Constructions of Distance-Almost-Increasing Mappings from Binary Vectors to Permutations

Jen-Chun Chang
Department of Computer Science & Information Engineering
National Taipei University
jcchang@csie.nctu.edu.tw

Shiao-Fan Chang
Department of Computer Science & Information Engineering
National Chiayi University
csf0427@hotmail.com

Abstract- *Mappings from the set of binary vectors of a fixed length to the set of permutations of the same length that increase the Hamming distance except the original Hamming distance is maximal (equal to the vector length) are useful for the construction of permutation codes. In this paper we propose recursive and explicit constructions of such mappings of length greater than 3 but not equal to 7. Some comparisons show that the new mappings have better distributions of distance increasing than other known distance-preserving mappings (DPMs). We also give some examples to illustrate the applications of these mappings to the constructions of permutation arrays (PAs).*

Keywords: *Code constructions, distance, mapping, distance-preserving mappings (DPMs), permutation arrays (PAs).*

1. Introduction

A distance-preserving mapping, shortly DPM, is a mapping from the set of all binary vectors of length n to the set of all n! permutations of $Z_n = \{1, 2, \ldots, n\}$ that preserves or increases the Hamming distance. Recently Chang (me) and others [1] proposed several constructions of DPMs and used their DPMs to improve some lower bounds on the size of permutation arrays. Lee [2] also devised a construction of DPMs of odd length. DPMs for vectors of length n are called n-DPMs.

The main objects studied in this paper are special distance-preserving mapping (DPMs) from the set of all binary vectors of length n to the set of all n! permutations of $Z_n = \{1, 2, \ldots, n\}$ that increase the Hamming distance except the original Hamming distance is maximal (equal to n). We call them n-DAIMs (distance-almost-increasing mappings for vectors of length n). From the view of DAIMs, for $n = 4$ or $n > 4$ and n mod $4 = 2$, Chang and others' n-DPMs are in fact n-DAIMs. Unfortunately, Lee's n-DPMs are not n-DAIMs.

In this paper we devise recursive and explicit constructions of n-DAIMs for all n greater than 3 but not equal to 7. Some comparisons of the distributions of distance increasing of the newly constructed DAIMs and other known DPMs are then given. In the last section, we also give some examples to illustrate the applications of these mappings to the constructions of permutation arrays (PAs).

2. Basic Notations

Let S_n be the set of all n! permutations of $Z_n = \{1, 2, \ldots, n\}$. A permutation $\pi : Z_n \to Z_n$ is represented by an n-tuple: $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ where $\pi_i = \pi(i)$. The set Z_2^n denotes the set of all binary vectors of length n. A binary vector $x \in Z_2^n$ is denoted by an n-tuple: $x = (x_1, x_2, \ldots, x_n)$ where x_i is the i-th bit of x.

The Hamming distance between two n-tuple $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ is the number of positions where they differ, and is represented by $d(\boldsymbol{a}, \boldsymbol{b})$.

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

A distance-almost-increasing mapping for length n (for short: an n-DAIM) is a mapping $f : Z_2^n \to S_n$ such that for any pair of binary vectors $x, y \in Z_2^n$,

if $d(x, y) = n$, then $d(f(x), f(y)) = d(x, y) = n$;
otherwise, $d(f(x), f(y)) > d(x, y)$.

Let $\boldsymbol{F}_n$ denote the set of all n-DAIMs.

