

EXPLORING PERFECT BINARY TREES WITH RELATION TO THE HK-PROPERTY

ATISHAYA MAHARJAN AND MAHSA N. SHIRAZI

ABSTRACT. A perfect binary tree is a full binary tree in which all leaves have the same depth. A set of cocliques of size k (k -coclique) in a graph, containing a fixed vertex v is called a star, and is denoted by $\mathcal{I}_G^n(v)$. We study the size of stars for different vertices in a perfect binary tree. This structure is useful in studying the Erdős-Ko-Rado theorem. Hurbert and Kumar conjectured that in trees, the largest stars are on the leaves. The conjecture was shown to be false independently by Baber, Borg, and Feghali, Johnson and Thomas. However, in some classes of trees such as caterpillars, the conjecture holds true. In this paper, we study the perfect binary trees through the lens of star centers and seek to answer if the HK-property holds for perfect binary trees. We give an formula and an inductive proof for the independence number of a perfect binary tree and show that the HK-property holds for perfect binary trees. We then show that the proof structure supports any perfect m -nary tree, where m is a natural number. Finally, we use an algorithm by Niskanen and R. J. to generate all cocliques of a perfect binary tree of depth d and compare the number of k -cocliques containing a vertex v and a leaf l to see, numerically, if the HK-property holds for perfect binary trees.

1. INTRODUCTION AND PRELIMINARIES

For a given graph $G = (V, E)$, $V(G)$ and $E(G)$ denotes the vertex sets and edge sets of the graph G . For an arbitrary vertex, $v \in V(G)$, all vertices adjacent to v are called the neighbours of v and the set of neighbours of v is denoted by $N_G(v)$. The degree of a vertex $v \in V(G)$ is the cardinality of the set of neighbours of v , and is denoted by $\deg_G(v)$.

For $n \in \mathbb{Z}^+$ such that $0 \leq n \leq |V(G)|$, a path of length n in G is a sequence of distinct vertices v_1, v_2, \dots, v_n such that v_i is adjacent to v_{i+1} for $1 \leq i \leq n-1$ and it is called a uv -path. A cycle is an extension of a path such that the last vertex is connected to the first vertex, i.e for a path of length n , $v_n v_1 \in E(G)$. As such, the length of the cycle is $n+1$.

A connected graph is a graph if for all $u, v \in V(G)$, there exists a uv -path. A coclique is a set of vertices such that no two vertices in the set are adjacent to each other. We denote a family of independent sets of a graph G by \mathcal{I}_G . A family of coclique of size n in a graph G is denoted by \mathcal{I}_G^n . For $v \in V(G)$, the family of independent sets, $\mathcal{I}_G^n(v) := \{A \in \mathcal{I}_G^n : v \in A\}$ is called a star of \mathcal{I}_G^n and v is called its center.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF MANITOBA, WINNIPEG, R3T 2N2, CANADA
E-mail addresses: maharjaa@myumanitoba.ca, mahsa.nasrollahi@gmail.com.

Date: June 18, 2024.

Key words and phrases. EKR, HK-property, Perfect binary trees, Independence number.

A tree is a connected graph with no cycles, it is denoted by T . For a vertex $v \in V(T)$, if $\deg_T(v) = 1$, it is called a leaf. A vertex that is not a leaf is called an interior vertex. The vertex at the top which branches 2 children is called the root vertex and is denoted by $Root$. The depth of a vertex is defined as the cardinality of the set of vertices of the path from the root vertex to it. The parent of a vertex v is the vertex connected to v on the path to $Root$. A child of a vertex v is a vertex of which v is the parent. We study a more particular class of trees called binary trees, denoted by T_B , where each interior vertex v has exactly 2 children and all leaves have the same depth. Further, a perfect binary tree is a binary tree in which every vertex $v \in V(T)$ has either 0 or 2 children. A perfect binary tree is denoted by T_{PB} , however in this paper we will simply drop the subscript and denote it as T for clarity. A level n of a perfect binary tree is a set of vertices such that all vertices in the set have a depth of n .

The star centers of a graph are interesting because they relate to the EKR theorem.

The Erdős-Ko-Rado (EKR) theorem limits the number of sets in a family of sets that can be pairwise intersecting. The theorem states that for a family of k -sets of a ground set of size n , the maximum number of sets that can be pairwise intersecting is $\binom{n-1}{k-1}$. This theorem has wide applications in combinatorics, graph theory, probability, and other areas of statistics and mathematics.

