A $k$-Cube Graph Construction for Mappings from Binary Vectors to Permutations

Khmaies Ouahada and Hendrik C. Ferreira
Department of Electrical and Electronic Engineering Science
University of Johannesburg, P.O. Box 524, Auckland Park, 2006, South Africa
Email: {kouahada, hcerreir}a@uj.ac.za

Abstract—A new graph theoretic construction mapping binary sequences to permutation sequences is presented. The $k$-cube graph construction has reached the upper bound on the sum of the distances for certain values of the length of the permutation sequence. This contributed in a better way to understand the distance-reducing mapping, which was not investigated before.

I. INTRODUCTION

The $k$-cube graph [1]–[2] is a very popular interconnection topology for parallel computers’ networks due to its ability to exploit communication locality found in many parallel applications to reduce message latency [3]–[5].

Ferreira and Vinck [6]–[7] have introduced distance-preserving mappings and combined permutation trellis codes with the M-FSK modulation scheme, in power-line communications [8]–[10]. Since then, much research has been done on permutation mappings to find new algorithms and better mappings [11]–[15].

Section II introduces distance-preserving mappings. In Section III we provide definitions [1]–[2] and notations to be used and introduce our new distance metric, which is called the cube distance. Section IV presents the cube construction. Section V presents the lower bound on the number of permutations in a distance-reducing mapping and emphasizes in a very simple way this type of mapping. We conclude with some final remarks.

II. DISTANCE-PRESERVING MAPPINGS

Distance-preserving mappings is a coding technique, which maps the outputs of a convolutional code to other codewords from a code with lesser error-correction capabilities. The purpose behind this technique is to, firstly, obtain suitably constrained output code sequences and, secondly, to exploit the error correction characteristics of the new code with the use of the Viterbi algorithm [11].

In [6] and [11] it was shown how the output binary $n$-tuple code symbols from an $R = \frac{m}{n}$ convolutional code can be mapped to non-binary $M$-tuple permutation code symbols, thereby creating a permutation trellis code.

To illustrate this idea, we use the binary convolutional code with octal generators 5 and 7 (see e.g. [16]) as base code. At the output of the encoder, we can map the set of binary 2-tuple code symbols onto a set of permutation $M$-tuples.

Let $[M] = \{1, 2, \ldots, M\}$. $S_M$ denotes the set of all permutations of $[M]$ and $\mathbb{Z}_2^n$ the set of all binary vectors of length $n$. We use $x^i$ to denote the $i$-th sequence from the possible outputs of the convolutional base code. Similarly, we use $y^i$ to denote the $i$-th permutation codeword in the mapping codebook. The matrices $D = [d_{ij}]$, with $d_{ij} = d_H(x^i, x^j)$ and $E = [e_{ij}]$, with $e_{ij} = d_H(y^i, y^j)$ are respectively the Hamming distance matrices for the binary input sequences and the permutation codewords of the resulting mapped code (see [11] for an expanded definition). The sum of all the distances in $E$ is denoted by $|E|$ and plays a role in the error correcting capabilities, as was shown in [15]. For $M = 4$ as presented in Fig. 1 we have,

$$D = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix},$$

here, $e_{ij} = d_{ij} + 1$ since for all $i \neq j$, the map of $\{0, 01, 10, 11\}$ onto $\{1243, 1342, 4213, 4312\}$, guarantees an increase of one unit of distance per step between any two unmerged paths in the trellis diagram of the resulting permutation trellis code, when comparing it to the base code. This example thus in fact represents a distance increasing map.

In general, three types of DPMs can be obtained, depending on how the Hamming distance is preserved.

- In the case where $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{1, 2, \ldots\}$, $\forall i \neq j$ we call such mappings distance-increasing mappings (DIMs).
- In the case where $e_{ij} \geq d_{ij}$, $\forall i \neq j$ and equality achieved at least once, we have distance-conserving mappings (DCMs).
In the case where \( e_{ij} \geq d_{ij} + \delta, \delta \in \{-1, -2, \ldots\}, \forall i \neq j \), we have distance-reducing mappings (DRMs).

We use \( Q(M, n, \delta) \) to denote the DPMs from binary sequences of length \( n \) to permutation sequences of length \( M \) making use of the \( k \)-cube graph construction. The type of mapping is indicated by \( \delta \), with \( \delta > 0 \) for DIMs, \( \delta = 0 \) for DCMs and \( \delta < 0 \) for DRMs.

III. \( k \)-CUBE GRAPH THEORY: PRELIMINARY

**Definition 1** A graph \( G = (V, E) \) is a mathematical structure consisting of two finite sets \( V \) and \( E \). The elements of \( V \) are called vertices, and the elements of \( E \) are called edges. Each edge has a set of one or two vertices associated with it, which are called its endpoints.