3. Basic DAIMs of Length ≤ 6

For $n < 4$, it is obvious that $|\boldsymbol{F}_n| = 0$. From [1] we know that for m=2 or $m > 2$ and odd, $|\boldsymbol{F}_{2m}| > 0$. The $2m$-DAIMs are constructed with the following algorithm quoted from [1]:

Mapping algorithm for h_{2m}
Input: $(x_1, x_2, \ldots, x_{2m}) \in Z_2^{2m}$
Output: $(\pi_1, \pi_2, \ldots, \pi_{2m}) = h_{2m}(x_1, x_2, \ldots, x_{2m}) \in S_{2m}$

Begin
$(\pi_1, \pi_2, \ldots, \pi_{2m}) \leftarrow (1, 2, \ldots, 2m)$;
for i **from** 1 **to** m **do**
if $(x_i = 1)$ **then** swap (π_{2i-1}, π_{2i}) ;
for i **from** m+1 **to** $2m$ **do**
if $(x_i = 1)$ **then** swap (π_{i-m}, π_i) ;
End

♦

With this algorithm, it is clear that $|\boldsymbol{F}_n| > 0$ for n = 4 or $n > 4$ and n mod 4 = 2. In our constructions of DAIMs in following sections, the existence of n-DAIMs for $n \geq 7$ is based on the existence of n-DAIMs for $n \leq 6$. Here we define $r_4 = h_4$ and $r_6 = h_6$, thus both r_4 and r_6 are DAIMs. The 5-DAIM r_5 is defined by the following table (which is found with an customized efficient search program):

x	$r_5(x)$	x	$r_5(x)$
(0,0,0,0,0)	(1,2,3,4,5)	(1,1,1,1,1)	(5,1,2,3,4)
(0,0,0,0,1)	(4,5,1,2,3)	(1,1,1,1,0)	(5,1,4,3,2)
(0,0,0,1,0)	(4,5,3,1,2)	(1,1,1,0,1)	(5,4,2,3,1)
(0,0,0,1,1)	(2,5,3,1,4)	(1,1,1,0,0)	(5,2,4,3,1)
(0,0,1,0,0)	(4,3,5,2,1)	(1,1,0,1,1)	(3,1,2,5,4)
(0,0,1,0,1)	(2,3,5,4,1)	(1,1,0,1,0)	(3,1,4,5,2)
(0,0,1,1,0)	(4,3,5,1,2)	(1,1,0,0,1)	(3,2,1,5,4)
(0,0,1,1,1)	(4,3,2,1,5)	(1,1,0,0,0)	(1,2,4,5,3)
(0,1,0,0,0)	(3,5,1,4,2)	(1,0,1,1,1)	(2,1,5,3,4)
(0,1,0,0,1)	(1,5,2,4,3)	(1,0,1,1,0)	(2,1,4,3,5)
(0,1,0,1,0)	(3,5,4,1,2)	(1,0,1,0,1)	(2,4,5,3,1)
(0,1,0,1,1)	(3,5,2,1,4)	(1,0,1,0,0)	(4,2,5,3,1)
(0,1,1,0,0)	(5,3,1,4,2)	(1,0,0,1,1)	(2,1,3,5,4)
(0,1,1,0,1)	(5,3,2,4,1)	(1,0,0,1,0)	(3,1,4,2,5)
(0,1,1,1,0)	(5,3,4,1,2)	(1,0,0,0,1)	(2,4,3,5,1)
(0,1,1,1,1)	(5,3,2,1,4)	(1,0,0,0,0)	(2,4,1,5,3)

4. A Recursive Construction of DAIMs

In this section, we propose a recursive construction of DAIMs.

Construction 1: Let $f \in \boldsymbol{F}_m$ and $g \in \boldsymbol{F}_n$. For $x = (x_1, x_2, \ldots, x_{m+n})$, $f(x_1, x_2, \ldots, x_m) = (u_1, u_2, \ldots, u_m)$ and $g(x_{m+1}, x_{m+2}, \ldots, x_{m+n}) = (v_1, v_2, \ldots, v_n)$, we define $f \otimes g : Z_2^{m+n} \to S_{m+n}$ as

$f\otimes g(x) = ((1-x_{m+n})u_1+x_{m+n}(v_n+m),\ u_2,\ u_3,\ \ldots,\ u_{m-1},\ (1-x_m)u_m+x_m(v_1+m),\ x_mu_m+(1-x_m)(v_1+m),\ v_2+m,\ v_3+m,\ \ldots,\ v_{n-1}+m,\ x_{m+n}u_1+(1-x_{m+n})(v_n+m))$.