Studying the EKR theorem, [5] Hulbert and Kamat tried to narrow it down to a more reduced class of graphs called the k -EKR graphs. A graph G is said to be k -EKR if for any family of cliques $\mathcal{F} \subset \mathcal{I}_G^k$ satisfying $A \cap B \neq \emptyset$ for every $A, B \in \mathcal{F}$, there is a vertex $x \in V(G)$ such that $|\mathcal{F}| \leq \mathcal{I}_G^k(x)$. They then conjectured the following:

Conjecture 1.1 (k -EKR Conjecture). *Let G be a graph, and let $\mu(G)$ be the size of its smallest maximal clique. Then G is k -EKR for every $1 \leq k \leq \frac{\mu(G)}{2}$.*

However, this conjecture is hard to understand and prove, so they narrowed the class of graphs to be trees and gave the following conjecture:

Conjecture 1.2 (HK-Property). *For any $k \geq 1$ and any tree T , there exists a leaf l of T such that $|\mathcal{I}_T^k(v)| \leq |\mathcal{I}_T^k(l)|$ for each $v \in V(T)$.*

The HK-property holds true for $k \leq 4$, but was proven false independently in [1, 2, 3]. The counterexample that they arrived at is a type of graph that is defined as a class of trees called *lobster* [4]. A tree C is called a *caterpillar* if removing the leaves and incident edges produces a path graph P , called the spine. A tree L is called a *lobster* if removing the leaves and incident edges produces a caterpillar C .

This is interesting for us as *lobster* graphs resemble a binary tree.

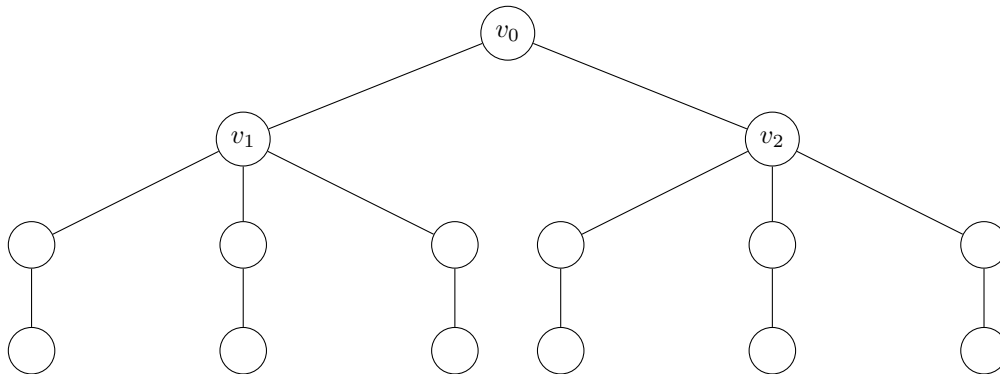
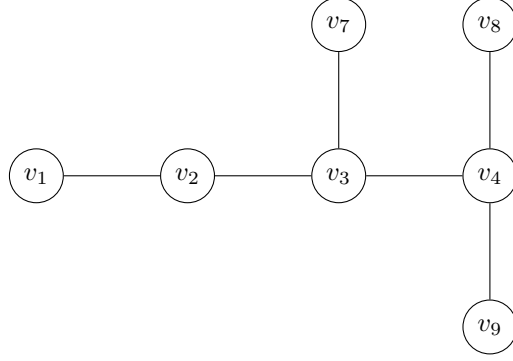


FIGURE 1. The largest k -star for $k \geq 5$ is centered at v_0 (A lobster)

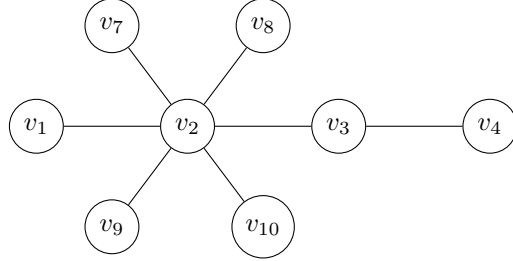
They figured out that the lobsters, while not completely obeying the HK-property, almost obey the HK-property by either having the largest stars centered around the leaves or at the root of the tree.

