Shortening Array Codes and the Perfect 1-Factorization Conjecture

Vasken Bohossian and Jehoshua Bruck, *Fellow, IEEE*

***Abstract*—The existence of a perfect 1-factorization of the complete graph with n nodes, namely, K_n, for arbitrary even number n, is a 40-year-old open problem in graph theory. So far, two infinite families of perfect 1-factorizations have been shown to exist, namely, the factorizations of K_{p+1} and K_{2p}, where p is an arbitrary prime number $(p > 2)$. It was shown in previous work that finding a perfect 1-factorization of K_n is related to a problem in coding, specifically, it can be reduced to constructing an MDS (Minimum Distance Separable), lowest density array code. In this paper, a new method for shortening arbitrary array codes is introduced. It is then used to derive the K_{p+1} family of perfect 1-factorization from the K_{2p} family. Namely, techniques from coding theory are used to prove a new result in graph theory—that the two factorization families are related.**

***Index Terms*—Array codes, error-correcting codes, graph theory, 1-factorization, perfect 1-factorization.**

I. Introduction

ARRAY CODES are erasure-correcting codes, represented by an array of bits. Erasures correspond to the loss of columns. A two-erasure correcting array code, for example, is capable of recovering any two lost columns. For a survey on array codes see [4]. For recent results in array codes see [1]–[3], [6].

Example 1 (Simple Array Code): A simple two-erasure correcting array code of length four is shown below:

a	b	c	d
$b+c$	$c+d$	$d+a$	$a+b$

The first row consists of four information bits a, b, c and d. The second row contains four parity bits. The "+" sign indicates bitwise exclusive-OR, so that $x+x=0$. One can verify that any two columns can recover all four information bits. Suppose, for example, that columns three and four are lost:

a	b		
$b+c$	$c+d$		

c can be recovered by adding $b+c$ to b

$$c = (b+c) + b$$

d can be recovered by adding $c+d$ to c:

$$d = (c+d) + c.$$

Similar decoding chains are used for other erasure patterns.

An example of a family of array codes that are Maximum Distance Separable (MDS), and have optimal update complexity, is the B-Code [8]. As in the case of Example 1, the code is represented by an array, each column corresponding to a symbol in the codeword. Thus, every symbol contains both information and parity, and the code is therefore nonsystematic.

In general, the B-Code is a two-erasure correcting array code of length $2n$, represented by an n by $2n$ array. It can recover the erasure of any two out of the $2n$ columns. The constructions of the B-Codes, the even case B_{2n} and the odd case B_{2n+1}, are based on the perfect 1-factorization of the complete graph, K_{2n+2}. In fact, the K_{p+1} infinite family of B-Codes was described using binary generator matrices in [10]. In [8], the connection between B-code and perfect 1-factorization of K_n was established and both K_{p+1} and K_{2p} families were derived using this connection.

Definition 1 (Perfect 1-Factorization): A perfect 1-factorization of a graph is a partitioning of the set of its edges into subsets, called factors, such that each factor is a graph of degree one, and the union of any two factors forms a Hamiltonian cycle.

Example 2 (A Perfect 1-Factorization of K_{10}): A perfect 1-factorization of K_{10}, is shown in Figs. 1–3. It consists of nine degree-one subgraphs of K_{10}, called factors. The union of any two factors is a Hamiltonian path (complete cycle). The factors are presented in three groups. The first group consists of only

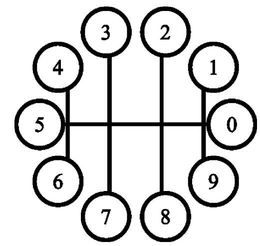


Fig. 1. f_5, the principal factor of the perfect 1-factorization of K_{10}. It is the factor that contains edge $(0,5)$ or in the general case of the perfect 1-factorization of K_{2p} it is the factor, f_p, that contains edge $(0,p)$.

Manuscript received August 20, 2007; revised October 25, 2008. Current version published February 04, 2009. The material in this paper was presented in part at the IEEE International Symposium on Information Theory Seattle, WA, July 2006.

The authors are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: vincent@paradise.caltech.edu; bruck@paradise.caltech.edu). .

