

# New Constructions of Distance-Increasing Mappings and Permutation Arrays\*

JEN-CHUN CHANG

*Department of Computer Science and Information Engineering  
National Taipei University  
Taipei, 237 Taiwan*

Permutation arrays (PAs, or permutation codes) are useful in communication over power lines. Distance-increasing mappings (DIMs) from the set of binary vectors of a fixed length to the set of permutations of the same length that strictly increase Hamming distances (except when that is obviously not possible) are useful to construct permutation arrays. In this paper, we first give a new recursive construction of DIMs. Next, with a tricky and very different way to use the recursive construction of DIMs, we find a new construction of PAs. Finally, a new lower bound for the size of PAs is given.

**Keywords:** distance-preserving mappings (DPMs), distance-increasing mappings (DIMs), code constructions, permutation arrays (PAs), Hamming distance

## 1. INTRODUCTION

Distance-preserving mappings of length  $n$ , shortly  $n$ -DPMs, are mappings from the set of all binary vectors of length  $n$  to the set of all permutations of  $Z_n = \{1, 2, 3, \dots, n\}$  that preserve or increase Hamming distances. These mappings are useful for the construction of permutation arrays (PAs). Recently, Lee [1] and Chang and others [2] proposed several nice constructions of DPMs.

An  $(n, k)$  distance-increasing mapping, shortly an  $(n, k)$ -DIM, is a mapping from the set of all binary vectors of length  $n$  to the set of all permutations of  $Z_n = \{1, 2, 3, \dots, n\}$  that strictly increases Hamming distances by at least  $k$  (except when that is obviously not possible). These  $(n, k)$ -DIMs are more useful for the construction of permutation arrays than  $n$ -DPMs. In fact, some  $n$ -DPMs given in [2] are  $(n, 1)$ -DIMs. In 2005, Chang [3] proposed a systematic construction of  $(n, 1)$ -DIMs for any  $n \geq 4$ . Recently, Chang [4] found that for any  $k \geq 1$ , there exists a minimal positive integer  $n_k$  such that an  $(n, k)$ -DIM can be constructed for any  $n \geq n_k$ . It is known that  $n_1 = 4$ ,  $n_2 \leq 48$ ,  $n_k \leq 22n_{k-1} + 4$  for  $k \geq 3$ . The exact values of  $n_k$  for  $k \geq 2$  are still unknown.

Permutation arrays (PAs) were studied early in 1970s. A recent application by Vinck [5] of PAs to a coding/modulation scheme for communication over power lines has created renewed interest in PAs. For reference, see [6-10]. It was shown in [2] and [3] that DPMs and DIMs are useful for the constructions of PAs.

In this paper, we are working on the following three tasks.

---

Received May 16, 2005; revised August 16 & October 24, 2005; accepted November 14, 2005.

Communicated by Chi-Jen Lu.

\* This work was supported in part by the National Science Council of Taiwan, R.O.C., under contracts No. NSC 94-2213-E-305-003 and NSC 95-2213-E-305-002.

- (1) To give a new recursive construction of  $(n, k)$ -DIMs.
- (2) With a tricky and very different way to use the new recursive construction of  $(n, k)$ -DIMs, to propose a new construction of permutation arrays of length  $mn$  for  $m \geq 1$ .
- (3) To give a new lower bound for the size of permutation arrays.

## 2. BASIC NOTATIONS

Let  $N$  be the set of all positive integers, and  $S_n$  be the set of all  $n!$  permutations of  $Z_n = \{1, 2, 3, \dots, n\}$ . A permutation of  $Z_n$  is an one-to-one and onto function  $\pi: Z_n \rightarrow Z_n$  represented by an  $n$ -tuple  $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$  where  $\pi_i = \pi(i)$ . Let  $Z_2^n$  denote the set of all binary vectors of length  $n$ . A binary vector  $x \in Z_2^n$  is denoted by an  $n$ -tuple  $x = (x_1, x_2, x_3, \dots, x_n)$  where  $x_i$  is the  $i$ th bit of  $x$ .

The Hamming distance between two  $n$ -tuples  $a = (a_1, a_2, \dots, a_n)$  and  $b = (b_1, b_2, \dots, b_n)$ , denoted by  $d(a, b)$ , is defined to be the number of positions where they differ, that is

$$d(a, b) = |\{j \in Z_n \mid a_j \neq b_j\}|.$$

Let  $\min$  be the minimal function. An  $(n, k)$  distance-increasing mapping, an  $(n, k)$ -DIM for short, is a mapping  $f: Z_2^n \rightarrow S_n$  such that for any pair of distinct binary vectors  $x, y \in Z_2^n$  and  $x \neq y$ ,

$$d(f(x), f(y)) \geq \min(d(x, y) + k, n).$$

Let  $F_{n,k}$  denote the set of all  $(n, k)$ -DIMs. We already have  $|F_{n,k}| > 0$  for  $k = 1$  and  $n \geq 4$  in [3].

