Commun. Combin., Cryptogr. & Computer Sci., 2 (2022), 195–200

ISSN:2783-5456

Communications in Combinatorics, Cryptography & Computer Science

Journal Homepage: http://cccs.sgh.ac.ir

Graph Metric Codes of Minimum Distance 3

Mehri Oghbaei[a,*], Zahra Hadizadeh[a], Amirreza Akbari[a], Javad B. Ebrahimi[a,b]

[a]Sharif University of Technology, Mathematics Department, Tehran, Iran.

[b]IPM, Institute for Research in Fundamental Sciences, Tehran, Iran.

Abstract

In this work, we introduce the notion of graph metric codes. In these codes, the code alphabet is the set of the vertices of a fixed graph G, the codewords are vectors over $V(G)$ and the distance of two vectors is defined as the summation of the graph distances of the corresponding entries. Then, we present a code construction with the minimum distance of 3. To the best of our knowledge, for the particular case of $G = K_2$, our result gives the best known lower bound for the size of minimum distance 3 codes, when the length of the vectors is large enough. Finally, we present an application of these codes in the context of the innate degree of freedom of Kekule structures in organic chemistry.

Keywords: Graph metric codes, Error correcting codes, Lee metric codes, Graph, Maximum fractional forcing number.

2020 MSC: 94A24, 05C90.

1. Introduction

Points of bounded pairwise distance in a space are important objects both from theoretical and applicable views. When the space is a product metric space of a discrete set equipped with Hamming metric, the set of mentioned points forms an object called a "code". In other words, a code $\mathcal{C}$ of length n over a metric space (Σ, d) is a collection of n-length sequences, called "codewords". Codes are useful tools in various applications such as data transmission and data storage. However, for different needs, we may need to study different distance measures.

In this paper, we consider the graph distance as a metric and define "graph metric codes". These codes generalize all the other known codes such as Hamming, Lee, and L_1 codes. Then, we consider the problem of maximum size codes of a given length and minimum distance 3. We explain a method to construct such codes. As an application of distance 3 graph metric codes, we describe a general method to obtain an upper bound on the maximum fractional forcing number of the graphs.

*Corresponding author

Email address: m.oghbaei95@student.sharif.edu (Mehri Oghbaei)

Received: November 3, 2022 Revised: November 10, 2022 Accepted: November 21, 2022

2. Preliminaries

Let (Σ, dist) be a discrete metric space where Σ is a set of elements and dist is a metric. A length n code $\mathcal{C}$ is a non-empty subset Σ^n equipped with the distance function which we also denote by dist and is defined as $\text{dist}(x, y) = \sum_{i=1}^{n} \text{dist}(x_i, y_i)$ where $x, y \in \Sigma^n$ and x_i, y_is are the coordinates of x, y, respectively. Elements of $\mathcal{C}$ are called codewords. The minimum distance of a code is the minimum of $\text{dist}(x, y)$ over all distinct codewords x and y and we denote it by d.

When Σ is the set V of the vertices of a graph G and $\text{dist}(x, y)$ is defined as the length of the shortest path from the vertex x to the vertex y, then dist is called the metric distance associated with the G. The resulting codes are also called "graph metric codes".

When G is isomorphic the complete graph K_n, the cycle graph C_n, or the path graph P_n, the metric distance is called "Hamming", "Lee" or "L_1-norm" distances, respectively. These codes are widely studied and in this paper, we introduce the notion of graph metric which simultaneously generalizes all of these three metrics.

Let $A_H(q, n, d_H)(A_G(q, n, d_G))$ be the largest possible size of code $\mathcal{C}$ over the alphabet $\Sigma(V(G))$ of size q, with the length n and the minimum Hamming distance d_H (minimum graph distance d_G). When $q = 2$, we show $A_H(2, n, d_H)$ with $A_H(n, d_H)$.