Communicated by I. Dumer, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2008.2009850

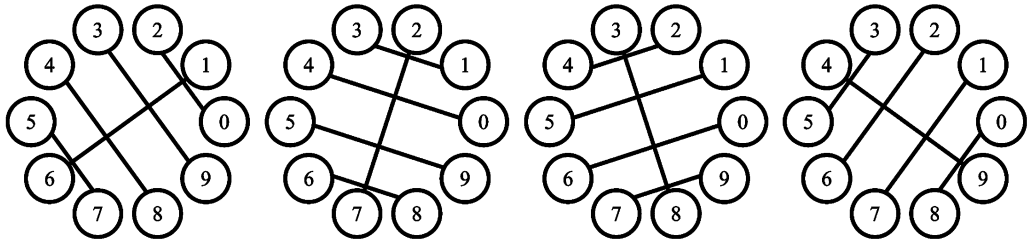


Fig. 2. Even factors of the perfect 1-factorization of K_{10}. The factors labeled f_i, where i is even. Factor f_i is defined as the factor that contains edge $(0, i)$. Notice that for even factors nondiagonal edges have even length—the length being the difference between the edge's endpoints.

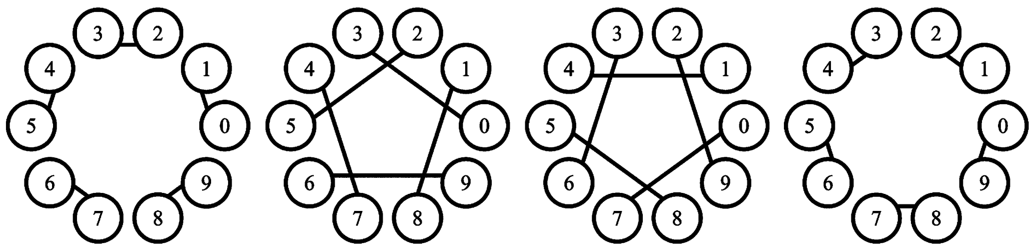


Fig. 3. Odd factors of the perfect 1-factorization of K_{10}. The factors labeled f_i, where i is odd. Factor f_i is defined as the factor that contains edge $(0, i)$. Notice that for odd factors all edges have odd length.

one factor: the principal factor, shown in Fig. 1, which contains the edge $(0, 5)$ (or in the general case the edge $(0, p)$). The second group consists of so called "even" factors (Fig. 2), which contain edges $(0, i)$ where i is even. The last group are the "odd" factors, (Fig. 3), which contain edges $(0, i)$, where i is odd.

A procedure introduced in [8] is used to derive the B-Code of length n, B_n, from a perfect 1-factorization of K_{n+2}. An example of this procedure is shown in Example 3, below.

In the case of the complete graph of even size, K_{2n}, it is still unknown whether or not a perfect 1-factorization exists for all values of n [5], [7]. The following was conjectured in 1963 by Kotzig [5].

Conjecture 1 (Perfect 1-Factorization): A perfect 1-factorizations of the complete graph K_{2n} exists for all values of n, $n > 1$.

So far, two infinite families of perfect 1-factorizations have been shown to exist, namely, the factorizations of K_{p+1} and K_{2p}, where p is an arbitrary prime number ($p > 2$).

The contributions of this paper are twofold.

- A method for **shortening** the B-Code is introduced. It could be used in general to shorten an arbitrary array code.
- The above method along with additional manipulation (**separation**) is used to derive the perfect 1-factorization of the complete graph K_{p+1} from the perfect 1-factorization of K_{2p}. The derivation consists of the following steps.
  1) $\mathcal{P}_{2p}$: perfect 1-factorization of K_{2p}, obtained by known construction ([5], [7]). Shown in Section II-A.
  2) $\mathcal{P}_{2p} \Longrightarrow B_{2p-1}$: extended B-Code of length $2p-1$, by known construction from [8]. Section II-B.
  3) $B_{2p-1} \Longrightarrow \tilde{X}_p$: generalized X-Code of length p, by **shortening** : new construction. Section IV-A .
  4) $\tilde{X}_p \Longrightarrow B_p$: extended B-Code of length p, by **separation**: new construction. Section IV-B.
  5) $B_p \Longrightarrow \mathcal{P}_{p+1}$, by [8, Theorem 5].

