Analysis of an algorithm used for finding Unipolar Orthogonal Codes

Vinutha J N¹, Hitesh Trivedi², Suraiya Tarannum³ and Harshavardhana M N¹
¹Cambridge Institute of Technology, Bengaluru, India
²ARDC, HAL, Bengaluru, India
³HKBK, College of Engineering, Bengaluru, India
vinutha.19lde07@cambridge.edu.in

Abstract. An Auto-Correlation and Cross-Correlation property of an Orthogonal Code enables the transfer of data with less interference. With more Orthogonal Codes, an Optical Fiber Communication channel can support a large number of users efficiently and reliably with high bandwidth. This paper analyses an algorithm used for finding Unipolar Orthogonal Codes using four major parameters (n, w, λa, and λc) where n is the length of a binary string, w is the weight of the string n (number of one’s in the string), λa is the Auto-Correlation, and λc is the Cross-Correlation. These four parameters jointly with the Johnson Bound (JA) condition are used in the algorithm. The paper illustrates the analysis of a possible number of Orthogonal Code Sets calculated using Orthogonal Codes generated from a software code that is based on the algorithm in consideration. The data analysis is carried out and the findings are summarized.

Keywords. Data Analysis, Orthogonal Codes, DOP, Correlation Code Matrix, DOP Matrix, Auto-Correlation, Cross-Correlation, Constraint, and Johnson Bound (JA).

1. Introduction
Orthogonal Codes are generated using four parameters (n, w, λa, and λc) where n is the length of the binary string, w is the weight of the string (count of one’s in the string), λa is the Auto-Correlation, and λc is the Cross-Correlation.

1.1. Auto-Correlation and Cross-Correlation
The λa and λc are calculated using DOP Matrix and used to form a Correlation Code Matrix. λa and λc are defined as for formula (1) and (2) presented in [1,3].

By considering, two binary strings P and Q,

\[ P = (P_0, P_1, \ldots, P_{n-1}) \]
\[ Q = (Q_0, Q_1, \ldots, Q_{n-1}) \]; where \( P_i, Q_i \in (0,1) \).

Auto-Correlation is defined as,

\[ \lambda_a \geq \sum_{t=0}^{n-1} P_t P_{t\oplus t} \]
when, \( \tau = 0 \), \( \lambda_a = w \), which is the peak of the Auto-Correlation. (1)

Cross-Correlation is defined as,

\[
\lambda_c \geq \sum_{t=0}^{n-1} P_t Q_{t\oplus \tau} \quad \text{and} \quad \lambda_c \geq \sum_{t=0}^{n-1} Q_t P_{t\oplus \tau} \quad \text{for} \quad 0 < \tau \leq n - 1
\]

where \( t\oplus \tau \) is equal to the convolution of \((t + \tau) \mod n\). (2)

There are two strings \( P \) and \( Q \) in which \( w \) is overlapped with the circular shifted version for the \( P \) string itself. The \( t\oplus \tau \) is the reason for the circular shifts; it is represented in formula (1). Formula (2) is used to determine the \( w \) overlapped with a circular shifted version of the \( P \) and \( Q \) string [3].

The method of generation of maximum Orthogonal Codes uses four parameters \((n, w, \lambda_a, \text{and } \lambda_c)\). By selecting Auto-Correlation \((\lambda_a)\) and Cross-Correlation \((\lambda_c)\) constraints from the Correlation Code Matrix, the maximum Unipolar Orthogonal Code generation is carried out. For the Orthogonal Code generation Johnson Bound \((J_A)\) condition is used.

### 1.2. Johnson Bound

If \( \lambda_a=\lambda_c=\lambda \), where \( 1 \leq \lambda \leq w - 1 \), then \( J_A(n, w, \lambda) \), \( J_B(n, w, \lambda) \) and \( J_C(n, w, \lambda) \) (Johnson Bounds) [3,4], are used for computing Maximum Orthogonal Codes \((Z)\) possible for different conditions of \( w, n \), and \( \lambda \).

If \( w < n \) then \( J_A \) is given as,

\[
Z \leq \left[ \frac{1}{w} \right] = J_A(n, w, \lambda)
\]

If \( w > n\lambda \) then \( J_B \) is given as,

\[
Z \leq \min \left( 1, \frac{w\lambda}{w^2 - n\lambda} \right) = J_B(n, w, \lambda)
\]

If \((w-k)^2 > (n-k)(\lambda-k)\), \( k \) an integer, \( 1 \leq k \leq \lambda - 1 \) then \( J_C \) is given as,

\[
Z \leq \left[ \frac{1}{w} \right] = J_C(n, w, \lambda)
\]

\[
h = \min \left( n-k, \frac{(n-k)(\lambda-k)}{(w-k)^2 - (n-k)(\lambda-k)} \right)
\]

\([x]=\) largest integer value less than ‘\(x\’.