♦

The following example is helpful to illustrate the construction.

Example 1: Let $f = r_5 \in \boldsymbol{F}_5$ and $g = r_4 \in \boldsymbol{F}_4$. Using Construction 1, we get a mapping $r_5 \otimes r_4$. Consider $x = (1, 0, 1, 1, 1, 1, 1, 0, 1) \in Z_2^9$ for example. Since $r_5(1, 0, 1, 1, 1) = (2, 1, 5, 3, 4)$ and $r_4(1, 1, 0, 1) = (2, 3, 4, 1)$, thus we get

$r_5 \otimes r_4(1, 0, 1, 1, 1, 1, 1, 0, 1) = (6, 1, 5, 3, 7, 4, 8, 9, 2)$.

♦

In fact, mappings generated from Construction 1 are DAIMs. We prove this fact in the following theorem.

Theorem 1: If $f \in \boldsymbol{F}_m$ and $g \in \boldsymbol{F}_n$, then $f \otimes g \in \boldsymbol{F}_{m+n}$.
Proof: Let $x = (x_1, x_2, \ldots, x_{m+n})$, $y = (y_1, y_2, \ldots, y_{m+n})$, $f(x_1, x_2, \ldots, x_m) = (u_1, u_2, \ldots, u_m)$, $g(x_{m+1}, x_{m+2}, \ldots, x_{m+n}) = (v_1, v_2, \ldots, v_n)$, $f(y_1, y_2, \ldots, y_m) = (w_1, w_2, \ldots, w_m)$, and $g(y_{m+1}, y_{m+2}, \ldots, y_{m+n}) = (t_1, t_2, \ldots, t_n)$.

We divide the proof into 4 cases.

Case I: $x_m = y_m$ and $x_{m+n} = y_{m+n}$.

Let $d((x_1, x_2, \ldots, x_m), (y_1, y_2, \ldots, y_m)) = d_1 < m$, and $d((x_{m+1}, x_{m+2}, \ldots, x_{m+n}), (y_{m+1}, y_{m+2}, \ldots, y_{m+n})) = d_2 < n$. Then we have $d(x, y) = d_1+d_2$, $d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) \geq d_1+1$, and $d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) \geq d_2+1$. Considering $d(f\otimes g(x), f\otimes g(y))$, we have

$$\begin{aligned}
&d(f\otimes g(x), f\otimes g(y)) \\
&= d(\ ((1-x_{m+n})u_1+x_{m+n}(v_n+m),\ u_2,\ u_3,\ \ldots,\ u_{m-1},\ (1-x_m)u_m+x_m(v_1+m),\ x_mu_m+(1-x_m)(v_1+m),\ v_2+m,\ v_3+m,\ \ldots,\ v_{n-1}+m,\ x_{m+n}u_1+(1-x_{m+n})(v_n+m)), \\
&\qquad ((1-x_{m+n})w_1+x_{m+n}(t_n+m),\ w_2,\ w_3,\ \ldots,\ w_{m-1},\ (1-x_m)w_m+x_m(t_1+m),\ x_mw_m+(1-x_m)(t_1+m),\ t_2+m,\ t_3+m,\ \ldots,\ t_{n-1}+m,\ x_{m+n}w_1+(1-x_{m+n})(t_n+m))\ ) \\
&= d((u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n), \\
&\qquad (w_1, w_2, \ldots, w_m, t_1, t_2, \ldots, t_n)) \\
&= d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) + \\
&\quad d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) \\
&\geq d_1 + 1 + d_2 + 1 = d_1 + d_2 + 2 = d(x, y) + 2 \\
&> d(x, y).
\end{aligned}$$

Case II: $x_m = y_m$ and $x_{m+n} \neq y_{m+n}$.