The above steps are illustrated, in Section III, by examples for $p = 5$. Proofs are provided for arbitrary p, for Steps 3 and 4 (Section IV-A and Section IV-B).

II. Constructions

In this section, we summarize two known constructions needed for Section III and Section IV, namely the perfect 1-factorization of K_{2p}, from [7] and the B-Code construction from [8].

A. $\mathcal{P}_{2p}$: Perfect 1-Factorization of k_{2p}

For any prime number p the complete graph of $2p$ vertexes, K_{2p}, has a perfect 1-factorization, $\mathcal{P}_{2p}$.

Construction 1 (General Case: $\mathcal{P}_{2p}$): The construction of $\mathcal{P}_{2p}$ is as follows: there are a total of $2p-1$ factors, denoted by f_i. They are organized in three groups.

1) The principal factor f_p, defined as the factor that contains edge $(0, p)$ (shown in Fig. 1 for $p = 5$)
2) $p-1$ even numbered factors, (labeled f_i with i even, shown in Fig. 2 for $p = 5$)
3) $p-1$ odd numbered factors, (labeled f_i with i odd, shown in Fig. 3 for $p = 5$)

Below is the formal definition of the construction. Note that a factor is labeled f_i if it contains edge $\{0, i\}$.

$$\mathcal{P}_{2p} = \{f_i\}, \text{ for } i \in \{1, \ldots, 2p-1\}$$

$$f_i = \begin{cases} \{\{0, p\}, e_{0,1}, e_{0,2}, \ldots, e_{0,p-1}\}, & \text{for } i = p \\ \left\{\left\{\frac{i}{2}, \frac{i}{2}+p\right\}, e_{i,1}, e_{i,2}, \ldots, e_{i,p-1}\right\}, & \text{for } i \text{ even}, i \neq 0 \\ \{e_{i,0}, e_{i,1}, e_{i,2}, \ldots, e_{i,p-1}\}, & \text{for } i \text{ odd}, i \neq p \end{cases}$$

$$e_{i,j} = \begin{cases} \left\{\left(\frac{i}{2}-j\right) \bmod 2p, \left(\frac{i}{2}+j\right) \bmod 2p\right\}, & \text{for } i \text{ even} \\ \{2j, (2j+i) \bmod 2p\}, & \text{for } i \text{ odd} \end{cases}$$

where $\{u, v\}$ is the edge between vertexes u and v. One can verify that factor f_i contains edge $\{0, i\}$, as stated above. Indeed, in the case of $i = p$ it is the first edge in the list. For i even, it is the edge $e_{i,\frac{i}{2}}$. For i odd, $i \neq p$, it is $e_{i,0}$.

The following tables show $\mathcal{P}_{2p}$ as a list of edges per factor. All entries are modulo $2p$. Principal factor f_p (shown in Fig. 1, for $p=5$):

f
$0,p$
$-1,1$
$-2,2$
$\vdots$
$-p+2,p-2$
$-p+1,p-1$

Even factors f_i, i even (shown in Fig. 2, for $p=5$) as shown in the first table at the bottom of the page. Odd factors f_i, i odd, $i \neq p$ (shown in Fig. 3, for $p=5$) as shown in the second table at the bottom of the page. Notice that the odd factors have edges of odd length, while the principal factor and the even factors have even-length edges, with the exception of exactly one edge per factor, the first one, which is of length p.

For the proof that $\mathcal{P}_{2p}$ is perfect see [7].

B. Erasure-Correcting Code Based on $\mathcal{P}_{2p}$

To each perfect 1-factorization of size $2p$, correspond two erasure-correcting array codes: the B-Code, of size $2p-2$, and the extended B-Code of size $2p-1$, B_{2p-2} and B_{2p-1}, respectively [8].