## 3. EXISTENCE OF $(n, k)$ -DIMS

For  $k = 1$ , Chang [3] has shown that an  $(n, 1)$ -DIM can be constructed by a systematic method for any  $n \geq 4$ .

For  $k \geq 2$ , Chang [4] also proved that for any  $k \geq 1$ , there always exists a minimal positive integer  $n_k$  such that an  $(n, k)$ -DIM can be constructed for any  $n \geq n_k$ . It is known that  $n_1 = 4$ ,  $n_2 \leq 48$ ,  $n_k \leq 22n_{k-1} + 4$  for  $k \geq 3$ . The exact values of  $n_k$  for  $k \geq 2$  are still unknown.

Constructions of  $(n, 2)$ -DIMs for  $n \in \{16, 20, 24, 25, 28, 30, 32, 35, 36, 40, 41, 42, 44, 45, 46\}$  or any  $n \geq 48$  were found in Chang's paper [4]. As an example to illustrate the existence of  $(n, k)$ -DIMs for  $k \geq 2$ , we quote the construction of a  $(16, 2)$ -DIM,  $Q_{16}$ , from [4] below.

**Algorithm  $Q_{16}$** , a  $(16, 2)$ -DIM

**Input:**  $(x_1, x_2, \dots, x_{16}) \in Z_2^{16}$

**Output:**  $(\pi_1, \pi_2, \dots, \pi_{16}) \in S_{16}$

**Begin**

$(\pi_1, \pi_2, \dots, \pi_{16}) \leftarrow (1, 2, \dots, 16)$   
for  $i = 0$  to 3 do

```

{
  for j = 0 to 1 do
    if (x4i+j+1 = 1) then swap(π4i+2j+1, π4i+2j+2)
  for j = 2 to 3 do
    if (x4i+j+1 = 1) then swap(π4i+j-1, π4i+j+1)
  }
for j = 0 to 3 do
{
  for i = 0 to 1 do
    if (x4i+j+1 = 1) then swap(π8i+j+1, π8i+j+5)
  for i = 2 to 3 do
    if (x4i+j+1 = 1) then swap(π4i+j-7, π4i+j+1)
  }
}
End
    
```

The distance expansion matrix of  $Q_{16}$  is given in Appendix where the element at the cross of row  $i$  and column  $j$  is the number of unordered pairs  $\{x, y\}$  of binary vectors of length 16 such that  $d(x, y) = i$  and  $d(Q_{16}(x), Q_{16}(y)) = j$ . In order to fit the width of the paper, we split the distance expansion matrix into four parts from left to right. To make the observation easy, cells on the diagonal line of the matrix are shadowed. With the distance expansion matrix, it is easy to check that  $Q_{16}$  is a (16, 2)-DIM.

#### 4. A NEW RECURSIVE CONSTRUCTION OF DIMS

In this section, we propose a new recursive construction of  $(n, k)$ -DIMS. It constructs an  $(m + n, k)$ -DIM from an  $(m, k)$ -DIM and an  $(n, k)$ -DIM.

**Construction 1** Let  $m, n, k \in \mathbb{N}$ ,  $\min(m, n) > 2k > 0$ ,  $f$  be an  $(m, k)$ -DIM, and  $g$  be an  $(n, k)$ -DIM. Define  $f \bullet g : Z_2^{m+n} \rightarrow S_{m+n}$  as follows. For any  $x = (x_1, x_2, x_3, \dots, x_{m+n}) \in Z_2^{m+n}$ , if

$$\begin{aligned}
 f(x_1, x_2, x_3, \dots, x_m) &= (u_1, u_2, u_3, \dots, u_m), \text{ and} \\
 g(x_{m+1}, x_{m+2}, x_{m+3}, \dots, x_{m+n}) &= (v_1, v_2, v_3, \dots, v_n),
 \end{aligned}$$

then

$$f \bullet g(x) = (\pi_1, \pi_2, \pi_3, \dots, \pi_{m+n})$$

where

$$\pi_i = \begin{cases} (1 - x_i)u_i + x_i(v_{n+1-i} + m) & 1 \leq i \leq k \\ u_i & k < i \leq m - k \\ (1 - x_{2m+1-i})u_i + x_{2m+1-i}(v_{m+1-i} + m) & m - k < i \leq m \\ x_i u_{2m+1-i} + (1 - x_i)(v_{i-m} + m) & m < i \leq m + k \\ v_{i-m} + m & m + k < i \leq m + n - k \\ x_{m+n+1-i}u_{m+n+1-i} + (1 - x_{m+n+1-i})(v_{i-m} + m) & m + n - k < i \leq m + n. \end{cases} \quad \square$$

To help the understanding of Construction 1, an alternative algorithmic description is given below.