3. Main results

In this section, we aim to present a method to construct graph metric codes of minimum distance 3. As we will see in the next section, such codes play an important role in the study of a parameter called "maximum fractional forcing number" of graphs. We first explain our method for the case of the graph K_2 and then we present the general code construction for arbitrary graphs.

3.1. Binary Hamming Metric Codes of Distance 3

In this part, we present a lower bound for $A_H(n, 3)$. Later in the next section, we will generalize this method for the case of graph metric codes for a particular family of graphs.
Let $\mathcal{L}$ be a code of fixed L distinct binary codewords of size n. First, to find a lower bound for $A_H(n, 3)$, we try to find an upper bound for n according to L. So, we show how to increase the distance of L codes which have $d_H(L) \geqslant 1$ by increasing their size. Then we can obtain the upper bound for n according to L. According to this upper bound for n, we can find a lower bound for $A_H(n, 3)$.

Definition 3.1 (Distance of binary matrix). Let $A_{m\times n} = [a_{i,j}] \in \{0, 1\}^{m\times n}$ be a binary matrix. We define the function d for each pair $(a_{i,*}, a_{j,*})$ of rows in matrix A as follows

$$d(a_{i,*}, a_{j,*}) = \sum_{k=0}^{n-1} |a_{i,k} - a_{j,k}| \tag{3.1}$$

and so we will define the distance of a binary matrix as below

$$d(A) = \min_{0\leqslant i<j\leqslant m-1} d(a_{i,*}, a_{j,*}) \tag{3.2}$$

Lemma 3.2. Let $A_{m\times n}$ be a binary matrix with $d(A) \geqslant 1$, and $A'_{m\times(n+1)}$ be the output matrix of implementing Algorithm 1 for input A, then we have $d(A') \geqslant 2$.

Proof. Consider b_i's from Algorithm 1, and define the column b as $b = [b_0, b_1, \cdots, b_{m-1}]^t$.
From Algorithm 1, we suppose that $A' = \left[\begin{array}{c|c} A & b \end{array}\right]$ and according to Definition 3.1, it can be concluded that $d(A') \geqslant d(A)$. It is clear that if we have $d(A) \geqslant 2$, then we can conclude $d(A') \geqslant 2$. So, we can assume

Input: Binary matrix $A_{m\times n} = [a_{i,j}]$ with $d(A) \geqslant 1$
Output: Binary matrix $A'_{m\times(n+1)} = [a'_{i,j}]$ with $d(A') \geqslant 2$
Initialize A' as a matrix of m rows and $n+1$ columns;
for $i < m$ do
    $b_i = 0$;
end
for $j < n$ do
    $b_i = b_i + a_{i,j}$;
    $a'_{i,j} = a_{i,j}$;
end
$a'_{i,n} = b_i \mod 2$;
return Matrix A';

Algorithm 1: min2discode

that $d(A) = 1$. In this case, if $d(A') < 2$ then, there exists at least a pair of (x, y) that $d(a'_{x,*}, a'_{y,*}) < 2$; As mentioned, we have

$$a'_{x,*} = \left[\begin{array}{c|c} a_{x,*} & b_x \end{array}\right],\ a'_{y,*} = \left[\begin{array}{c|c} a_{y,*} & b_y \end{array}\right] \tag{3.3}$$

On the other hand, we assumed that $d(A) \geqslant 1$. Thus we can conclude that

$$1 \geqslant d(a'_{x,*}, a'_{y,*}) \geqslant d(a_{x,*}, a_{y,*}) \geqslant 1 \tag{3.4}$$

According to Equation 3.3, it can be obtained that

$$d(a_{x,*}, a_{y,*}) = d(a'_{x,*}, a'_{y,*}) = 1 \Rightarrow b_x = b_y \tag{3.5}$$