Construction 2 (From P_{2p} to B_{2p-1}): The construction of B_{2p-1} is as follows: there are a total of $2p-1$ columns, of which one is a column of pure information bits corresponding to the edges of the principal factor, f_p, of $\mathcal{P}_{2p}$. The remaining $2p-2$ columns correspond to $p-1$ even factors and $p-1$ odd factors. They each contain one parity bit, corresponding to vertex i, and $p-2$ information bits, corresponding to the other edges in f_i (0 and p are omitted). The parity bit is the sum of all information bits corresponding to edges connected to vertex i. Here is the formal definition of the construction

$$B_{2p-1} = \{b_i\}, \text{ for } i \in \{1, \ldots, 2p-1\}, \text{ where:}$$

$$b_i = \begin{cases} \{\hat{e}_{0,1}, \hat{e}_{0,2}, \ldots, \hat{e}_{0,p-1}\}, & \text{for } i = p \\ \left\{\hat{p}_i, I_{\{\frac{i}{2}, \frac{i}{2}+p\}}, \hat{e}_{i,1}, \hat{e}_{i,2}, \ldots, \hat{e}_{i,p-1}\right\}, & i \text{ even}, i \neq 0 \\ \{\hat{p}_i, \hat{e}_{i,0}, \hat{e}_{i,1}, \hat{e}_{i,2}, \ldots, \hat{e}_{i,p-1}\}, & i \text{ odd}, i \neq p \end{cases}$$

$$\hat{p}_i = \sum_{\{j,k\}, \quad i \in e_{j,k}} \hat{e}_{j,k}$$

$$\hat{e}_{i,j} = \begin{cases} \emptyset, & \text{if } 0 \in e_{i,j} \\ \emptyset, & \text{if } p \in e_{i,j} \\ I_{e_{i,j}}, & \text{otherwise} \end{cases}$$

$$e_{i,j} = \begin{cases} \left\{\left(\frac{i}{2}-j\right) \bmod 2p, \left(\frac{i}{2}+j\right) \bmod 2p\right\}, & \text{for } i \text{ even} \\ \{2j, (2j+i) \bmod 2p\}, & \text{for } i \text{ odd} \end{cases}$$

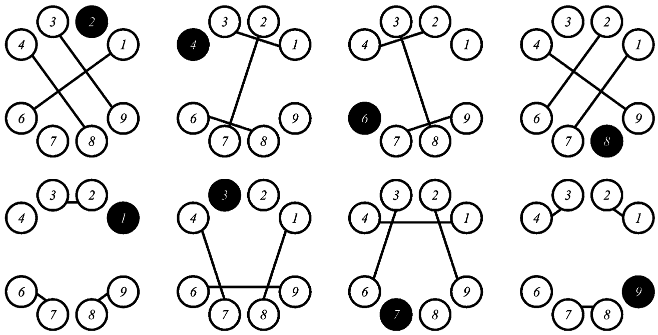
where $e_{i,j}$ are the edges of $\mathcal{P}_{2p}$. $I_{e_{i,j}}$ and $\hat{e}_{i,j}$ are the information bits. They correspond to the edges $e_{i,j}$ minus edges connected to vertexes 0 and p. $\hat{p}_i$ is the parity bit in column i. It is defined as the sum of all information bits $\hat{e}_{j,k}$, such that $i \in e_{j,k}$, that is all information bits represented by an edge containing vertex i. Notice that by this definition every information bit appears in exactly two parity bits.

III. Examples

Example 3 (B_8: B-Code of Size 8): We construct B_8 from $\mathcal{P}10$ by deleting the principal factor f_5, as well as vertexes 0 and 5 and all edges connected to them. Here are the resulting factors in graph format as follows:

f_2	f_4	$\cdots$	f_{2p-2}
$1,1+p$	$2,2+p$	$\cdots$	$p-1,-1$
$0,2$	$1,3$	$\cdots$	$p-2,0$
$-1,3$	$0,4$	$\cdots$	$p-3,1$
$\vdots$	$\vdots$	$\cdots$	$\vdots$
$-p+3,p-1$	$-p+4,p$	$\cdots$	$1,-3$
$-p+2,p$	$-p+3,p+1$	$\cdots$	$0,-2$

f_1	f_3	$\cdots$	f_{p-2}	f_{p+2}	$\cdots$	f_{2p-1}
$0,1$	$0,3$	$\cdots$	$0,p-2$	$0,p+2$	$\cdots$	$0,-1$
$2,3$	$2,5$	$\cdots$	$2,p$	$2,p+4$	$\cdots$	$2,1$
$4,5$	$4,7$	$\cdots$	$4,p+2$	$4,p+6$	$\cdots$	$4,3$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$-4,-3$	$-4,-1$	$\cdots$	$-4,p-6$	$-4,p-2$	$\cdots$	$-4,-5$
$-2,-1$	$-2,1$	$\cdots$	$-2,p-4$	$-2,p$	$\cdots$	$-2,-3$