**Algorithm of Construction 1**

**Input:**  $f \in F_{m,k}$ ,  $g \in F_{n,k}$ , and  $(x_1, x_2, x_3, \dots, x_{m+n}) \in Z_2^{m+n}$

**Output:**  $(\pi_1, \pi_2, \pi_3, \dots, \pi_{m+n}) \in S_{m+n}$

**Begin**

$(u_1, u_2, u_3, \dots, u_m) \leftarrow f(x_1, x_2, x_3, \dots, x_m)$

$(v_1, v_2, v_3, \dots, v_n) \leftarrow g(x_{m+1}, x_{m+2}, x_{m+3}, \dots, x_{m+n})$

$(\pi_1, \pi_2, \pi_3, \dots, \pi_{m+n}) \leftarrow (u_1, u_2, u_3, \dots, u_m, v_1 + m, v_2 + m, v_3 + m, \dots, v_n + m)$

for  $i = 1$  to  $k$  do

{

if  $(x_i = 1)$  then swap( $\pi_i, \pi_{m+n+1-i}$ )

if  $(x_{m+i} = 1)$  then swap( $\pi_{m+i}, \pi_{m+1-i}$ )

}

**End**

In order to increase the readability, we give an example to illustrate the construction.

**Example 1:** Let  $f$  be the  $(4, 1)$ -DIM given in [3] and  $g$  be the  $(5, 1)$ -DIM given in [3]. By Construction 1,  $f \bullet g : Z_2^9 \rightarrow S_9$  is well-defined. Consider  $x = (1, 1, 0, 1, 0, 0, 0, 1, 1)$ . Since  $f(1, 1, 0, 1) = (2, 3, 4, 1)$  and  $g(0, 0, 0, 1, 1) = (2, 5, 3, 1, 4)$ ,  $f \bullet g(x) = (8, 3, 4, 1, 6, 9, 7, 5, 2)$ .  $\square$

In fact, the function  $f \bullet g$  obtained from Construction 1 is an  $(m + n, k)$ -DIM. We prove it in the following theorem.

**Theorem 1** For  $m, n, k \in N$ , and  $\min(m, n) > 2k > 0$ , if  $f$  is an  $(m, k)$ -DIM and  $g$  is an  $(n, k)$ -DIM, then  $f \bullet g$  is an  $(m + n, k)$ -DIM.

**Proof:** Let  $\delta$  be a comparison function defined as follows,

$$\delta(a, b) = \begin{cases} 1 & a \neq b \\ 0 & a = b. \end{cases}$$

Consider any two distinct binary vectors  $x, y \in Z_2^{m+n}$  and  $x \neq y$ . Let

$$f(x_1, x_2, x_3, \dots, x_m) = (u_1, u_2, u_3, \dots, u_m)$$

$$g(x_{m+1}, x_{m+2}, x_{m+3}, \dots, x_{m+n}) = (v_1, v_2, v_3, \dots, v_n)$$

$$f(y_1, y_2, y_3, \dots, y_m) = (w_1, w_2, w_3, \dots, w_m)$$

$$g(y_{m+1}, y_{m+2}, y_{m+3}, \dots, y_{m+n}) = (t_1, t_2, t_3, \dots, t_n)$$

$$f \bullet g(x) = (p_1, p_2, p_3, \dots, p_{m+n})$$

$$f \bullet g(y) = (q_1, q_2, q_3, \dots, q_{m+n}).$$

Furthermore, we define

$$d_1 = d((x_1, x_2, \dots, x_m), (y_1, y_2, \dots, y_m))$$

$$d_2 = d((x_{m+1}, x_{m+2}, \dots, x_{m+n}), (y_{m+1}, y_{m+2}, \dots, y_{m+n})).$$

Thus  $d(x, y) = d_1 + d_2 > 0$ . We first consider

$$d((p_1, p_{m+n}), (q_1, q_{m+n})) = \delta(1 - x_1)u_1 + x_1(v_n + m), (1 - y_1)w_1 + y_1(t_n + m) + \delta x_1 u_1 + (1 - x_1)(v_n + m), y_1 w_1 + (1 - y_1)(t_n + m).$$