The assumption $d(a_{x,*}, a_{y,*}) = 1$ says that parities of the bits 1 in the rows $a_{x,*}$ and $a_{y,*}$ are not equal. According to Algorithm 1, b_i is the parity of the numbers of bits 1 in the i-th row, and the equation $b_x = b_y$ contradicts the fact that parities of the bits 1 in the rows $a_{x,*}$ and $a_{y,*}$ are different. □

Input: Binary matrix $A_{m\times n} = [a_{i,j}]$ with $d(A) \geqslant 2$
Output: Binary matrix $A''_{m\times(n+\lceil\log n\rceil)} = [a''_{i,j}]$ with $d(A'') \geqslant 3$
Initialize matrix A'' as a copy of A;
Initialize columns $P_0,\ P_1,\ \ldots, P_{\lceil\log n\rceil-1}$ of m elements;
for $i < \lceil\log n\rceil$ do
    for $j < m$ do
        $P_i[j] = 0$;
        for $k < n$ do
            if $\frac{k}{2^i} \mod 2 = 1$ then
                $P_i[j] += a_{i,k} \mod 2$;
            end
        end
        $P_i[j] = P_i[j] \mod 2$;
    end
    Append column P_i to A'';
end
return Matrix A'';

Algorithm 2: min3discode

Lemma 3.3. Let $A_{m\times n}$ be a binary matrix with $d(A) \geqslant 2$, and $A'_{m\times(n+\lceil\log n\rceil)}$ be the output matrix of implementing Algorithm 2 for input A. Then we have $d(A') \geqslant 3$.

Proof. According to Algorithm 2, we have that

$$A'' = \left[\ A \mid P_0 \mid P_1 \mid .... \mid P_{\lceil\log n\rceil-1}\ \right] \tag{3.6}$$

Suppose that $a''_{x,*}$ and $a''_{y,*}$ are two arbitrary rows of matrix A''. We know that

$$a''_{x,*} = \left[\ a_{x,*} \mid P_0[x] \mid P_1[x] \mid \dots \mid P_{\lceil\log n\rceil-1}[x]\ \right] \tag{3.7}$$

$$a''_{y,*} = \left[\ a_{y,*} \mid P_0[y] \mid P_1[y] \mid \dots \mid P_{\lceil\log n\rceil-1}[y]\ \right] \tag{3.8}$$

If the inequality $d(a''_{x,*}, a''_{y,*}) > 2$ holds, the lemma is proved; Also we have $d(A) \geqslant 2$. So, we claim that

$$d(a''_{x,*}, a''_{y,*}) = d(a_{x,*}, a_{y,*}) = 2 \tag{3.9}$$

and the whole difference of two rows $a''_{x,*}$ and $a''_{y,*}$ are in the first n elements. Let these two differences occur in s-th and t-th elements. In other words, we have

$$a_{x,s} = a''_{x,s} \neq a''_{y,s} = a_{y,s} \tag{3.10}$$

$$a_{x,t} = a''_{x,t} \neq a''_{y,t} = a_{y,t} \tag{3.11}$$

As s and t are two distinct numbers, there exists at least one number k that binary representation of s and t are different in k-th bit. We want to show that $P_k[x] \neq P_k[y]$.
Without loss of generality, we can consider that k-th bit of binary representation of t and s are 0 and 1, respectively. According to Algorithm 2,

$$P_k[x] \quad = \sum_{r=0}^{n-1} a_{x,r} \times \left(\tfrac{r}{2^k} \mod 2\right) \tag{3.12}$$

$$P_k[y] \quad = \sum_{r=0}^{n-1} a_{y,r} \times \left(\tfrac{r}{2^k} \mod 2\right) \tag{3.13}$$