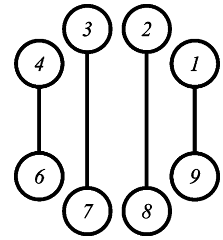


Vertexes correspond to parity bits, while edges are information bits. Black vertexes are the ones that were connected to vertex 0. They indicate the placement of parity bits relative to columns of information bits. Notice that the union of any two factors forms a graph such that starting at the black nodes and following edges one can uniquely traverse all remaining nodes (this is the erasure-recovery path). Here follows the array representation of the above graphs as shown in the first table at the bottom of the page. The $a_{j,k}$ are information bits. The p_i are parity bits. They are related by

$$p_i = \sum_{(j,k)/i\in\{j,k\}} a_{j,k}.$$

In other words, parity bit i is the sum of all information bits that have i as one of their two indexes. Notice that by definition every information bit appears in exactly two parity bits.

Example 4 (Extended B-Code of Size 9, B_9): B_8 can be extended by the addition of a column of information bits. Those are the bits corresponding to the edges of the principal factor f_5 (which was deleted in the construction of B_8):



And see also the second table at the bottom of the next page.

Example 5 (Shortening B_8 Into $\tilde{X}_4$): In the array representing B_8 (from Example 3), we set all information bits in the last four columns to zero. We obtain the following array shown in the third table at the bottom of the page. Notice that the zeroed columns correspond to parities with odd indexes:

$$\begin{aligned} p_1 &= a_{1,6} + a_{1,3} + a_{1,7} \\ p_3 &= a_{9,3} + a_{1,3} + a_{3,8} \\ p_7 &= a_{2,7} + a_{9,7} + a_{1,7} \\ p_9 &= a_{9,3} + a_{9,7} + a_{4,9}. \end{aligned}$$

Notice that each equation has exactly one information bit, $a_{i,j}$, with an even index. Rewriting the above equations, we get

$$\begin{aligned} a_{1,6} &= p_1 + a_{1,3} + a_{1,7} \\ a_{3,8} &= a_{9,3} + a_{1,3} + p_3 \\ a_{2,7} &= p_7 + a_{9,7} + a_{1,7} \\ a_{4,9} &= a_{9,3} + a_{9,7} + p_9. \end{aligned}$$

Renaming $a_{1,6}$, $a_{3,8}$, $a_{2,7}$, and $a_{4,9}$ as parities and p_1, p_3, p_7, and p_9 as information bits, we set the information bits to zero and relabel the table shown at the bottom of the following page.

$a_{1,6}$	$a_{2,7}$	$a_{3,8}$	$a_{4,9}$	$a_{2,3}$	$a_{4,7}$	$a_{2,9}$	$a_{2,1}$
$a_{9,3}$	$a_{1,3}$	$a_{2,4}$	$a_{2,6}$	$a_{6,7}$	$a_{6,9}$	$a_{4,1}$	$a_{4,3}$
$a_{8,4}$	$a_{8,6}$	$a_{9,7}$	$a_{1,7}$	$a_{8,9}$	$a_{8,1}$	$a_{6,3}$	$a_{8,7}$
p_2	p_4	p_6	p_8	p_1	p_3	p_7	p_9

.

$a_{1,9}$	$a_{1,6}$	$a_{2,7}$	$a_{3,8}$	$a_{4,9}$	$a_{2,3}$	$a_{4,7}$	$a_{2,9}$	$a_{2,1}$
$a_{2,8}$	$a_{9,3}$	$a_{1,3}$	$a_{2,4}$	$a_{2,6}$	$a_{6,7}$	$a_{6,9}$	$a_{4,1}$	$a_{4,3}$
$a_{3,7}$	$a_{8,4}$	$a_{8,6}$	$a_{9,7}$	$a_{1,7}$	$a_{8,9}$	$a_{8,1}$	$a_{6,3}$	$a_{8,7}$
$a_{4,6}$	p_2	p_4	p_6	p_8	p_1	p_3	p_7	p_9

.