When  $x_1 = y_1$ , it equals  $d((u_1, v_n), (w_1, t_n))$ ; when  $x_1 \neq y_1$ , it is always 2. Thus we can rewrite the equation as follows.

$$d((p_1, p_{m+n}), (q_1, q_{m+n})) = d((u_1, v_n), (w_1, t_n)) + \delta(x_1, y_1)(2 - d((u_1, v_n), (w_1, t_n))).$$

Similar equations for  $d((p_2, p_{m+n-1}), (q_2, q_{m+n-1}))$ ,  $d((p_3, p_{m+n-2}), (q_3, q_{m+n-2}))$ , ...,  $d((p_k, p_{m+n+1-k}), (q_k, q_{m+n+1-k}))$ , and  $d((p_{m+1}, p_m), (q_{m+1}, q_m))$ ,  $d((p_{m+2}, p_{m-1}), (q_{m+2}, q_{m-1}))$ , ...,  $d((p_{m+k}, p_{m+1-k}), (q_{m+k}, q_{m+1-k}))$  can also be derived. Therefore,

$$d(f \bullet g(x), f \bullet g(y)) = d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) + d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) + \sum_{i=1}^k \delta(x_i, y_i)(2 - d((u_i, v_{n+1-i}), (w_i, t_{n+1-i}))) + \sum_{i=m+1}^{m+k} \delta(x_i, y_i)(2 - d((u_{2m+1-i}, v_{i-m}), (w_{2m+1-i}, t_{i-m}))). \tag{1}$$

We divide the remaining part of proof into several cases according to the following table.

	$d_2 = 0$	$1 \leq d_2 \leq n - k$	$d_2 > n - k$
$d_1 = 0$	impossible	Case 1	Case 2
$1 \leq d_1 \leq m - k$	Case 6	Case 3	Case 4
$d_1 > m - k$	Case 7	Case 8	Case 5

**Case 1:**  $d_1 = 0, 1 \leq d_2 \leq n - k$ . In this case,

$$d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) = 0$$

$$d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) \geq d_2 + k.$$

By Eq. (1),

$$d(f \bullet g(x), f \bullet g(y)) \geq d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) + d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) \geq d_2 + k = d(x, y) + k \geq \min(d(x, y) + k, m + n).$$

**Case 2:**  $d_1 = 0, n \geq d_2 > n - k$ . In this case,

$$d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) = 0$$

$$d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) = n.$$

By Eq. (1),

$$\begin{aligned} d(f \bullet g(x), f \bullet g(y)) &= d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) + d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) \\ &\quad + \sum_{i=1}^k \delta(x_i, y_i) + \sum_{i=m+1}^{m+k} \delta(x_i, y_i) = 0 + n + 0 + \sum_{i=m+1}^{m+k} \delta(x_i, y_i) \\ &\geq 0 + n + 0 + (d_2 - (n - k)) = d_2 + k = d(x, y) + k \\ &\geq \min(d(x, y) + k, m + n). \end{aligned}$$

**Case 3:**  $1 \leq d_1 \leq m - k$ ,  $1 \leq d_2 \leq n - k$ . In this case,

$$\begin{aligned} d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) &\geq d_1 + k \\ d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) &\geq d_2 + k. \end{aligned}$$

By Eq. (1),

$$\begin{aligned} d(f \bullet g(x), f \bullet g(y)) &\geq d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) + d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) \\ &\geq d_1 + k + d_2 + k > d_1 + d_2 + k = d(x, y) + k \geq \min(d(x, y) + k, m + n). \end{aligned}$$

**Case 4:**  $1 \leq d_1 \leq m - k$ ,  $n \geq d_2 > n - k$ . In this case,

$$\begin{aligned} d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) &\geq d_1 + k \\ d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) &= n. \end{aligned}$$

By Eq. (1),

$$\begin{aligned} d(f \bullet g(x), f \bullet g(y)) &\geq d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) + d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) \\ &\geq d_1 + k + n \geq d_1 + d_2 + k = d(x, y) + k \geq \min(d(x, y) + k, m + n). \end{aligned}$$

**Case 5:**  $m \geq d_1 > m - k$ ,  $n \geq d_2 > n - k$ . In this case,

$$\begin{aligned} d((u_1, u_2, \dots, u_m), (w_1, w_2, \dots, w_m)) &= m \\ d((v_1, v_2, \dots, v_n), (t_1, t_2, \dots, t_n)) &= n. \end{aligned}$$