**Definition 2** A graph \( G' = (V', E') \) is a subgraph of another graph \( G = (V, E) \) iff \( V' \subseteq V \) and \( E' \subseteq E \).

**Definition 3** A bipartite graph is a graph whose vertex-set can be split into sets \( A \) and \( B \) in such a way that each edge of the graph joins a vertex in \( A \) to a vertex in \( B \). The \( k \)-cube graph is a complete bipartite graph since every vertex in \( A \) is adjacent to every vertex in \( B \).

**Definition 4** The \( k \)-cube is a graph with vertex set \( \{0, 1\}^k \) and with an edge between each pair of sequences that differ in exactly one position (i.e. have Hamming distance 1). It has \( 2^k \) vertices, each of which has the degree \( k \) as shown for the example of 4-cube in Fig. 2.

**Definition 5** The number of edges in a \( k \)-cube is \( k \times 2^{k-1} \).

**Definition 6** A cube distance denoted by \( D_Q \), is the absolute value of the difference between two integer values corresponding to two vertices \( v_1 \) and \( v_2 \) in a cube. It is defined as:
\[
D_Q(v_1, v_2) = |(v_1 - v_2)|, \quad 1 \leq v_1, v_2 \leq M.
\]

**Definition 7** A distance grouping, denoted by \( DG_i \) with \( 1 \leq i \leq k \), is a set of all pairs of symbols with equidistant edges. We have \( k \) distance groupings for each \( k \)-cube graph.

IV. \( k \)-CUBE GRAPH CONSTRUCTION

New terminologies should be introduced in this section since the \( k \)-cube graph is used in distance-preserving mappings. The vertices will be called as the symbols of a permutation sequence and the edges as the transpositions or swaps between them.

The value \( M \) representing the length of the permutation sequence is the same as the number of the vertices in a cube graph. In the cube graph the value \( M \) is always equal to \( 2^k \). Since we are using permutation sequences with lengths not always equal to \( 2^k \), we have defined the value of \( k \) as:
\[
k = \lfloor \log_2 M \rfloor.
\]

The idea of the \( k \)-cube graph construction is based on the concept of the cube distance. Fig. 3 shows different ways of presenting distances in a cube. It is clear that the Hamming distance whether for binary sequences or integer sequences does not help in differentiating the edges since all of them remain equidistant. In contrary the cube distance changes all edges’ distances to a geometric series of ratio 2. The distances are from \( 2^0 \) till \( 2^{k-1} \). Fig. 3 shows that when using the cube distance approach, all parallel edges are equidistant.

We can summarize the properties of a \( k \)-cube based on the cube distance as follow:

1) Geometric series with a ratio of 2, from \( 2^0 \) till \( 2^{k-1} \);
2) Any vertex in a \( k \)-cube graph has a degree \( k \) and all its distances are in a geometric series from \( 2^0 \) till \( 2^{k-1} \);
3) Parallel edges are equidistant. Diagonal edges are considered as parallel to one another;
4) Distances in a \( k \)-cube increase when \( k \) increases;
5) \((k - 1)\)-cube graph is a subgraph of the \( k \)-cube graph.

**Example 1** If we take the case of \( M = 4 \), which corresponds to 2-cube graph, we can break down our symbols into two distance groupings as presented in (3). We denote by \( D_Q(DG_i) \), with \( 1 \leq i \leq k \), the corresponding cube-distance value of \( DG_i \).
\[
D_Q(DG_1) = 2^0 \to DG_1 = \{(1, 2); (3, 4)\}
\]
\[
D_Q(DG_2) = 2^1 \to DG_2 = \{(1, 3); (2, 4)\}
\]

Fig. 4 shows all distance groupings in a 2-cube graph. We correspond an input \( x_j \), \( 1 \leq j \leq k \times 2^{k-1} \), to each pair having the same distance grouping. The total number of inputs is \( n = 4 \), which is the output of the base code.

![Fig. 2. Spacial presentation of a 4-cube](image)

![Fig. 3. Distance presentation of a 3-cube](image)

![Fig. 4. Distance groupings of a 2-cube](image)
Starting with any permutation sequence, the mapping algorithm based on the swapping of the symbols of the equidistant edges is presented as follow:

Mapping algorithm for $Q(4, 4, 0)$
Input: $(x_1, x_2, x_3, x_4) \in \mathbb{Z}_2^4$
Output: $(y_1, y_2, y_3, y_4) \in S_4$
Start Sequence: $(1, 2, 4, 3) \rightarrow (y_1, y_2, y_3, y_4)$
begin
  if $x_1 = 1$ then swap($y_1, y_2$)
  if $x_2 = 1$ then swap($y_3, y_4$)
  if $x_3 = 1$ then swap($y_1, y_3$)
  if $x_4 = 1$ then swap($y_2, y_4$)
end.