$a_{1,6}$	$a_{2,7}$	$a_{3,8}$	$a_{4,9}$	0	0	0	0
$a_{9,3}$	$a_{1,3}$	$a_{2,4}$	$a_{2,6}$	0	0	0	0
$a_{8,4}$	$a_{8,6}$	$a_{9,7}$	$a_{1,7}$	0	0	0	0
p_2	p_4	p_6	p_8	p_1	p_3	p_7	p_9

.

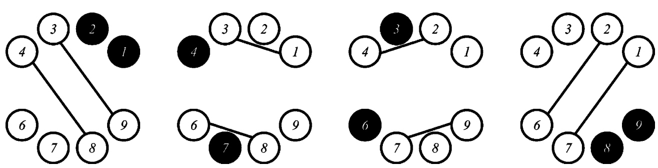
Notice that because of the change of variables, the even-indexed parity bits depend on two extra information bits. This fact will be ignored in the graphical representation, but will be taken into account in the final proof (Section IV-B). As part of the proof those information bits will be set to zero.

Rearranging and removing the zeroed columns, we get the array representing $\tilde{X}_4$:

$a_{9,3}$ $a_{8,4}$	$a_{1,3}$ $a_{8,6}$	$a_{2,4}$ $a_{9,7}$	$a_{2,6}$ $a_{1,7}$
p_1 p_2	p_7 p_4	p_3 p_6	p_9 p_8

.

In the graph domain each factor has two edges and two black vertexes:

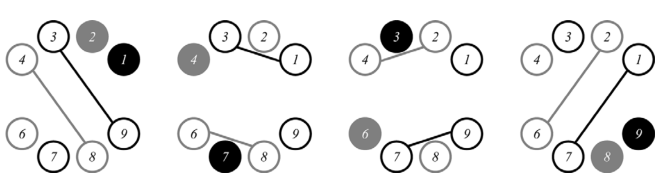


Example 6 (Shortening B_9 Into $\tilde{X}_5$): B_9 has an extra column of information bits. Those bits are of the form $a_{i,j}$ where i and j are either both odd, or both even. Therefore they do not interfere with the choice of information bits that are substituted with parities. The resulting array for $\tilde{X}_9$ is

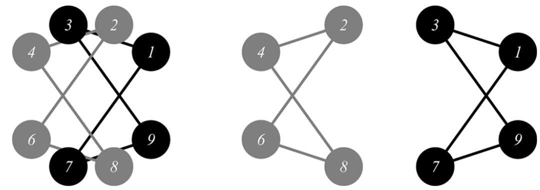
$a_{1,9}$ $a_{2,8}$	$a_{9,3}$ $a_{8,4}$	$a_{1,3}$ $a_{8,6}$	$a_{2,4}$ $a_{9,7}$	$a_{2,6}$ $a_{1,7}$
$a_{3,7}$ $a_{4,6}$	p_1 p_2	p_7 p_4	p_3 p_6	p_9 p_8

.

Example 7 (Separation of $\tilde{X}_4$): Notice that the arrays for $\tilde{X}_4$ and $\tilde{X}_5$ above contain edges in which either both indices are odd or both indices are even. We color gray all even nodes and edges touching them. All odd nodes and edges—black:



Taking the union and identifying the connected components of the graph:



IV. Theorems and Proofs

In this section, we present the contributions of the paper, namely, the following.

- A method for shortening array codes. While it is easy to shorten systematic error-correcting codes, in the case of array codes, which are not systematic, shortening is not obvious. The reason is that every symbol contains both information and parity bits.
- A method for separating an array code. Given an array code with array of height h, we produce two arrays of height $\frac{1}{2}h$ each corresponding to an array code.
- A procedure that uses the above two constructions to derive one family of B-Codes from another one.
- Based on that, and a result from [8], that relates B-Codes to perfect 1-factorizations of the complete graphs, K_n, we present a procedure that derives the perfect 1-factorization of K_{p+1} from the perfect 1-factorization of K_{2p}.