By Eq. (1),

$$d(f \bullet g(x), f \bullet g(y)) = m + n \geq \min(d(x, y) + k, m + n).$$

**Case 6:**  $1 \leq d_1 \leq m - k$ ,  $d_2 = 0$ . The proof is similar to Case 1.

**Case 7:**  $m \geq d_1 > m - k$ ,  $d_2 = 0$ . The proof is similar to Case 2.

**Case 8:**  $m \geq d_1 > m - k$ ,  $1 \leq d_2 \leq n - k$ . The proof is similar to Case 4.  $\square$

Based on Construction 1 and Theorem 1, we can derive a recursive lower bound for  $|F_{m+n,k}|$  as follows.

**Theorem 2** For  $m, n, k \in \mathbb{N}$ , and  $\min(m, n) > 2k > 0$ ,  $|F_{m+n,k}| \geq |F_{m,k}| |F_{n,k}|$ .

**Proof:** Since for each  $f \in F_{m,k}$  and each  $g \in F_{n,k}$ , the mapping  $f \bullet g$  has been proved to be in  $F_{m+n,k}$ , it is only necessary to prove that if  $f_1 \bullet g_1 = f_2 \bullet g_2$ , then  $f_1 = f_2$  and  $g_1 = g_2$ . But we prove another equivalent sentence: if  $f_1 \neq f_2$  or  $g_1 \neq g_2$ , then  $f_1 \bullet g_1 \neq f_2 \bullet g_2$ .

- (1) Let  $f_1 \neq f_2$ . There must exist a binary vector  $x$  of length  $m$  such that  $f_1(x) \neq f_2(x)$ . Extending  $x$  to  $x'$  by appending  $n$  zeroes at its tail, we get a binary vector  $x'$  of length  $m + n$ . Since  $f_1 \bullet g_1(x') \neq f_2 \bullet g_2(x')$ , so  $f_1 \bullet g_1 \neq f_2 \bullet g_2$ .
- (2) If  $g_1 \neq g_2$ , the proof is similar to the above case 1. □

### 5. NEW CONSTRUCTIONS OF PERMUTATION ARRAYS

An  $(n, d)$  PA is a subset of  $S_n$  where the Hamming distance between any two distinct permutations is at least  $d$ . An  $(n, d)$  binary code is a subset of  $Z_2^n$  where the Hamming distance between any two distinct binary vectors is at least  $d$ .

In order to construct PAs with larger sizes, the way we use  $(n, k)$ -DIMs are tricky and very different. Normally, applying an  $(n, k)$ -DIM on an  $(n, d)$  binary code, we can only obtain an  $(n, \min(d + k, n))$  PA whose size is equal to that of the  $(n, d)$  binary code. But now in our new way, applying an  $(n, k)$ -DIM on an  $(mn, d_1)$  binary code, with the effects of permutations in an  $(m, d_2)$  PA, we will get an  $(mn, \min(d_1 + k, d_2(n - 2k), mn))$  PA whose size is the product of the sizes of the  $(mn, d_1)$  binary code and the  $(m, d_2)$  PA.

Before introducing our new construction, we need some definitions. The following definition defines  $\hat{\pi}^n$  to be an extension of  $\pi$ . When represented by tuples,  $\hat{\pi}^n$  is  $n$  times longer than  $\pi$ .

**Definition 1** Let  $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_m)$  be a permutation of  $Z_m$  where  $\pi_i = \pi(i)$ . For any  $n > 0$ , the permutation  $\hat{\pi}^n$  of  $Z_{mn}$  is defined as

$$\hat{\pi}^n(i) = \hat{\pi}_i^n = (\pi(\lceil \frac{i}{n} \rceil) - 1)n + ((i - 1) \bmod n) + 1. \quad \square$$

We give an example to illustrate Definition 1.

**Example 2:** Let  $\pi = (2, 1, 4, 3)$  and we want to compute  $\hat{\pi}^3$ . In this case,  $m = 4$  and  $n = 3$ . By Definition 1,  $\hat{\pi}^3 = (4, 5, 6, 1, 2, 3, 10, 11, 12, 7, 8, 9)$ . □

For an  $(n, k)$ -DIM  $f$ , the following definition defines  $f^m$  to be the resulting DIM by repeating Construction 1 over  $m$  copies of  $f$ . The operator  $\bullet$  is right associative.