2. SOME CLASSES OF GRAPHS THAT ARE HK AND A COUNTEREXAMPLE

In the paper by Estrugo and Passtine [4], they gave a couple of classes of graphs that have the HK-property. The classes of graphs that they gave were the *caterpillars* and the *spiders*. The definition of *caterpillars* is given above. The *spiders* are trees that only have 1 vertex with degree greater than 2.



(A) A caterpillar



(B) A Spider

3. PERFECT BINARY TREES ARE HK

Definition 3.1 (Maximum Coclique). *We denote the maximum coclique of a perfect binary tree of depth d by \mathcal{I}_d .*

Conjecture 3.2 (HK property for a perfect binary tree). *For any given perfect binary tree T , the maximum number of cocliques lie in the leaves. The number of cocliques is denoted by k . $\alpha(T)$ denotes the independence number of a tree T .*

Lemma 3.3. *For a given perfect binary tree T with depth d and maximum number of cocliques possible, i.e. $k = \alpha(T)$ we have:*

$$\alpha(T) = \begin{cases} 2^{d-1} + 2^{d-3} + \dots + 1 & \text{for odd } d \\ 2^{d-1} + 2^{d-3} + \dots + 2 & \text{for even } d \end{cases}$$

Or, in summation notation:

$$\alpha(T) = \sum_{i=1}^{\lceil \frac{d}{2} \rceil} 2^{d-2i+1}$$

And for odd d , $\text{Root} \in \mathcal{I}_d$, otherwise $\text{Root} \notin \mathcal{I}_d$.

Proof for 3.3.

We shall proceed by inducing on d .

We will have two cases, one for d being odd and another for d being even.

Case 1: d is odd

For our base case, consider the trivial case of $d = 1$. Here, $\alpha(T) = 1$. Hence, the base case holds.

Now, say that the statement holds for all d . We now have to show that it holds for $d + 1$. That is, we need to show that it holds for a given perfect binary tree T of depth $d + 1$. Note that if d is odd, then $d + 1$ is even. Hence, we expect that:

$$\alpha(T) = 2^d + 2^{d-2} + \dots + 2$$

Consider these 2 cases:

Case 1: $Root \in \mathcal{I}_{d+1}$

Let the left and right child of the root be v_0 and v_1 respectively. Let C_1, C_2, C_3, C_4 be the four perfect binary tree components generated by $T \setminus \{Root, v_0, v_1\}$.

If the root $\in \mathcal{I}_{d+1}$, then $v_0 \notin \mathcal{I}_d$ and $v_1 \notin \mathcal{I}_d$. This means that the remaining vertices of \mathcal{I}_d is in one of the 4 components. Since the current depth from the *Root* is $d + 1$, then $T \setminus \{Root, v_0, v_1\}$ will have depth of $d + 1 - 2 = d - 1$. Since d is odd, then $d + 1$ is even which implies that $d - 1$ is also even.

By symmetry, it is enough to consider one of the 4 components' maximum coclique for our calculations. Since the 4 components are disjoint, then we can add their independence numbers together along with the *Root* and obtain the following:

$$\alpha(T_{d+1}) = 4(\alpha(T_{d-1})) + 1$$

Then, from our induction hypothesis, we get that:

$$\begin{aligned} \alpha(T_{d+1}) &= 4 \underbrace{(2^{d-2} + 2^{d-4} + \dots + 2)}_{\frac{d-1}{2} \text{ terms}} + 1 \\ &= 4 \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i} + 1 \end{aligned} \tag{1}$$

Case 2: $Root \notin \mathcal{I}_{d+1}$

If $Root \notin \mathcal{I}_d$, then the remaining elements of \mathcal{I}_d are from $T \setminus \{Root\}$.

Let $T' = T \setminus \{Root\}$. Then T' is a forest of 2 disjoint and distinct components. Let C_1 and C_2 be the 2 components of T' . Since T was a perfect binary tree of depth $d + 1$, then C_1 and C_2 are

also perfect binary trees of depth:

$$\begin{aligned} &= (d+1) - 1 \\ &= d \end{aligned}$$

Since d is odd, and C_1 and C_2 are disjoint and distinct perfect binary trees,

$$\alpha(T') = \alpha(C_1) + \alpha(C_1)$$

By symmetry we get that,

$$\alpha(T') = 2\alpha(C_1)$$

Since $Root \notin \mathcal{I}_d$,

$$\alpha(T) = \alpha(T') = 2\alpha(C_1)$$

Then, by our induction hypothesis,

$$\begin{aligned} \alpha(T) &= \underbrace{2(2^{d-1} + 2^{d-3} + \dots + 1)}_{\lceil \frac{d}{2} \rceil \text{ terms}} \\ &= 2 \left(\sum_{i=1}^{\lceil \frac{d}{2} \rceil} 2^{d-(2i-1)} \right) \end{aligned}$$