Let $d((x_1, x_2, \ldots, x_m), (y_1, y_2, \ldots, y_m)) = d_1 < m$, and $d((x_{m+1}, x_{m+2}, \ldots, x_{m+n}), (y_{m+1}, y_{m+2}, \ldots, y_{m+n})) = d_2 \leq n$. Then we have $d(x, y) = d_1+d_2$, $d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) \geq d_1+1$, and $d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) \geq d_2$. Considering $d(f\otimes g(x), f\otimes g(y))$, we have

$$\begin{aligned}
&d(f\otimes g(x), f\otimes g(y)) \\
&= d(\ ((1-x_{m+n})u_1+x_{m+n}(v_n+m),\ u_2,\ u_3,\ \ldots,\ u_{m-1},\ (1-x_m)u_m+x_m(v_1+m),\ x_mu_m+(1-x_m)(v_1+m),\ v_2+m,\ v_3+m,\ \ldots,\ v_{n-1}+m,\ x_{m+n}u_1+(1-x_{m+n})(v_n+m)), \\
&\qquad ((1-y_{m+n})w_1+y_{m+n}(t_n+m),\ w_2,\ w_3,\ \ldots,\ w_{m-1},\ (1-x_m)w_m+x_m(t_1+m),\ x_mw_m+(1-x_m)(t_1+m),\ t_2+m,\ t_3+m,\ \ldots,\ t_{n-1}+m,\ y_{m+n}w_1+(1-y_{m+n})(t_n+m))\ ) \\
&\geq d((u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n), \\
&\qquad (w_1, w_2, \ldots, w_m, t_1, t_2, \ldots, t_n)) \\
&= d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) + \\
&\quad d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) \\
&\geq d_1 + 1 + d_2 = d_1 + d_2 + 1 = d(x, y) + 1 \\
&> d(x, y).
\end{aligned}$$

Case III: $x_m \neq y_m$ and $x_{m+n} = y_{m+n}$.

The proof of this case is similar to that of Case II.

Case IV: $x_m \neq y_m$ and $x_{m+n} \neq y_{m+n}$.

Let $d((x_1, x_2, \ldots, x_m), (y_1, y_2, \ldots, y_m)) = d_1 \leq m$, and $d((x_{m+1}, x_{m+2}, \ldots, x_{m+n}), (y_{m+1}, y_{m+2}, \ldots, y_{m+n})) = d_2 \leq n$. We further divide this case into two subcases.

Subcase IV-1: $d_1 + d_2 = m + n$.

In this subcase, it is clear that $d_1 = m$ and $d_2 = n$. Thus we have $d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) + d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) = d_1 + d_2 = m + n$. Considering $d(f\otimes g(x), f\otimes g(y))$, we have

$$\begin{aligned}
&d(f\otimes g(x), f\otimes g(y))\\
&= d(((1-x_{m+n})u_1+x_{m+n}(v_n+m), u_2, u_3, \ldots, u_{m-1}, (1-x_m)u_m+x_m(v_1+m), x_mu_m+(1-x_m)(v_1+m), v_2+m, v_3+m, \ldots, v_{n-1}+m, x_{m+n}u_1+(1-x_{m+n})(v_n+m)),\\
&\qquad ((1-y_{m+n})w_1+y_{m+n}(t_n+m), w_2, w_3, \ldots, w_{m-1}, (1-y_m)w_m+y_m(t_1+m), y_mw_m+(1-y_m)(t_1+m), t_2+m, t_3+m, \ldots, t_{n-1}+m, y_{m+n}w_1+(1-y_{m+n})(t_n+m)))\\
&\geq d((u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n),\\
&\qquad (w_1, w_2, \ldots, w_m, t_1, t_2, \ldots, t_n))\\
&= d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) +\\
&\quad d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n))\\
&= m + n = d(x, y).
\end{aligned}$$

This result does not negate $f\otimes g$ to be a DAIM since $d(x, y)$ has reached the maximum $m+n$.