**Definition 2** Let  $f$  be an  $(n, k)$ -DIM. For any  $m > 0$ , the mapping  $f^m$  of length  $mn$  is defined as

$$f^m = (f \bullet \dots \bullet (f \bullet (f \bullet f))). \quad \square$$

Next, for an  $(n, k)$ -DIM  $f$ , we define  $\tilde{f}^m$  to be a function from  $Z_2^{mn} \times S_m$  to  $S_{mn}$ .

**Definition 3** Let  $f$  be an  $(n, k)$ -DIM,  $\pi$  be a permutation of  $Z_m$ , and  $c$  be a binary vector of length  $mn$ . The function  $\tilde{f}^m$  from  $Z_2^{mn} \times S_m$  to  $S_{mn}$  is defined as

$$\tilde{f}^m(c, \pi) = \hat{\pi}^n f^m(c)$$

where the  $i$ th element of  $\tilde{f}^m(c, \pi)$  is  $\hat{\pi}^n(f^m(c)(i))$ . □

We give an example to illustrate Definition 3.

**Example 3:** Let  $f$  be the  $(5, 1)$ -DIM published in [3],  $\pi = (3, 1, 2)$ ,  $c = (1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1)$ . Since  $f(0, 0, 0, 1, 1) = (2, 5, 3, 1, 4)$  and  $f(1, 0, 0, 1, 1) = (2, 1, 3, 5, 4)$ , we have

$$\begin{aligned} f^2(1, 0, 0, 1, 1, 0, 0, 0, 1, 1) &= (9, 1, 3, 5, 4, 7, 10, 8, 6, 2) \text{ and} \\ f^3(1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1) &= (7, 1, 3, 5, 14, 4, 6, 8, 10, 9, 12, 15, 13, 11, 2). \end{aligned}$$

Furthermore,

$$\hat{\pi}^5 = (11, 12, 13, 14, 15, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10).$$

Thus,

$$\begin{aligned} \tilde{f}^3(c, \pi) &= \hat{\pi}^5(f^3(1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1)) \\ &= (2, 11, 13, 15, 9, 14, 1, 3, 5, 4, 7, 10, 8, 6, 12). \end{aligned} \quad \square$$

For an  $(n, k)$ -DIM  $f$ , the next definition defines the  $\otimes_f$  operator. When operating on a binary code of length  $mn$  and a PA of length  $m$ ,  $\otimes_f$  will generate a PA of length  $mn$ .

**Definition 4** Let  $n > 2k > 0$ ,  $m > 0$ ,  $f$  be an  $(n, k)$ -DIM,  $C$  be a binary code of length  $mn$ , and  $D$  be a permutation array of length  $m$ . We define  $C \otimes_f D = \{\tilde{f}^m(c, \pi) \mid c \in C, \pi \in D\}$ . □

The following theorem shows that the size of PA generated by  $\otimes_f$  is the product of the sizes of the binary code and the PA which  $\otimes_f$  operates on.

**Theorem 3** For  $n > 2k > 0$ ,  $m > 0$ , if  $f$  is an  $(n, k)$ -DIM,  $C$  is a binary code of length  $mn$ , and  $D$  is a permutation array of length  $m$ , then  $|C \otimes_f D| = |C| |D|$ .

**Proof:** This theorem is based on the fact: “For any  $c_A, c_B \in C$  and  $\pi_A, \pi_B \in D$ , if  $\tilde{f}^m(c_A, \pi_A) = \tilde{f}^m(c_B, \pi_B)$ , then  $c_A = c_B$  and  $\pi_A = \pi_B$ .”

We prove the fact now. Let  $\tilde{f}^m(c_A, \pi_A) = \tilde{f}^m(c_B, \pi_B)$  and  $\pi_A \neq \pi_B$ . There must be an index  $i$  such that  $(\pi_A)_i \neq (\pi_B)_i$ . Consider the  $((i - 1)n + k + 1)$ th elements of both  $\tilde{f}^m(c_A, \pi_A)$  and  $\tilde{f}^m(c_B, \pi_B)$ . Since

$$(i - 1)n + 1 \leq f^m(c_A)((i - 1)n + k + 1) \leq i n$$

we have

$$((\pi_A)_i - 1)n + 1 \leq \hat{\pi}_A^n(f^m(c_A)((i - 1)n + k + 1)) \leq ((\pi_A)_i - 1)n + n.$$

Similarly,

$$((\pi_B)_i - 1)n + 1 \leq \hat{\pi}_B^n(f^m(c_B)((i - 1)n + k + 1)) \leq ((\pi_B)_i - 1)n + n.$$

Since  $(\pi_A)_i \neq (\pi_B)_i$ , the  $((i - 1)n + k + 1)$ th elements of  $\tilde{f}^m(c_A, \pi_A)$  and  $\tilde{f}^m(c_B, \pi_B)$  are different. This is a contradiction. Once  $\pi_A = \pi_B$ ,  $\tilde{f}^m(c_A, \pi_A) = \tilde{f}^m(c_B, \pi_B)$  implies  $c_A = c_B$ .  $\square$

Next, we show that when operating on an  $(mn, d_1)$  binary code and an  $(m, d_2)$  PA,  $\otimes_f$  will generate an  $(mn, \min(d_1 + k, d_2(n - 2k), mn))$  PA.