Note that $\lceil \frac{d}{2} \rceil = \frac{d+1}{2}$, since d is odd, then,

$$\begin{aligned} \alpha(T) &= 2 \left(\sum_{i=1}^{\lceil \frac{d}{2} \rceil} 2^{d-(2i-1)} \right) \\ &= 2 \left(\sum_{i=1}^{\frac{d+1}{2}} 2^{d-(2i-1)} \right) \end{aligned} \tag{2}$$

Since $\alpha(T)$ is the maximum coclique,

$$\alpha(T) = \max(1, 2)$$

Remember that our objective is to show that $\alpha(T) = (2)$ as this aligns with our inductive hypothesis.

From (2), we can simplify it as the following:

$$\begin{aligned}
\alpha(T) &= 2 \left(\sum_{i=1}^{\frac{d+1}{2}} 2^{d-(2i-1)} \right) \\
&= 2 \left(2^{d-2(\frac{d+1}{2})+1} + \sum_{i=1}^{\frac{d-1}{2}} 2^{d-(2i-1)} \right) \\
&= 2 \left(2^{d-d-1+1} + \sum_{i=1}^{\frac{d-1}{2}} 2^{d-(2i-1)} \right) \\
&= 2 \left(2^0 + \sum_{i=1}^{\frac{d-1}{2}} 2^{d-(2i-1)} \right) \\
&= 2 \left(1 + \sum_{i=1}^{\frac{d-1}{2}} 2^{d-(2i-1)} \right) \\
&= 2 \left(1 + \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i+1} \right) \\
&= 2 \left(1 + 2 \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i} \right) \\
&= 2 + 4 \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i} \\
&= 4 \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i} + 2 \\
&> 4 \sum_{i=1}^{\frac{d-1}{2}} 2^{d-2i} + 1 \\
&= (1)
\end{aligned}$$

Which implies that,

$$\alpha(T) = \max(1, 2) = (2) \text{ as required.}$$

Case 2: d is even

We will proceed similarly to the odd case.

For our base case, consider the case of $d = 2$. Let v_0 and v_1 be the leaves of the perfect binary tree of depth 2. We claim that $\mathcal{I}_2 = \{v_0, v_1\}$. This implies that $Root \notin \mathcal{I}_2$.

We will prove the claim by using contradiction.

For contradiction, say that $Root \in \mathcal{I}_2$, then $v_0 \notin \mathcal{I}_2$ and $v_1 \notin \mathcal{I}_2$ since both v_0 and v_1 are neighbours of $Root$. This then implies that $\alpha(T) = 1$.

However, the set of leaves $\{v_0, v_1\}$ has cardinality $2 > 1 = \alpha(T)$ which is a contradiction.

Hence, for a perfect binary tree T of depth 2, $\alpha(T) = 2$. Thus, our base case holds.

Now, say that the statement holds for all d . We now have to show that it holds for $d + 1$. That is, we need to show that it holds for a given perfect binary tree T of depth $d + 1$. Note that if d is even, then $d + 1$ is odd. Hence, we expect that:

$$\alpha(T) = 2^d + 2^{d-2} + \dots + 1$$

which follows from our previous case. □

Lemma 3.4. *Following from the previous lemma 3.3, we claim that if $|\mathcal{I}_d| = \alpha(T)$, then \mathcal{I}_d is unique.*

Proof for 3.4.

Let \mathcal{I}_d be the maximum coclique, we will first construct a set of vertices that build \mathcal{I}_d and then show that no other vertex can be added to the set nor removed which will satisfy uniqueness.

We shall proceed by cases.

Case 1: d is odd

From 3.3, we know that if d is odd, then $Root \in \mathcal{I}_d$.

Let v_0 and v_1 be the children of $Root$,

If $Root \in \mathcal{I}_d$, then we know that $v_0 \notin \mathcal{I}_d$ and $v_1 \notin \mathcal{I}_d$.

Let T_1, T_2, T_3, T_4 be the disjoint components obtained from $T \setminus \{v_0, v_1, Root\}$.

Note that T_1, T_2, T_3 , and T_4 are all perfect binary trees of depth $d - 2$.