Subcase IV-2: $d_1 + d_2 < m + n$.

In this subcase, it is clear that $d_1 < m$ or $d_2 < n$. Thus we have $d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) + d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n)) \geq d_1 + d_2 + 1$. Considering $d(f\otimes g(x), f\otimes g(y))$, we have

$$\begin{aligned}
&d(f\otimes g(x), f\otimes g(y))\\
&= d(((1-x_{m+n})u_1+x_{m+n}(v_n+m), u_2, u_3, \ldots, u_{m-1}, (1-x_m)u_m+x_m(v_1+m), x_mu_m+(1-x_m)(v_1+m), v_2+m, v_3+m, \ldots, v_{n-1}+m, x_{m+n}u_1+(1-x_{m+n})(v_n+m)),\\
&\qquad ((1-y_{m+n})w_1+y_{m+n}(t_n+m), w_2, w_3, \ldots, w_{m-1}, (1-y_m)w_m+y_m(t_1+m), y_mw_m+(1-y_m)(t_1+m), t_2+m, t_3+m, \ldots, t_{n-1}+m, y_{m+n}w_1+(1-y_{m+n})(t_n+m)))\\
&\geq d((u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n),\\
&\qquad (w_1, w_2, \ldots, w_m, t_1, t_2, \ldots, t_n))\\
&= d((u_1, u_2, \ldots, u_m), (w_1, w_2, \ldots, w_m)) +\\
&\quad d((v_1, v_2, \ldots, v_n), (t_1, t_2, \ldots, t_n))\\
&\geq d_1 + d_2 + 1 = d(x, y) + 1\\
&> d(x, y).
\end{aligned}$$

QED

Corollary 1: For all $m, n \geq 4$, $|\boldsymbol{F}_{m+n}| \geq |\boldsymbol{F}_m|\cdot|\boldsymbol{F}_n|$.

Proof: This corollary is directly based on two facts:

1. If $f_1 \neq f_2$, then $f_1\otimes g_1 \neq f_2\otimes g_2$, no matter g_1 and g_2 are different or not.
2. If $g_1 \neq g_2$, then $f\otimes g_1 \neq f\otimes g_2$.

We prove these two facts.

(Proof of fact 1): Let f_1 and f_2 be any two different m-DAIMs. Since $f_1 \neq f_2$, there must exist a binary vector of length m, say $x = (x_1, x_2, \ldots, x_m)$, such that $f_1(x) \neq f_2(x)$. For any two n-DAIMs g_1 and g_2, it is always true that $f_1\otimes g_1(x, y) \neq f_2\otimes g_2(x, y)$, where y can be any binary vector of length n. Therefore $f_1\otimes g_1 \neq f_2\otimes g_2$.

(Proof of fact 2): Let f be any m-DAIM, and g_1, g_2 are any two different n-DAIMs. Since $g_1 \neq g_2$, there must be a binary vector of length n, say $x = (x_1, x_2, \ldots, x_n)$, such that $g_1(x) \neq g_2(x)$. Let y be any binary vector of length m, it is obviously true that $f\otimes g_1(y, x) \neq f\otimes g_2(y, x)$. Therefore $f\otimes g_1 \neq f\otimes g_2$.

QED

Corollary 2: For all $n \geq 4$ and $n \neq 7$, $|\boldsymbol{F}_n| > 0$.

Proof: In Section 3 we have defined some basic n-DAIMs for $4 \leq n \leq 6$. Specifically, r_4, r_5, and r_6 are 4-DAIM, 5-DAIM, and 6-DAIM, respectively. That is, $|\boldsymbol{F}_n| > 0$ for $4 \leq n \leq 6$. Wth Corollary 1, the general statement in this corollary immediately follows by induction.