There are three conditions in Johnson Bound \( J_A, J_B, \) and \( J_C \) as given above. The algorithm under analysis is limited to using only Johnson Bound \( J_A \) for which \( w < n \).
2. Analyzed Algorithm
There is a method to generate a maximum number of Unipolar Orthogonal Codes. The number of users allowed to use the Optical Fiber Communication channel is equal to the total number of Orthogonal Codes generated.

2.1. Steps used to generate the maximum Orthogonal Codes are the following
- Generate DOP and DOP Matrix using \( n \) and \( w \).
- Calculate \( \lambda_a \) and \( \lambda_c \) using DOP Matrix.
- Form a Correlation Code Matrix using \( \lambda_a \) and \( \lambda_c \) values. It will be of a size equivalent to the number of DOP.
- The \( \lambda_a \) and \( \lambda_c \) constraints are selected from the Correlation Code Matrix generated in step 2.
- Using \((n, w, \lambda_a, \lambda_c)\), the maximum number of Orthogonal Codes \((Z)\) are generated with a Johnson Bound \((J_A)\) condition.
- The maximum possible numbers of Orthogonal Sets are generated using Orthogonal Codes generated in step 4. While generating maximum Orthogonal Sets, the Johnson Bound \((J_A)\) condition \(\lambda_a = \lambda_c = \lambda\) is used, given in introduction Para I.

Among the generated Orthogonal Sets, only one of these sets will be used for applications in incoherent Optical CDMA [1].

**Example:**
Table 1 shows some of the combinations of \( n \) and \( w \) with condition \((w < n)\).

| Table 1. Few examples for values \( n \) and \( w \) |
|------------------------------------------|
| \( n = 7 \) and \( w = 3 \)               |
| 1110000                                   |
| 1101000                                   |
| 1100100                                   |
| 0101010                                   |
| \( n = 10 \) and \( w = 2 \)             |
| 0011000000                                |
| 0101000000                                |

2.2. DOP Generation
Firstly, compute all the **Differences of Positions**, DOP \([2, 3]\) for given input values of \( n \) and \( w \). The DOP \((x_1, x_2, x_3, \ldots, x_w)\) presents combinations of \( n \) and \( w \) as shown in table 1. There are two standard formats followed for the generation of DOP for given \( n \) and \( w \). Following conditions (i) and (ii) are used in DOP generation.

\[
i. \quad x_w = (x_1, x_2, x_3, \ldots, x_{w-1})
\]
\[
ii. \quad \left\lfloor \frac{n}{w} \right\rfloor \leq x_w \leq (n-w+1)
\]

\([x] = \text{ceiling integer} > x\)

The following points are to be considered before generating DOP:
- Position of one’s in a binary string, for example, 01101000, is \((2, 3, 5)\)
- But the Differences of Positions (DOP) for the same string is \((1, 2, 5)\)

Here DOP is defined as the numbers of positions that are there between two 1’s from left to right and circular shift.

![Figure 1. DOP for ‘01101000’ is (1,2,5)](image-url)
Figure 1 shows the DOP relation concerning for to position of 1’s in the binary string. The table 2 shows the DOP generation for n = 8 and w = 3.

Table 2. DOP for inputs n and w for n = 8 and w = 3

| Differences of Positions (DOP) | 01101000 (1,2,5) |
|--------------------------------|------------------|
| 7th bit                       | 0                |
| 6th bit                       | 1                |
| 5th bit                       | 1                |
| 4th bit                       | 0                |
| 3rd bit                       | 1                |
| 2nd bit                       | 0                |
| 1st bit                       | 0                |
| 0th bit                       | 0                |

2.3. Correlation Code Matrix Generation

To generate a Correlation Code Matrix, initially, it is required to form a DOP Matrix. Using the DOP matrix, \( \lambda_a \) and \( \lambda_c \) are calculated, which are the elements of the Correlation Code Matrix. The DOP Matrix is formed using the conditions presented under DOP Generation and the following is the format of the DOP matrix.

\[
\begin{bmatrix}
 x_1 & x_1 + x_2 & \ldots & x_1 + x_2 + \ldots + x_{w-1} \\
 x_2 & x_2 + x_3 & \ldots & x_2 + x_3 + \ldots + x_w \\
 x_3 & x_3 + x_4 & \ldots & x_3 + x_4 + \ldots + x_1 \\
 \vdots & \vdots & \ddots & \vdots \\
 x_w & x_w + x_1 & \ldots & x_w + x_1 + \ldots + x_{w-2}
\end{bmatrix}
\]