It is clear that in this example the mapping is considered as conserving since $\delta = 0$.

We can combine the equidistant edges and their corresponding symbol pairs to reduce the number of transpositions and thus the number of the base code’s outputs $n$ to have $n < M$. In this case we have a DIM as depicted in the following algorithm:

Mapping algorithm for $Q(4, 3, 1)$
Input: $(x_1, x_2, x_3) \in \mathbb{Z}_2^3$
Output: $(y_1, y_2, y_3, y_4) \in S_4$
Start Sequence: $(1, 2, 4, 3) \rightarrow (y_1, y_2, y_3, y_4)$
begin
  if $x_1 = 1$ then swap($y_1, y_2$)
  if $x_2 = 1$ then swap($y_3, y_4$)
  if $x_3 = 1$ then swap($y_1, y_3$)
end.

Example 2 In the case of $M = 8$, it is clear from Fig. 5 that we have three distance groupings $D_Q(DG_1) = 2^0$, $D_Q(DG_2) = 2^1$ and $D_Q(DG_3) = 2^2$ presented as follow:

$D_Q(DG_1) = 2^0 \rightarrow DG_1 = \{(1,2); (3,4); (5,6); (7,8)\}$
$D_Q(DG_2) = 2^1 \rightarrow DG_2 = \{(1,3); (2,4); (5,7); (6,8)\}$
$D_Q(DG_3) = 2^2 \rightarrow DG_3 = \{(1,5); (2,6); (3,7); (4,8)\}$

The corresponding algorithm for this case is,

Mapping algorithm for $Q(8, 12, -4)$
Input: $(x_1, x_2, \ldots, x_{12}) \in \mathbb{Z}_2^{12}$
Output: $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8) \in S_8$
Start Sequence: $(1, 2, 3, 4, 5, 6, 7, 8) \rightarrow (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$
begin
  if $x_1 = 1$ then swap($y_1, y_2$)
  if $x_2 = 1$ then swap($y_3, y_4$)
  if $x_3 = 1$ then swap($y_5, y_6$)
  if $x_4 = 1$ then swap($y_7, y_8$)
  if $x_5 = 1$ then swap($y_1, y_3$)
  if $x_6 = 1$ then swap($y_2, y_4$)
  if $x_7 = 1$ then swap($y_5, y_7$)
  if $x_8 = 1$ then swap($y_6, y_8$)
  if $x_9 = 1$ then swap($y_1, y_5$)
end.

We can see that $n = 12$, which is larger than $M = 8$. In this case we have $\delta = -4$ and the mapping is considered as reducing.

Example 3 Here we take the case of $M \neq 2^k$ (e.g. $M = 5$), which could be identified by using some of the graph theory definitions or the concept of the cube distance that we have introduced.

A. Using the Subgraph Theory

We consider the 3-cube graph, which has 8 vertices representing the length of the permutation sequence. We can reduce the length of the permutation sequence by eliminating the vertices that we do not need. In this case, the vertices are 6, 7 and 8. Then we eliminate all the edges connected to those vertices, which means we are not interested in their corresponding swaps. Fig. 6 shows that the vertex 5 is connected only to the vertex 1 and the final number of edges, which corresponds to the number of swaps is five. In this case we are dealing with conserving mapping since $\delta = 0$.

The corresponding algorithm for this DCM is,

Mapping algorithm for $Q(5, 5, 0)$
Input: $(x_1, x_2, x_3, x_4, x_5) \in \mathbb{Z}_2^5$
Output: $(y_1, y_2, y_3, y_4, y_5) \in S_5$
Start Sequence: $(1, 2, 3, 4, 5) \rightarrow (y_1, y_2, y_3, y_4, y_5)$
begin
  if $x_1 = 1$ then swap($y_1, y_2$)
  if $x_2 = 1$ then swap($y_3, y_4$)
end.
if \( x_3 = 1 \) then swap\( (y_1, y_3) \) 
if \( x_4 = 1 \) then swap\( (y_2, y_4) \) 
if \( x_5 = 1 \) then swap\( (y_1, y_5) \) 
end.

### B. Using the cube distance

The 5th vertex could be connected to any vertex of the 2-cube. Fig. 7 shows the possible distances when the symbol 5 is connected to others. It is clear that the distance in the case of \((2,5)\) is not acceptable since it is equal to 3, which is not in the form of a geometric series of ratio 2. So the symbol 5 could be linked to 1 with a distance of 2\(^2\) or to 3 with a distance of 2\(^1\) or to 4 with a distance of 2\(^0\). Since \(k\) is defined in (2), the symbol 5 belongs to the 3-cube and not to 2-cube and here the geometric series ends at the value of 2\(^{(3-1)}\), which is 2\(^2\). Thus the symbol 5 should be connected to the symbol 1 as shown in Fig. 7.