As we know $d(a_{x,*}, a_{y,*}) = 2$, and these two differences only occur in two s-th and t-th elements. So, we can suppose that

$$P_k[x] - P_k[y] \quad = (a_{x,s} - a_{y,s})\left(\tfrac{s}{2^k} \mod 2\right) \tag{3.14}$$

$$+(a_{x,t} - a_{y,t})\left(\tfrac{t}{2^k} \mod 2\right) \tag{3.15}$$

As we assumed that k-th bit of binary representation of s is zero, then it holds $\frac{t}{2^k} \mod 2 = 0$. It shows that

$$P_k[x] - P_k[y] = (a_{x,s} - a_{y,s})\left(\frac{s}{2^k} \mod 2\right) \tag{3.16}$$

On the other hand, we know that $(\frac{s}{2^k} \mod 2) = 1$ and $a_{x,s} \neq a_{y,s}$, so it shows that $P_k[x] \neq P_k[y]$ □

Theorem 3.4. Let $A_H(n,3)$ be denoted to the largest possible size of code $\mathcal{C}$ over the field $\{0,1\}$, with length n and minimum Hamming distance 3. Then the following inequality holds:

$$\frac{2^{n-1}}{n+1-\log n} \leqslant A_H(n,3) \tag{3.17}$$

Proof. Let $M_{L\times\lceil\log L\rceil} = [m_{i,j}] \in \{0,1\}^{L\times\lceil\log L\rceil}$ be a matrix that $m_{i,j}$ equals to the j-th bit of the binary representation of number i. So, every two rows of this matrix have at least distance 1. Then we have: $d(M) \geqslant 1$
First of all, we run Algorithm 1 with input M and achieve matrix $M'_{L\times(\lceil\log L\rceil+1)}$ which leads that: $d(M') \geqslant 2$
After that, it is only needed to run Algorithm 2 with input M' and get the matrix $M''_{L\times(\lceil\log L\rceil+1+\lceil\log(\lceil\log L\rceil+1)\rceil)}$ where we have $d(M'') \geqslant 3$.
It is clear to see that $L = \frac{2^{n-2}}{n+1-\log n}$ satisfies the inequality $\lceil\log L\rceil + 1 + \lceil\log(\lceil\log L\rceil + 1)\rceil \leqslant n$. □

3.2. Graph Metric Codes Of Distance 3

In this part, we use the same technique as the previous section and present a lower bound for $A_G(n, 3)$ where G is a complete regular p-partite graph. First of all, since G is a regular p-partite graph, there exists a natural number m that $G \simeq K_{m,m,\dots,m}$.
Assume that we want to present graph code $\mathcal{L}$ which contains L codewords with $d_G(\mathcal{L}) \geqslant 3$. First, consider codewords $c_1, c_2, \dots, c_L$ of size $k = \lceil \log_{pm} L \rceil$ that c_i is the base-pm representation of number i. Since each pair of these codewords are the base-pm representation of distinct numbers, for each $1 \leqslant i < j \leqslant L$: $d_G(c_i, c_j) \geqslant 1$
Now we want to construct graph code $\mathcal{L}' = \{c'_1, c'_2, \dots, c'_L\}$ as follows:

$$c'_i = c_i \| g_i \text{ and } g_i = (c_i[0] + \cdots + c_i[k-1]) \mod pm \tag{3.18}$$

Which $\|$ means adding the bit g_i to the end of row c_i.
Now we want to construct graph code $\mathcal{L}''$ with $d_G(\mathcal{L}'') \geqslant 3$. Let $k' = \lceil \log_p(k+1) \rceil$ and for every natural number b, we define function f_b as follows:$\forall x, y \in \mathbb{N} \cup \{0\} : f_b(x, y) = a_y$ where $(a_{k'} \dots a_y a_{y-1} \dots a_1 a_0)_b$ is the base-b representation of number x. For each $i \in \{1, \dots, L\}$, we define s_i as follows:

$\forall j \in \{0, \dots, k'\} : s_i[j] = [\sum_{t=0}^{k} c'_i[t] \times f_p(t+1, j)] \mod p$. It is clear that all digits of s_i's are less than p. So, we can consider each s_i is the base-p representation of a number. Let $r_1, \dots, r_L$ be the base-pm representation corresponding to $s_1, \dots, s_L$. Now we construct graph code $\mathcal{L}'' = \{c''_1, c''_2, \dots, c''_L\}$ where we have $c''_i = c'_i \| r_i$ for each $i \in \{1, \dots, L\}$.