A. Shortening: $B_{2p-1} \longrightarrow \tilde{X}_p$

Let $n = p - 1$. B_{2n} is represented by a $n \times 2n$ array. By setting n^2 information bits to zero, the array can be shortened into a $n \times n$ array corresponding to a new, square-shaped array code of size n. It has the dimension of the X-Code [9] of same length. We call this new code "generalized" X-Code of size n and denote it by $\tilde{X}_n$.

Construction 3 (Shortening B_{2p-2} and B_{2p-1}): Referring to the construction of B_{2p-1} namely Construction 2.

1) In the columns corresponding to odd factors set all information bits to zero:
$$I_{e_{i,j}} = 0, \text{ for } i \text{ odd}, i \neq p.$$
2) For the parity bit in each zeroed column identify an information bit in a non-zeroed column and exchange them by a change of variables:
$$\hat{p}_i = \hat{e}_{j,k} + S_i \Longrightarrow \hat{e}_{j,k} = \hat{p}_i + S_i$$
where $\hat{e}_{j,k}$ is interpreted as a new parity bit, and $\hat{p}_i$ a new information bit. S_i is the sum of the remaining information bits in the original $\hat{p}_i$.
3) Set the new information bits to zero:
$$\hat{p}_i = 0.$$

p_1 $a_{9,3}$ $a_{8,4}$	p_7 $a_{1,3}$ $a_{8,6}$	p_3 $a_{2,4}$ $a_{9,7}$	p_9 $a_{2,6}$ $a_{1,7}$	0 0 0	0 0 0	0 0 0	0 0 0
p_2	p_4	p_6	p_8	0	0	0	0

.

Theorem 1 ($\tilde{X}_{p-1}$ and $\tilde{X}_p$ are MDS): The shortened B-Codes, $\tilde{X}_{p-1}$ and $\tilde{X}_p$, obtained by the construction described above are MDS.

Proof: The proof consists of two parts. We first show that a single change of variables between a parity bit and an information bit preserves the MDS properties of the array. We then show that to the parity bit, $\hat{p}_i$, in every zeroed column uniquely corresponds one non-zeroed column, and an information bit, $\hat{e}_{j,k}$, in it, such that

$$\hat{p}_i = \hat{e}_{j,k} + S_i.$$

Part 1: A single change of variables described in the construction above corresponds to adding a row to another row of the parity check matrix of the B-Code, thus preserving the MDS property of the code.

Part 2: Consider the information bit indexed by

$$e = \left\{\frac{i}{2}, \frac{i}{2} + p\right\}$$

(see Construction 2). One such bit appears in each even numbered column (i even) of the array. p being and odd prime implies that either $\frac{i}{2}$ is odd or $\frac{i}{2} + p$ is odd, but not both. Therefore I_e the bit indexed by edge e, appears in exactly one parity bit p_k, k odd, i.e., in exactly one of the zeroed columns.

B. Separation: From $\tilde{X}_p$ to B_p

Because of the particular shortening used in Section IV-A, we show that each column, C_i, of $\tilde{X}_p$ divides into two sets of bits A_i and B_i, such that the parity bit of set A_i only depends on information bits in sets A_j (and not on information bits of any of the B_j sets). We can therefore extract a new array based on the A_i, which turns out to be the array representing the B-Code, B_p. Here follows the formal theorem and proof.

Theorem 2 (Separation of $\tilde{X}_p$): $\tilde{X}_p$ can be separated into two arrays, of which one corresponds to B_p.

Proof: By examining Construction 2, notice that after shortening the extended B-Code, B_{2p-1}, we are left only with edges of even length. In other words, all information bits in $\tilde{X}_p$ are indexed by pairs of the form:

$$e = \left\{\left(\frac{i}{2} - j\right) \bmod 2p, \left(\frac{i}{2} + j\right) \bmod 2p\right\}.$$

Therefore, half of the information bits of every column in $\tilde{X}_p$ are of the form $a_{i,j}$ where both i and j are odd. For the other half both i and j are even. By definition, the even-indexed parities in B_{2p-1} depend only on information bits with at least one even index. That is true also for the new parities of $\tilde{X}_p$, defined by the change of variable during the shortening process. Therefore, all even-indexed bits form an independent $p \times \frac{p-1}{2}$ array such that every information bit appears in exactly two parity bits. As mentioned in Example 5, the even parities depend on some odd-indexed information bits. Those are set to zero. As a result the odd-indexed subarray is set to zero, and the even-indexed subarray is used to define the new code. The resulting code is MDS since $\tilde{X}_p$ is MDS. A counting argument shows that such a code can only be B_p. Indeed the number of parity bits (nodes) is $p-1$ and the number of information bits (edges) is

$$p\frac{p-1}{2} - (p-1) = \frac{(p-1)(p-2)}{2}.$$

Those are all the edges over $p-1$ vertexes.