Table 3 contains DOP Matrix which is computed using the above format. The DOP Matrix for w = 3 are given as follows:

\[
\begin{bmatrix}
 x_1 & x_1 + x_2 \\
 x_2 & x_2 + x_3 \\
 x_3 & x_3 + x_1
\end{bmatrix}
\]

For a binary string = 01101000, the DOP is (1,2,5) and the DOP Matrix is given as:

\[
\begin{bmatrix}
 1 & 3 \\
 2 & 6 \\
 5 & 7
\end{bmatrix}
\]

2.3.1. Auto-Correlation (\( \lambda_a \)) computation

Table 3 shows the computation of Auto-Correlation using the DOP Matrix. Compare two different rows of the same DOP matrix. Note the common elements (N) between them. Compute \( \lambda_a \) using condition: \( i \neq j \) & \( N_{ij} = N_{ji} \) where ‘i’ and ‘j’ are row indices of DOP Matrix. \( \lambda_a = \text{maximum number of common elements} + 1 \), \([3]\)

For example, if string = 01101000, w = 3, \( \lambda_{11} = N_{ij} \), \( i \neq j \)

- N12 = compare 1st row (1 3) with 2nd row (2 7) of DOP Matrix = 0
- N13 = compare 1st row (1 3) with 3rd row (5 6) of DOP Matrix = 0
- N23 = compare 2nd row (2 7) with 3rd row (5 6) of DOP Matrix = 0

Therefore, \( \lambda_a = \lambda_{11} = 0 + 1 = 1 \).
Table 3. Computation of $\lambda a$ and $\lambda c$ using DOP Matrix

| Code (n=8, w=3) | DOP | DOP Matrix | $\lambda a$ | $\lambda c$ |
|----------------|-----|------------|-------------|-------------|
| 11100000       | (1,1,6) | $\begin{bmatrix} 1 & 2 \\ 1 & 7 \\ 6 & 7 \end{bmatrix}$ | 2           | 2           |
| 01101000       | (1,2,5) | $\begin{bmatrix} 1 & 3 \\ 2 & 7 \\ 5 & 6 \end{bmatrix}$ | 1           | 2           |
| 11001000       | (1,3,4) | $\begin{bmatrix} 1 & 4 \\ 3 & 7 \\ 4 & 5 \end{bmatrix}$ | 2           | 2           |

2.3.2. Cross-Correlation ($\lambda c$) computation

Table 3 also shows the computation of Cross-Correlation using the DOP Matrix. Cross-Correlation is given by:

$$\lambda c = \text{Max} \{N_{ij}\} + 1 \ \forall \ i,j$$

It increases the complexity; hence formula can be used as:

$$\lambda c = \text{Max} \{N_{ij}\} + 1 \ \forall \ i$$

The Correlation Code Matrix is formed with $\lambda a$ and $\lambda c$ as its elements. $\lambda a$ are the diagonal elements and $\lambda c$ are the remaining elements of the Correlation Code Matrix. The format for the Correlation Code Matrix is as follows:

$$\begin{bmatrix}
\text{code1} & code2 & code3 & codeX & codeN \\
\lambda_{11} & \lambda_{12} & \lambda_{13} & \ldots & \lambda_{1N} \\
\lambda_{21} & \lambda_{22} & \lambda_{23} & \ldots & \lambda_{2N} \\
\lambda_{31} & \lambda_{32} & \lambda_{33} & \ldots & \lambda_{3N} \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\lambda_{N1} & \lambda_{N2} & \lambda_{N3} & \ldots & \lambda_{NN}
\end{bmatrix}$$

2.4. Maximum Orthogonal Codes (Z) Generation:

From the Correlation Code Matrix given above, Constraints $\lambda = \lambda a = \lambda c$ are taken. Using parameters n, w, and $\lambda$, computation of a maximum number of orthogonal codes are carried out by following the Johnson Bound ($J_{\lambda}$) condition.

2.5. Maximum Orthogonal Code Sets Generation:

The maximum possible number of Orthogonal Code Sets are calculated using the maximum number of Orthogonal Codes generated using n, w, $\lambda a$, and $\lambda c$ [2].

3. Findings of the Analysis

To evaluate the performance of the considered algorithm, coding and data analysis are carried out using MATLAB R2018a. Also, a parallel analysis is carried out using Python 2.7 Language. The maximum possible Orthogonal Sets generated are 375168 for 6 Orthogonal Codes for n = 39 and w =3.

3.1. The following are the key points of the data analysis carried out:

A. The n < w is not a valid entry as per the Johnson Bound ($J_{\lambda}$) condition which is used for the generation of maximum Orthogonal Codes (Z).
B. Case $w = 1$:
Based on condition $1 \leq \lambda \leq w-1$, giving $w=1$ as input is invalid as the above condition is not satisfied.