Since d was odd, then $d - 2$ is also odd. □

APPENDIX A. COCLIQUES ALGORITHM AND ANALYSIS

To validate our conjecture, we present a simple algorithm to generate all cocliques of a perfect binary tree of cardinality k . We then compare the number of cocliques containing a vertex v and a leaf l to see if the HK-property holds for perfect binary trees.

To begin with, we present the following algorithm to generate a perfect binary tree of depth n :

Algorithm 1: Perfect Binary Tree Generator

Data: $n \geq 0$, where n is the depth of the perfect binary tree

Result: A perfect binary tree graph and its leaves

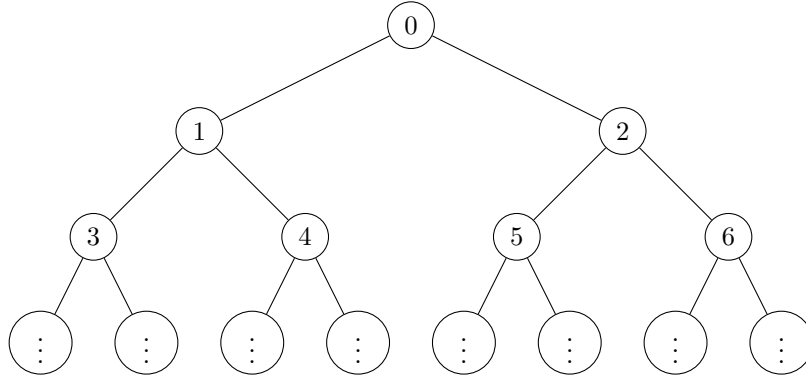
Function `perfect_binary_tree_generator(n):`

```

    if  $n = 0$  then
        return  $Graph()$ 
    else
         $g \leftarrow Graph()$ ;
         $g.add\_vertices([2^n])$ ;
        for  $i$  in range( $2^n - 1$ ) do
             $g.add\_edge(i, 2 * i + 1)$ ;
             $g.add\_edge(i, 2 * i + 2)$ ;
        end
        return  $g$ 
    end

```

The algorithm will generate a perfect binary tree of this form:



It is easy to see that the leaves will start with the value of $\left\lfloor \frac{2^{n+1} - 1}{2} \right\rfloor$, where n is the depth of the perfect binary tree.

So to generate all the leaves of the perfect binary tree of depth n , we present the following algorithm:

Algorithm 2: Perfect Binary Tree Leaves Generator

Data: $n \geq 0$, where n is the depth of the perfect binary tree

Result: A perfect binary tree graph's leaves

Function `perfect_binary_tree_generator(n):`

```

     $num\_vertices \leftarrow 2^{n+1} - 1$ ;
     $leaves \leftarrow []$ ;
     $last\_row\_start \leftarrow floor(num\_vertices/2)$ ;
    for  $vertex$  in  $range(last\_row\_start, num\_vertices)$  do
        |  $leaves.append(vertex)$ ;
    end
    return  $leaves$ 

```

We then use the algorithm from [6] to generate a coclique of maximum cardinality for our perfect binary tree.

Algorithm 3: Maximum Independent Set Algorithm

Data: A perfect binary tree graph T

Result: A maximum coclique of T

Function `maximum_independent_set(T):`

```

     $cliquer \leftarrow Cliquer(T)$ ;
    return  $cliquer.get\_maximum\_independent\_set()$ 

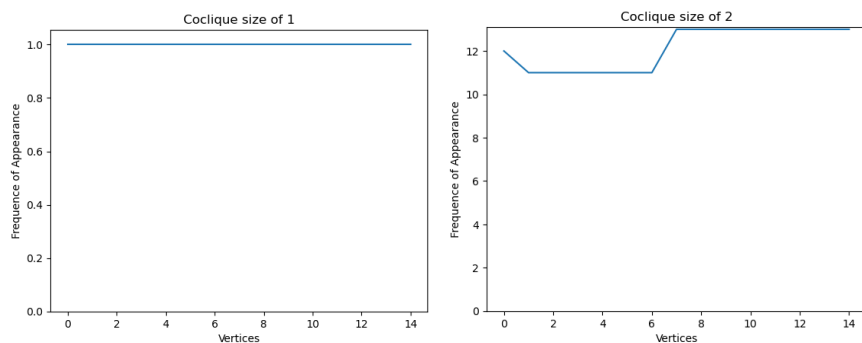
```

The next couple of pages show the results of the algorithm for a perfect binary tree of depth 4, 5, and 6. The X-axis denotes the vertex's labels (not the actual numbers) and the Y-axis denotes the cardinality of the stars centered around the vertices.