**Theorem 4** For  $n > 2k > 0, m > 0$ , if  $f$  is an  $(n, k)$ -DIM,  $C$  is an  $(mn, d_1)$  binary code, and  $D$  is an  $(m, d_2)$  permutation array, then  $C \otimes_f D$  is an  $(mn, \min(d_1 + k, d_2(n - 2k), mn))$  permutation array.

**Proof:** From Theorem 3 we have known that if  $\tilde{f}^m(c_A, \pi_A) = \tilde{f}^m(c_B, \pi_B)$ , then  $c_A = c_B$  and  $\pi_A = \pi_B$ . That is, any permutation in  $C \otimes_f D$  is generated from a unique  $c \in C$  and an unique  $\pi \in D$ . Consider any two different permutations  $\alpha, \beta \in C \otimes_f D$  and  $\alpha \neq \beta$ . Let  $\alpha = \tilde{f}^m(c_\alpha, \pi_\alpha)$  and  $\beta = \tilde{f}^m(c_\beta, \pi_\beta)$ .

(1) If  $\pi_\alpha = \pi_\beta$ , then  $c_\alpha \neq c_\beta$ . Therefore,

$$d(\alpha, \beta) = d(\tilde{f}^m(c_\alpha, \pi_\alpha), \tilde{f}^m(c_\beta, \pi_\beta)) = d(\hat{\pi}_\alpha^n f^m(c_\alpha), \hat{\pi}_\alpha^n f^m(c_\beta)) = d(f^m(c_\alpha), f^m(c_\beta)) \geq \min(d_1 + k, mn).$$

(2) If  $\pi_\alpha \neq \pi_\beta$ , let  $I = \{i \mid (\pi_\alpha)_i \neq (\pi_\beta)_i, 1 \leq i \leq m\}$ . We have  $|I| \geq d_2$ . For each  $i \in I$  and  $j \in \{k + 1, k + 2, \dots, n - k\}$ , consider the  $((i - 1)n + j)$ th elements of  $\alpha$  and  $\beta$ . Since

$$(i - 1)n + 1 \leq f^m(c_\alpha)((i - 1)n + j) \leq i n$$

we have

$$((\pi_\alpha)_i - 1)n + 1 \leq \hat{\pi}_\alpha^n (f^m(c_\alpha)((i - 1)n + j)) \leq ((\pi_\alpha)_i - 1)n + n.$$

Similarly,

$$((\pi_\beta)_i - 1)n + 1 \leq \hat{\pi}_\beta^n (f^m(c_\beta)((i - 1)n + j)) \leq ((\pi_\beta)_i - 1)n + n.$$

Since  $(\pi_\alpha)_i \neq (\pi_\beta)_i$ , the  $((i - 1)n + j)$ th elements of  $\alpha$  and  $\beta$  are different. Thus

$$\begin{aligned} d(\alpha, \beta) &= d(\tilde{f}^m(c_\alpha, \pi_\alpha), \tilde{f}^m(c_\beta, \pi_\beta)) = d(\hat{\pi}_\alpha^n f^m(c_\alpha), \hat{\pi}_\beta^n f^m(c_\beta)) \\ &\geq \sum_{i \in I} \sum_{j=k+1}^{n-k} \delta(\hat{\pi}_\alpha^n (f^m(c_\alpha)((i - 1)n + j)), \hat{\pi}_\beta^n (f^m(c_\beta)((i - 1)n + j))) \\ &= |I|(n - 2k) \geq d_2(n - 2k). \end{aligned}$$

Parts (1) and (2) totally imply

$$d(\alpha, \beta) \geq \min(d_1 + k, d_2(n - 2k), mn). \quad \square$$

## 6. A NEW LOWER BOUND FOR THE SIZE OF PERMUTATION ARRAYS

Let  $P(n, d)$  denote the maximal size of an  $(n, d)$  permutation array (PA). Furthermore, we use  $A(n, d)$  to denote the maximal size of an  $(n, d)$  binary code. A lower bound of  $P(n, d)$  given in [3] shows that for  $n \geq 4$  and  $2 \leq d \leq n$ ,

$$P(n, d) \geq A(n, d - 1).$$

By a simple extension of the lower bound in [3], it is easy to prove that if there is an  $(n, k)$ -DIM,  $k \geq 1$  and  $k + 1 \leq d \leq n$ , then

$$P(n, d) \geq A(n, d - k).$$

This bound is not good enough. In the next theorem, a new better lower bound of  $P(n, d)$  is to be proposed (in a special form).