Fig. 7. Distance based construction of a cube

It can be seen that we could reach the same results by either using the graph theory or the cube distance approach. Our results prove the reliability of our cube-distance theory without changing the properties of the cube graph.

### V. The Lower Bound of \( \delta \)

Taking into account the number of vertices in a cube as the number of permutation symbols, we can consider the total number of edges as the total number of transpositions or swaps that can be achieved with our distance-preserving mappings and more precisely distance-reducing mappings.

**Proposition 1** The lower bound of \( \delta \) in distance-preserving mappings for a cube construction is \( \delta_{\text{min}} = 2^k - 1(2 - k) \).

**Proof:** From previous definitions and results, the maximum number of swaps for an optimum mapping with cube construction is in fact the number of edges. Thus maximum number of output bits of a convolutional code should be equal to the total number of edges in the cube, which means that \( n_{\text{max}} = k \times 2^k - 1 \).

From previous definitions of the distance-preserving mappings we have defined \( \delta \) as \( \delta = M - n \) and thus \( \delta_{\text{min}} = M - n_{\text{max}} \). So the lower bound is \( \delta_{\text{min}} = 2^k - k \times 2^{k-1} \), which leads to \( \delta_{\text{min}} = 2^k - 1(2 - k) \).

Although the value of \( \delta \) is easy to be calculated, the corresponding value to an optimum mapping stays a problem especially when it comes to a distance-reducing mapping. The simplicity of the cube graph construction has solved this problem as presented in the proposition. Table I shows the values of \( \delta \) for different values of \( M \) and their corresponding types of mappings.

### VI. Conclusion

We have designed a very simple and reliable construction based on the graph theory. The new concept of the cube distance introduced in this paper has played a major role in the construction of distance-preserving mappings, without changing the theoretical properties of the cube graph.

We have reached the upper bound on the sum of the distances for all values of \( M = 2^n \). Although cube and multilevel constructions have reached the same results, the cube graph construction is considered the simplest and easiest construction to be used and to help understand the concept of the mapping technique. To our knowledge this is the only construction that has emphasized better and explained in a very simple manner the mechanism of a distance-reducing mapping.

The complete bipartite property of the cube plays a role in the optimality since the symbols are split into two different sets which help avoid repetitions that might cause loss on the sum of the distances.

Table II compares the sum on the distance achieved by our \( k \)-cube graph construction to other published constructions [15]. Table III presents few designed optimum permutation trellis codes using the cube graph construction.

Further work on the derivation of the upper bound on the sum of the distances based on the cube construction as well as the generalization of the mapping algorithm will be presented in our journal paper.

### References

### TABLE II
Comparison of distances for different conserving mappings

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>708</td>
<td>708</td>
<td>708</td>
<td>708</td>
<td>708</td>
</tr>
<tr>
<td>5</td>
<td>498752</td>
<td>3616</td>
<td>3712</td>
<td>312</td>
<td>3712</td>
</tr>
<tr>
<td>8</td>
<td>498752</td>
<td>358840</td>
<td>367744</td>
<td>48572</td>
<td>48572</td>
</tr>
<tr>
<td>12</td>
<td>18459344</td>
<td>-</td>
<td>13887080</td>
<td>180355072</td>
<td>180355072</td>
</tr>
<tr>
<td>16</td>
<td>6442450944</td>
<td>-</td>
<td>4700630976</td>
<td>6442450944</td>
<td>6442450944</td>
</tr>
</tbody>
</table>

### TABLE III
Some conserved permutation trellis codes

<table>
<thead>
<tr>
<th>$Q(M, n, \delta)$</th>
<th>$E_{\text{max}}$</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4, 1, 0)</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>(4, 1, 0)</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>(5, 5, 0)</td>
<td>4090</td>
<td></td>
</tr>
<tr>
<td>(4, 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4, 1)</td>
<td>708</td>
<td></td>
</tr>
<tr>
<td>(4, 1)</td>
<td>708</td>
<td></td>
</tr>
<tr>
<td>(6, 7, -1)</td>
<td>89192</td>
<td></td>
</tr>
<tr>
<td>(7, 7, 0)</td>
<td>98294</td>
<td></td>
</tr>
<tr>
<td>(8, 8, 0)</td>
<td>458752</td>
<td></td>
</tr>
<tr>
<td>(8, 8, 0)</td>
<td>458752</td>
<td></td>
</tr>
</tbody>
</table>

---