The data shown in the figures above verifies that the Partial HK-Property holds for perfect binary trees of depth 5. The next step would be to verify this for perfect binary trees of depth 6 and 7.

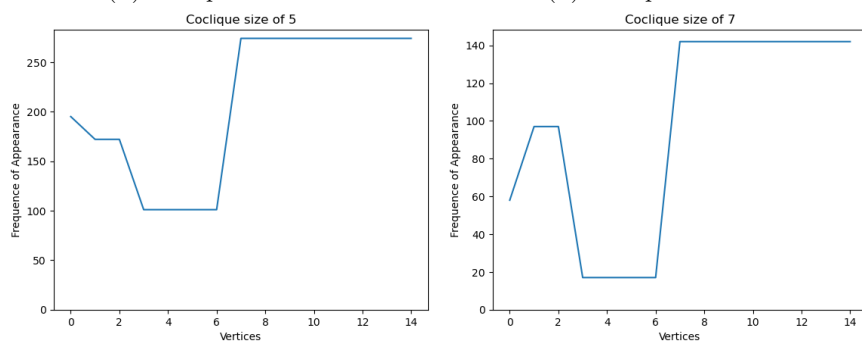
However, the algorithm is very slow and inefficient and it scaled exponentially. Hence, running the algorithm for perfect binary trees of depth 6 and 7 would be very computationally expensive.