**Theorem 5** For  $n > 2k > 0$ ,  $m > 0$ , and  $k + 1 \leq d \leq mn$ , if there is an  $(n, k)$ -DIM, then

$$P(mn, d) \geq A(mn, d - k)P(m, \lceil d/(n - 2k) \rceil).$$

**Proof:** We only need to consider the case  $A(mn, d - k) > 0$  and  $P(m, \lceil d/(n - 2k) \rceil) > 0$ ; otherwise,  $P(mn, d) \geq 0$  is trivial. Let  $C$  be an  $(mn, d - k)$  binary code with the maximal size  $A(mn, d - k)$ , and  $D$  be an  $(m, \lceil d/(n - 2k) \rceil)$  PA with the maximal size  $P(m, \lceil d/(n - 2k) \rceil)$ . Let  $f$  be an  $(n, k)$ -DIM. By Theorem 4,  $C \otimes_f D$  is a PA of length  $mn$  with minimal distance

$$\min((d - k) + k, \lceil d/(n - 2k) \rceil(n - 2k), mn) \geq d.$$

By Theorem 3, the size of  $C \otimes_f D$  is

$$A(mn, d - k)P(m, \lceil d/(n - 2k) \rceil). \quad \square$$

This result is a new type of lower bound for  $P(n, d)$ . In many cases, this bound is much better than the lower bound published in [3]. Consider the  $k = 1$  case. Since it was shown  $|F_{n,1}| > 0$  for  $n \geq 4$  in [3], so the corollary below immediately follows.

**Corollary 1** For  $n \geq 4$ ,  $m > 0$ ,  $2 \leq d \leq mn$ ,  $P(mn, d) \geq A(mn, d - 1)P(m, \lceil d/(n - 2) \rceil)$ .  $\square$

We give an example to illustrate that even when  $k$  is only 1, the result of Corollary 1 is still better than the lower bound in [3].

**Example 4:** Consider  $P(24, 9)$ . The lower bound given in [3] is  $P(24, 9) \geq A(24, 8)$ . By Corollary 1,  $P(24, 9) \geq A(24, 8)P(6, \lceil 9/(4 - 2) \rceil) \geq A(24, 8)P(6, 5)$ .  $\square$

## ACKNOWLEDGEMENT

The author is indebted to the three anonymous reviewers for valuable comments on



0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

The middle-left part of the distance expansion matrix of  $Q_{16}$ :

0	0	0	0
360448	1384448	1654784	532480
0	47616	1357824	4156928
0	0	0	937216
0	0	0	1024
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

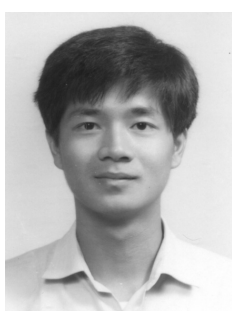
The middle-right part of the distance expansion matrix of  $Q_{16}$ :

0	0	0	0
0	0	0	0
6922240	4658688	1132544	74240
5039616	14822400	20462592	14085120
633856	5943552	22266624	45335040
0	605696	6858752	33374720
0	0	666368	9396480
0	0	0	978944
0	0	0	4096
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

The rightmost part of the distance expansion matrix of  $Q_{16}$ :

0	0	0	0
0	0	0	0
0	0	0	0

3853824	415744	20480	768
45777408	19457280	3530496	185344
80211968	93073920	41945088	6336000
51191296	132870656	136668928	44072192
14880768	87142400	192393216	126328832
1867776	29925376	147193856	195874816
49152	5267456	67174400	189915136
0	368640	18530304	124231680
0	0	2883584	56754176
0	0	196608	18153472
0	0	0	3932160
0	0	0	524288
0	0	0	32768



**Jen-Chun Chang (張仁俊)** received the B.S. and M.S. degrees in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1991, respectively. He received his Ph.D. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2000. He is a full professor of the Department of Computer Science and Information Engineering in National Taipei University, Taipei, Taiwan. His research interests include coding theory, cryptography, reliability theory, and algorithms.