QED

5. An Explicit Construction of DAIMs

In this section, an explicit construction of DAIMs is to be proposed. We first describe the explicit construction with the following algorithm. Note that r_4, r_5, and r_6 are already known.

Construction 2:

Mapping algorithm for r_n ($n\geq 4$ and $n\neq 7$)

Input: $(x_1, x_2, \ldots, x_n) \in Z_2^{\,n}$

Output: $(\pi_1, \pi_2, \ldots, \pi_n) \in S_n$

Begin

```
  if (n ≤ 6) then
  {
     (π1, π2, …, πn) ← rn(x1, x2, …, xn) ;
     stop and exit ;
  }
  if (n mod 4 ≠ 3) then
  {
    k ← (n mod 4) + 4 ;
    (π1, π2, …, πk) ← rk(x1, x2, …, xk) ;
  }
  if (n mod 4 = 3) then
  {
    (π1, π2, …, π5) ← r5(x1, x2, …, x5) ;
    (π6, π7, …, π11) ← r6(x6, x7, …, x11) ;
    πi ← πi + 5, for all i = 6, 7, …, 11 ;
    if (x5 = 1) then swap (π5, π6) ;
    k ← 11 ;
  }
```

while $(k < n)$ **repeat**
{
  $(\pi_{k+1}, \pi_{k+2}, \pi_{k+3}, \pi_{k+4}) \leftarrow r_4(x_{k+1}, x_{k+2}, x_{k+3}, x_{k+4})$;
  $\pi_i \leftarrow \pi_i + k$, for all $i = k+1, k+2, k+3, k+4$;
  if $(x_k = 1)$ **then** swap (π_k, π_{k+1}) ;
  $k \leftarrow k + 4$;
}
if $(x_n = 1)$ **then** swap (π_n, π_1) ;
End

♦

In fact, r_n constructed in Construction 2 is an n-DAIM. This fact will be proved in the next theorem.

Theorem 2: For any positive integer $n \geq 4$ and $n \neq 7$, mapping r_n generated from Construction 2 is an n-DAIM, that is, $r_n \in \boldsymbol{F}_n$.
Proof: It is clear that for any $4 \leq n \leq 6$, r_n is always an n-DAIM. For $n \geq 8$, there are 4 cases. If (n mod 4) = 0, we construct r_n by combining two or more copies of the 4-DAIM r_4. If (n mod 4) = 1, we construct r_n by combining the 5-DAIM r_5 with one or more copies of the 4-DAIM r_4. If (n mod 4) = 2, we construct r_n by combining the 6-DAIM r_6 with one or more copies of the 4-DAIM r_4. If (n mod 4) = 3, we construct r_n by combining the 5-DAIM r_5 and the 6-DAIM r_6 with zero or more copies of the 4-DAIM r_4. Though the combining procedure is slightly different from that of Construction 1, the proof is similar and skipped here.

QED

6. Comparisons of our DAIMs and other mappings

The main objective in this section is to compare the distributions of Hamming distance increasing of different mappings, including DPMs from [1], DPMs of odd length from [2], and our new DAIMs. The notations we use to represent these mappings are listed in the following table.

N	n-DPM from [1]	n-DPM from [2]	n-DAIM
5	h_5^4	l_5	r_5
6	h_6	–	$r_6 = h_6$
7	h_7^6	l_7	–
8	h_8^6	–	r_8
9	h_9^6	l_9	r_9

It is clear that we only need to compare mappings r_5, r_8 and r_9 with other DPMs. For each mapping f of length n, we use an $n \times n$ matrix $(D_{i,j})_{n\times n}$ to show the distribution of distance increasing, where each element $D_{i,j}$ denotes the number of unordered pairs $\{x, y\}$ of binary vectors of length n such that $d(x, y) = i$ and $d(f(x), f(y)) = j$.