Depth 4



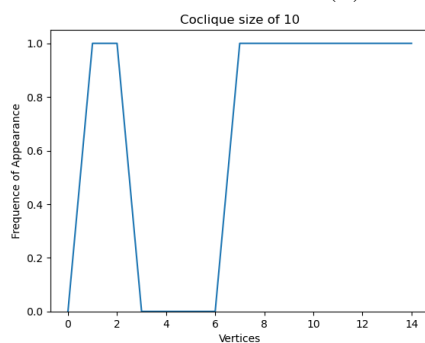
(A) coclique of size 1

(B) coclique of size 2



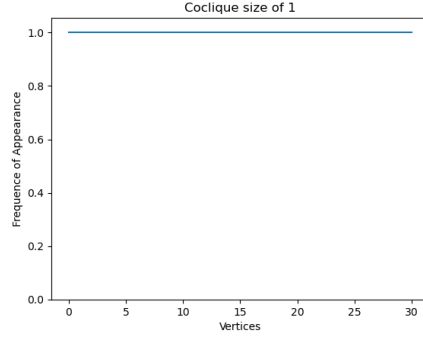
(C) coclique of size 5

(D) coclique of size 7

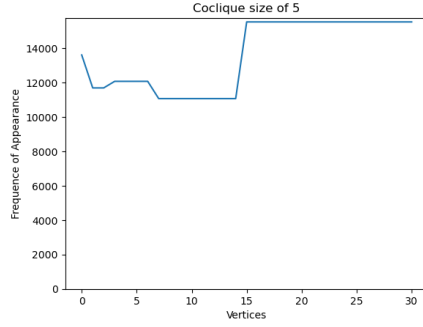


(E) coclique of size 10

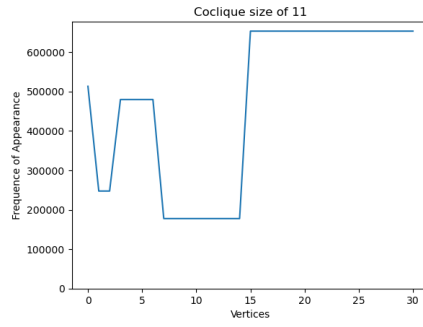
Depth 5



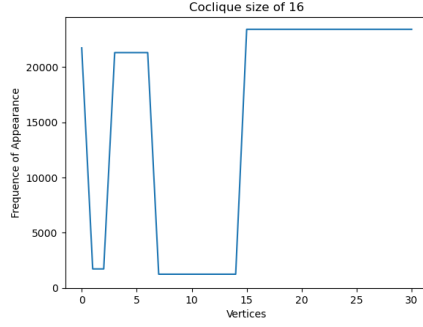
(A) coclique of size 1



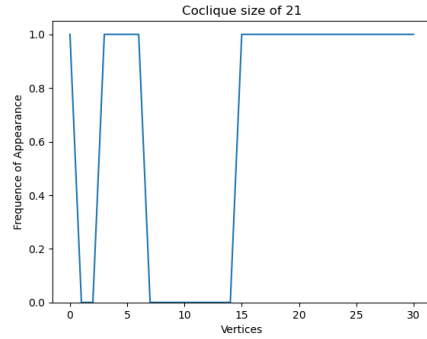
(B) coclique of size 5



(C) coclique of size 11

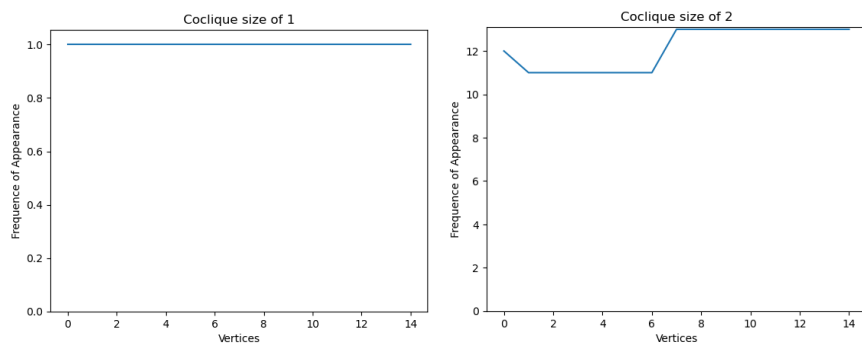


(D) coclique of size 16



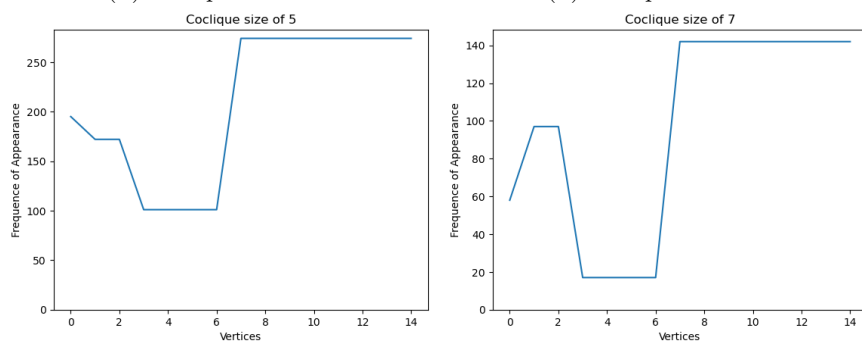
(E) coclique of size 21

Depth 6



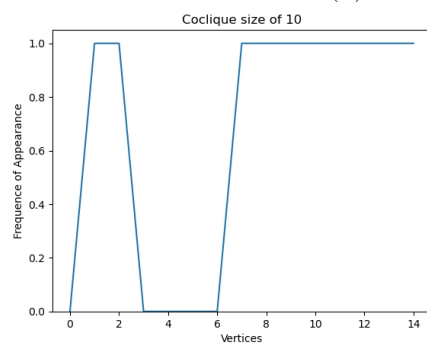
(A) coclique of size 1

(B) coclique of size 2



(C) coclique of size 5

(D) coclique of size 7



(E) coclique of size 10

REFERENCES

- [1] R. Baber. *Some results in extremal combinatorics*. ProQuest LLC, Ann Arbor, MI, 2011. Thesis (Ph.D.)—University of London, University College London (United Kingdom).
- [2] Peter Borg. Stars on trees. *Discrete Math.*, 340(5):1046–1049, 2017.
- [3] Peter Borg and Fred Holroyd. The Erdős-Ko-Rado properties of various graphs containing singletons. *Discrete Math.*, 309(9):2877–2885, 2009.
- [4] Emiliano J.J. Estrugo and Adrián Pastine. On stars in caterpillars and lobsters. *Discrete Appl. Math.*, 298:50–55, 2021.
- [5] Glenn Hurlbert and Vikram Kamat. Erdős-Ko-Rado theorems for chordal graphs and trees. *J. Combin. Theory Ser. A*, 118(3):829–841, 2011.
- [6] Sampo Niskanen and Patric R. J. Östergård. Cliquer user’s guide, version 1.0. In *Cliquer user’s guide, version 1.0*, 2003.