C. Case $w = 2$:
- For any value of $n$, constraint $\lambda=1$, the $Z$ increases with the value of $n$.
- For any value of $n$, all the elements of the Correlation Code Matrix are $1$. Therefore, the $\lambda a$ and $\lambda c$ constraint equal to $1$ is used. It is observed that the resultant maximum Unipolar Orthogonal Code is equal to the Size of the Correlation Code Matrix.
- For any value of $n$, the constraint $\lambda=2$, cannot be used, if it is used $Z$ becomes infinite, and the ‘for loop’ error arises in software code. The reason is that there is no element in Correlation Code Matrix which is equal to $2$. So, it is important to observe the Correlation Code Matrix to select constraint $\lambda$ for generating $Z$.
- For $w=2$, constraint $\lambda=1$, the code runs without any errors for any value of $n$. It is observed that $Z$ is half of the value of $n$ for most of the cases.
- The plot of $Z$ is shown in figure 2 for various lengths of binary strings from $n=3$ to $101$, $w=2$, and constraint $\lambda=1$. But the maximum number of possible Orthogonal Code Sets is computed to be only $1$, for all the cases.

![Figure 2. Maximum Orthogonal Codes for $w=2$](image)

D. Case $w = 3$:
- The generation of $Z$ for constraint $\lambda=1$ is carried out for $n=4$ to $n=40$.
- Whenever $Z$ value exceeding $6$ for constraint $\lambda=2$ the possible Orthogonal Code Sets are not generating as Code Sets Matrix is exceeding its index.
- The plot of $Z$ for various lengths of binary strings from $n=4$ to $40$, $w=3$, constraint $\lambda=1$ is shown in figure 3.
Figure 3. Maximum Orthogonal Sets Possible and Maximum Orthogonal Codes for \( w=3 \)

E. Case \( w=4 \):
- The generation of \( Z \) for constraint \( \lambda=3 \) is carried out for \( n=5 \) to \( n=30 \).
- The possible maximum Orthogonal Code Sets are generated only for \( n=5 \) and \( n=6 \).
- For other values of \( n \) from 7 to 30, the Orthogonal Code Sets are not generated as the Code Sets Matrix is exceeding its index.
- Figure 4 shows the plot of Maximum Orthogonal Codes for various lengths of binary strings from \( n=5 \) to 30, \( w=4 \) and constraint \( \lambda=3 \).

Figure 4. Maximum Orthogonal Codes for \( w=4 \)

F. For larger values of \( w \), eg. \( w=5,6 \), etc. the \( Z \) can be generated but Orthogonal Code Sets cannot be generated. Hence, the algorithm is analyzed for \( w=2 \) to 4.

G. It is also found that there is a need for an industrial-grade PC with high-speed computing hardware for this algorithm to be faster and efficient. This will help users to find out the maximum number of Orthogonal Code Sets possible rapidly.
H. The maximum Orthogonal Codes (Z) are equal to the highest possible number of users of the optical fiber communication channel. The maximum possible Orthogonal Code Sets are generated which can be used for applications in incoherent CDMA systems [3].

4. Conclusion

The considered algorithm presents a simplified way to obtain possible Orthogonal Code Sets from the maximum number of Unipolar Orthogonal Codes. The algorithm was coded in Matlab and Python separately. Python code is found to be sluggish compared to the Matlab code. The computer processor becomes idle for a long time while running the iterations in Python. It was assumed that Python code will give faster results, but it did not because the complexity of ‘for loop’ remains the same for an algorithm. Many iterations were carried out in both the software. Data Analysis of the output offered many findings, which are mentioned in Para IV. These findings have also led to the unearthing of new problems which can be further explored and solved by engineers and researchers working in this technical field.

5. References

[1] Chung F R K, Salehi J A and Wei V K 1989 Optical orthogonal codes: design, analysis and applications in *IEEE Transactions on Information Theory* 35 pp 595-604

[2] Chauhan R C S, Asthana R and Singh Y N 2010 A general algorithm to design sets of all possible one dimensional unipolar orthogonal codes of same code length and weight *IEEE International Conference on Computational Intelligence and Computing Research* pp. 1-7

[3] Harshavardhana M N and Tarannum S 2019 Algorithm for Enhanced Design Set of all Possible Unipolar Orthogonal Codes *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* pp. 1-3

[4] Omrani R, Kumar P V 2006 Codes for Optical CDMA. In: Gong G., Helleseth T., Song HY., Yang K. (eds) Sequences and Their Applications – SETA 4086