Case n=5

h_5^4

0	80	0	0	0
	0	96	64	0
		0	112	48
			16	64
				16

l_5

0	64	6	2	8
	4	68	64	24
		14	76	70
			22	58
				16

r_5

0	49	8	10	13
	0	68	68	24
		0	93	67
			0	80
				16

Case n=8

h_8^6

0	1024	0	0	0	0	0	0
	0	1280	2304	0	0	0	0
		0	1600	4160	1408	0	0
			0	1920	4992	1920	128
				0	2240	3840	1088
					128	1792	1664
						192	832
							128

r_8

0	768	256	0	0	0	0	0
	0	512	2432	512	128	0	0
		0	256	3840	2304	768	0
			0	256	4224	3584	896
				0	512	3840	2816
					0	768	2816
						0	1024
							128

Case n=9

h_9^6

0	2304	0	0	0	0	0	0	0
	0	3072	6144	0	0	0	0	0
		0	4160	12096	5248	0	0	0
			0	5376	16384	9472	1024	0
				0	6592	16128	8768	768
					256	6272	11520	3456
						448	4672	4096
							512	1792
								256

l_9

0	2048	0	0	0	0	6	68	182
	0	3076	4092	0	0	40	514	1494
		0	4176	8016	2144	126	1646	5396
			0	4848	9512	3560	3170	11166
				0	4492	7650	5462	14652
					4	3200	5496	12804
						82	1980	7154
							136	2168
								256

r_9

0	1360	496	224	208	16	0	0	0
	0	1008	4256	2112	944	864	32	0
		0	784	6784	7696	4320	1472	448
			0	512	8944	13168	7520	2112
				0	528	10256	15200	6272
					0	992	10432	10080
						0	1776	7440
							0	2304
								256

7. Applications to Permutation Arrays

It was shown in [1] that distance preserving mappings (DPMs and also DAIMs) are useful for the construction of permutation arrays (PAs). An (n, d) PA is a subset of S_n where the Hamming distance of any two distinct permutations is at least d. Let $P(n, d)$ denote the maximal size of such an (n, d) PA. Furthermore, we use $A(n, d)$ to denote the maximal size of an (n, d) binary code of length n and minimum distance d.

We give a different and simpler proof of the same lower bound of $P(n, d)$ that was proved in Theorem 5 in [1]. Here the lower bound is proved without the help of the Plotkin bound,

$$A(n,d) \le \frac{2d}{2d-n} \quad \text{for} \quad d > \frac{n}{2} .$$

Theorem 3: For $n \ge 4$, $n \ne 7$, and $2 \le d \le n$, $P(n, d) \ge A(n, d-1)$.

Proof: Let C be an $(n, d-1)$ binary code of size $A(n, d-1)$. We first explicitly construct an n-DAIM r_n by applying Construction 2, and then construct $r_n(C)$. It is obvious that $r_n(C)$ is an (n, d) permutation array. Therefore $P(n, d) \ge A(n, d-1)$.

QED

The following example uses the perfect [23, 12, 7] Golay code to construct a permutation array of minimum distance 8 and size 4096.

Example 3: It is known that the size of the perfect [23, 12, 7] Golay code reaches the upper bound $A(23, 7)=2^{12}=4096$. With the DAIM r_{23} constructed from the algorithm in Construction 2, we can obtain a (23, 8) permutation array. Thus $P(23, 8) \ge 4096$.

♦

Acknowledgment

This research was supported in part by the National Science Council of Taiwan, ROC under the contract NSC 93-2213-E-305-003.

References

[1] J.C. Chang, R.J. Chen, T. Klove, and S.C. Tsai, “Distance-preserving mappings from binary vectors to permutations,” *IEEE Trans. Information Theory*, vol. 49, no. 4, pp. 1054-1059, Apr. 2003.

[2] K. Lee, “New distance-preserving maps of odd length,” to appear in *IEEE Trans. Information Theory*, 2004.