Theorem 3.5. For the labeling of the vertices such that labels of the nodes within each part are congruent modulo p, $d_G(\mathcal{L}'') \geqslant 3$.

Proof. If $d_G(c_i, c_j) = 2$ and their difference is only one digit. It is noticeable that $c'_i[k] \neq c'_j[k]$, then $d_G(c''_i, c''_j) \geqslant 3$. If $d_G(c_i, c_j) = 2$ and their difference are in indices w_1 and w_2. It is clear to see that if $c'_i[k] \neq c'_j[k]$, then we have $d_G(c''_i, c''_j) \geqslant 3$. So, assume that $c'_i[k] = c'_j[k]$ or in other words,

$$c_i[w_1] + c_j[w_2] \overset{pm}{\equiv} c_j[w_1] + c_j[w_2] \tag{3.19}$$

Since $w_1 \neq w_2$, there is an integer x where $f_p(w_1, x) \neq f_p(w_2, x)$. Now we want to prove that $s_i[x] \neq s_j[x]$. For the sake of contradiction, assume $s_i[x] = s_j[x]$

$$0 \overset{p}{\equiv} s_i[x] - s_j[x]$$

$$(c_i[w_1] - c_j[w_1]) f_p(w_1, x) \overset{p}{\equiv} (c_j[w_2] - c_i[w_2]) f_p(w_2, x)$$

According to Equation 3.19 and $f_p(w_1, x) \overset{p}{\not\equiv} f_p(w_2, x)$, we can conclude that $c_i[w_1] - c_j[w_1] \overset{p}{\equiv} 0 \overset{p}{\equiv} c_j[w_2] - c_i[w_2]$. It shows that both $(c_i[w_1], c_j[w_1])$ and $(c_i[w_2], c_j[w_2])$ are in the same part of graph. The distance of each two would be two so the total distance would be at least four. If $d_G(c_i, c_j) = 1$ holds, it can be concluded that $c'_i[k] \neq c'_j[k]$; so $d_G(c'_i, c'_j) \geqslant 2$, and it can be obtained $d_G(c''_i, c''_j) \geqslant 3$ with some simple algebraic calculations. □

Exactly the same as Section 3.1, this can be obtained that $A_G(n, 3) \geqslant \frac{(pm)^{n-1}}{n}$.

4. Application

Beyond the fact that graph metric codes generalize other types of codes, they are useful in other contexts. As an example, in [2], Ebrahimi and Ghanbari proved that if G is a connected graph on q vertices and $S \subseteq V(G^n)$ such that no two vertices in S have a common neighbour, then $q^n - |S|$ is an upper bound of the fractional forcing number of G^n (See [2] for the precise definition of maximum forcing number problem).

This parameter is closely related to the notion of innate degree of freedom of Kekul´e structures in the context of organic chemistry. (See [1]).

In the language of coding theory, S is in fact a length n code of distance 3 over G metric, and in order to obtain the best upper bound of the fractional forcing number, we must find the best code of minimum graph distance 3. Results of Section 3.2 is particularly useful to obtain such bounds.

5. Conclusion

In this paper, we first introduced graph metric codes which generalize codes over several distance measures. Then we provide a code construction for distance 3. Our construction, provides the best known lower bound even for the special case of Hamming metric. We present an application of distance 3 graph metric codes in the problem of maximum fractional forcing number.

References

[1] Cai, Jinzhuan and Zhang, Heping, Global forcing number of some chemical graphs, Match-Communications in Mathematical and Computer Chemistry, 2012. 4

[2] Ebrahimi, Javad B and Ghanbari, Babak, Fractional forcing number of graphs, 2020. 4