructing B_{2n} from a Perfect One-Factorization of a $2n$-Regular Graph on $2n+2$ Vertices

(S1) Choose arbitrary pair of vertices that are not adjacent to each other in the regular graph and label them with 0 and ∞. Then, label the other $2n$ vertices of the regular graph with integers from 1 to $2n$.

(S2) If a perfect one-factorization exists for the regular graph, then let F_i denote the one-factor that contains the edge $\{0, i\}$, where $i = 1, 2, \ldots, 2n$.

(S3) In each F_i, delete the two vertices 0 and ∞ and all the edges that are incident to these two vertices. For $i = 1, 2, \ldots, 2n$, label all the remaining edges in F_i with i.

It can be proved as follows that Algorithm 1 constructs a $(2n-2)$-regular graph on $2n$ vertices, which meets Condition 1 in Section II. Thus, a corresponding B_{2n} is constructed by Algorithm 1.

According to Definition 1 in Section I, in a perfect one-factorization, for any pair of one-factors F_{i_1} and F_{i_2}, the union of them forms a Hamiltonian cycle. Then, in the union of F_{i_1} and F_{i_2}, after we delete all the edges that are incident to the two vertices 0 and ∞, no cycle can exist. In addition, there also does not exist a path whose terminal vertices are the two vertices i_1 and i_2, otherwise the union of the path and the two edges $\{0, i_1\}$ (contained in F_{i_1}) and $\{0, i_2\}$ (contained in F_{i_2}) can form a cycle that does not visit the vertex ∞, which conflicts with the fact that the union of the two one-factors F_{i_1} and F_{i_2} forms a Hamiltonian cycle. Thus, the $(2n-2)$-regular graph on $2n$ vertices constructed in Algorithm 1 meets Condition 1 in Section II.

To make Algorithm 1 more easily understood, we give an example of constructing B_4 from a perfect one-factorization of a 4-regular graph on 6 vertices in Figure 4.

Then, the next natural question is: Can we get a perfect one-factorization of a $2n$-regular graph on $2n+2$ vertices from B_{2n}? A positive answer to this question will be given by the following algorithm.

Algorithm 2: Constructing a Perfect One-Factorization of a $2n$-Regular Graph on $2n+2$ Vertices from B_{2n}

(S1) If B_{2n} exists, use the graph description of B_{2n} mentioned in Section II and let $\widetilde{F}_i$ denote the set of edges with the label i, where $i = 1, 2, \ldots, 2n$.

(S2) Add two vertices 0 and ∞ to the $(2n-2)$-regular graph of vertices $1, 2, \ldots, 2n$.

(S3) For $i = 1, 2, \ldots, 2n$, add the two edges $\{0, i\}$ and $\{k_i, \infty\}$ to $\widetilde{F}_i$, where k_i is an integer from 1 to $2n$ such that the expanded set F_i is a one-factor of the $2n$-regular graph of vertices $0, 1, 2, \ldots, 2n, \infty$.

From Condition 1 in Section II, we can deduce that for any pair of columns of B_{2n}, the union (denoted by G^*) of the two corresponding subgraphs can be in one of the following two forms:

1) G^* consists of an isolated vertex corresponding to a parity bit contained in the two columns and a path of length $2n-2$ one of whose terminal vertices is the vertex corresponding to the other parity bit contained in the two columns; or
2) G^* consists of two paths that satisfy: i) the sum of their length is $2n-2$, and ii) one of the terminal vertices of each path is a vertex corresponding to a parity bit contained in the two columns.

Thus, in the one-factorization constructed in Algorithm 2, the union of any pair of one-factors forms a Hamiltonian cycle. According to Definition 1 in Section I, Algorithm 2 constructs a perfect one-factorization of a $2n$-regular graph on $2n+2$ vertices.

In this section, we have shown that the B_{2n} constructions are equivalent to perfect one-factorizations of a $2n$-regular graph on $2n+2$ vertices. Combining our result on the B_{2n} constructions with the result on the B_{2n+1} constructions given by Xu et al. in [16], we can deduce that the B_l constructions are equivalent to perfect one-factorizations of a l-regular graph on $l+r$ vertices, where r is equal to 1 for an odd l or 2 for an even l. This gives a proof of Theorem 1 in Section I.

IV. Conclusion and Remarks

In this paper, we showed an equivalence between the constructions of the B-Code of length l and perfect one-factorizations of a l-regular graph on $l+r$ vertices, where l is an integer not smaller than 4, and r is equal to 1 for an odd l or 2 for an even l.

In graph theory, it is obvious that a perfect one-factorization of a $2n$-regular graph on $2n+2$ vertices can be obtained from a perfect one-factorization of a $(2n+1)$-regular graph on $2n+2$ vertices (i.e. a complete graph on $2n+2$ vertices) by deleting arbitrary one of its one-factors. However, intuitively, we believe that the reverse proposition is not always true. Then, based on the result obtained in this paper, we make a conjecture, which conflicts with that made in [16], as follows:

Conjecture 1: For some positive integer n not smaller than 2, the B-Code (or its dual [16]) of length $2n+1$ cannot be constructed from the B-Code (or its dual) of length $2n$.

To date, we have not found a way of proving the above conjecture, but leave it as an open problem.

References

[1] M. Li, J. Shu, and W. Zheng, "GRID codes: Strip-based erasure codes with high fault tolerance for storage systems," *ACM Transactions on Storage*, vol. 4, no. 4, pp. 1–22, Jan. 2009.

[2] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07)*, San Jose, CA, Feb. 2007, pp. 17–28.

[3] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" in *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07)*, San Jose, CA, Feb. 2007, pp. 1–16.

[4] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," in *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'07)*, San Diego, CA, Jun. 2007, pp. 289–300.

[5] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High-performance, reliable secondary storage," *ACM Computing Surveys*, vol. 26, no. 2, pp. 145–185, Jun. 1994.

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York: North-Holland, 1977.

[7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[8] R. M. Roth and A. Lempel, "On MDS codes via Cauchy matrices," *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1314–1319, Nov. 1989.

[9] M. Blaum, P. Farrell, and H. van Tilborg, "Array codes," in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier Science B.V., 1998.

[10] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[11] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST'04)*, San Francisco, CA, Mar. 2004, pp. 1–14.

[12] J. S. Plank, "The RAID-6 liberation codes," in *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST'08)*, San Jose, CA, Feb. 2008, pp. 97–110.

[13] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 529–542, Mar. 1996.

[14] G. V. Zaitsev, V. A. Zinov'ev, and N. V. Semakov, "Minimum-check-density codes for correcting bytes of errors, erasures, or defects," *Problems of Information Transmission*, vol. 19, no. 3, pp. 197–204, 1983.

[15] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 272–276, Jan. 1999.

[16] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS codes and factors of complete graphs," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1817–1826, Sep. 1999.

[17] C. Jin, H. Jiang, D. Feng, and L. Tian, "P-code: A new RAID-6 code with optimal properties," in *Proceedings of the 23rd International Conference on Supercomputing (ICS'09)*, Yorktown Heights, NY, Jun. 2009, pp. 360–369.

[18] W. D. Wallis, *One-Factorizations*. Norwell, MA: Kluwer, 1997.