V. Conclusion

The perfect 1-factorization of K_{2n} is a 40-year-old open problem in graph theory. Two infinite families of solutions are known, indexed by prime number p, namely, for K_{2p} and for K_{p+1}. In [8], the extended B-Code, an erasure correcting code introduced in [10], was shown to be equivalent to a perfect 1-factorization of the complete graph.

In this paper, we presented a general method for shortening an array code and applied it to the B-Code. The resulting procedure allows one to derive the perfect 1-factorization of the complete graph K_{p+1} from the perfect 1-factorization of K_{2p}. The procedure consists of the following steps.

1) $\mathcal{P}_{2p}$: perfect 1-factorization of K_{2p}, obtained by known construction ([5], [7]). Shown in Section II-A.
2) $\mathcal{P}_{2p} \Longrightarrow B_{2p-1}$: extended B-Code of length $2p-1$, by known construction from [8]. Section II-B.
3) $B_{2p-1} \Longrightarrow \tilde{X}_p$: generalized X-Code of length p, by **shortening**: new construction. Section IV-A.
4) $\tilde{X}_p \Longrightarrow B_p$: extended B-Code of length p, by **separation**: new construction. Section IV-B.
5) $B_p \Longrightarrow \mathcal{P}_{p+1}$, by [8, Theorem 5].

References

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

[2] 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, 1996.

[3] M. Blaum and R. M. Roth, "On lowest density MDS codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 46–59, 1999.

[4] P. G. Farrell, "A survey of array error control codes," *ETT*, vol. 3, no. 5, pp. 441–454, 1992.

[5] A. Kotzig, "Hamilton graphs and Hamilton circuits," in *Theory of Graphs and Its Applications, Proc. Sympos.*, Smolenice, Hungary, 1963, pp. 63–82.

[6] E. Luidor and R. M. Roth, Lowest-Density MDS Codes Over Extension Alphabets Technion CS Technical report [Online]. Available: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2005/CS/CS-2005-09.pdf

[7] W. D. Wallis, *One-Factorizations*. New York: Kluwer Academic, 1997.

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

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

[10] G. V. Zaitsev, V. A. Zinov'ev, and N. V. Semakov, "Minimum-check-density codes for correcting bytes of errors, erasures or defects," *Problems Inf. Transm.*, no. 19, pp. 197–204, 1981.

Vasken Bohossian received the B.Sc. degree in electrical engineering from McGill University, Montreal, in 1993, the M.Sc. degree in electrical engineering, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, in 1994 and 1998, respectively.

His professional experience includes Bell Canada, the "Centre National de la Recherche Scientifique", and Wilshire Associates. He was a cofounder of Rainfinity, a spin-off company from Caltech that focused on software products for management of network information storage systems. His interests include error-correcting codes, graph theory, and algorithms for efficient, fault-tolerant, distributed storage and communictions, and their application to computer networks.

Jehoshua (Shuki) Bruck (S'86–M'89–SM'93–F'01) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion–Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1989.

He is the Gordon and Betty Moore Professor of Computation and Neural Systems and Electrical Engineering at the California Institute of Technology. His research focuses on information theory and systems and the theory of biological networks. He has an extensive industrial experience, including working with IBM Research where he participated in the design and implementation of the first IBM parallel computer. He was a cofounder and chairman of Rainfinity, a spin-off company from Caltech that focused on software products for management of network information storage systems. He published more than 200 journal and conference papers in his areas of interests and he holds 28 U.S. patents.

Dr. Bruck's awards include the National Science Foundation Young Investigator award and the Sloan fellowship. His publications were recognized by awards, including, a selection as an ISI highly cited researcher, winning the 2005 S. A. Schelkunoff Transactions prize paper award from the IEEE Antennas and Propagation Society and the Best Paper Award in the 2003 Design Automation Conference.