Available online at http://scik.org
J. Math. Comput. Sci. 7 (2017), No. 3, 537-553
ISSN: 1927-5307

UNIFORM SET LABELING VERTICES TO ENSURE ADJACENCY COINCIDES WITH DISJOINTNESS

MAHIPAL JADEJA*, RAHUL MUTHU

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar 382007, India

Abstract. Given a set of nonempty subsets of some universal set, their intersection graph is defined as the graph with one vertex for each set and two vertices are adjacent precisely when their representing sets have non-empty intersection. These sets may be finite or infinite (for example, in the case of geometric graphs). One can also study the reverse problem of expressing the vertices of a given graph as distinct sets in such a way that adjacency coincides with intersection of the corresponding sets. The problem of representing a graph as an intersection graph of sets was first introduced by Erdos et al. and they looked at minimising the underlying universal set necessary to represent any given graph and also showed that the problem is NP-complete. In this paper, we study a natural variant of this problem which is to consider graphs where vertices represent distinct as well as uniform cardinality sets and adjacency coincides with disjointness. Our motivation to look at disjointness instead of intersection is that several well-known graphs like the Petersen graph and Kneser graphs are expressed in the latter method. The parameters we take into account are: (1) UUSN the minimum universe size possible for obtaining uniform set labeling (2) ILN the smallest size of the largest label over all set labelings not necessarily uniform (where adjacency coincides with disjointness).

Keywords: uniform set labeling; set labeling; intersection number; intersection graphs; vertex labeling; Knesser graphs.

2010 AMS Subject Classification: 05C78.

*Corresponding author

E-mail address: jadeja_mahipal@daiict.ac.in

Received March 12, 2017; Published May 1, 2017

1. Introduction

Knesser graphs $KG_{n,k}$ are graphs whose vertices correspond to the k element subsets of an n element set and two vertices are adjacent precisely when their corresponding subsets are disjoint. Clearly if $n < 2k$ then the graph is an independent set of vertices. If $n = 2k$ then the graph is a matching. When $n = 2k+1$ we get the special family of odd graphs [6]. Knesser graphs are well studied [10] [11] [3]. Many problems on them can be solved clearly and efficiently using this set-theoretic definition. A natural question, therefore, is to try and model an arbitrary graph in this fashion. That is, come up with an underlying universal set and a choice of unique subsets to associate with each vertex such that adjacency is characterised by disjointness of the corresponding subsets. Clearly for an arbitrary graph the above choice of all identical sized subsets of a certain set will not work, because a graph defined in that manner is necessarily vertex transitive.

Our motivation to look at disjointness instead of intersection is that several well known graphs like the Petersen graph and Knesser graphs are expressed in the latter method, and the complements of these families are not well studied. Thus our choice is justified and not merely an attempt to artificially deviate from existing work.

The closely related concept is intersection graphs [9] for finite sets in which non-adjacency is characterised by disjointness of the corresponding subsets of underlying universal set. This was studied by Erdos et al. [4]. In that paper, they also obtain a tight upper bound of $n^2/4$ on intersection number where the sets are not required to be distinct. In Section 6 of that paper, the authors point out that the problem in general becomes more difficult when the constraint of the distinctness is added. They, however, observe that the universal upper bound applies to that variant as well. We, thus, make inroads into an open problem posed by them obtaining some general results as well as results for some special classes of graphs. We use the slightly different framework of disjointness graphs as many well known families of graphs are defined in this way, as mentioned earlier. So for a given graph, these two labeling approaches are entirely different (except for self-complementary graphs).

For a graph with m edges and n vertices, a trivial upper bound for intersection number is m (see [2]). Alon Noga [1] derived an upper bound of any N- vertex graph as a function of maximum degree of a graph: $2e^2(d+1)^2 lnN$ where d=maximum degree of the complement graph of G and e=base of the natural logarithm.

Since the problems of intersection and disjointness on graph representation are equivalent, the disjointness version is also NP Complete. The problem of finding a vertex labeling for an arbitrary graph using distinct sets for different vertices and all its standard variants we have listed are NP Complete. This follows from the fact that the equivalent problem of determining the intersection number of an arbitrary graph is NP Complete [4] [5]. Intersection graphs have many applications in the fields of scheduling, biology, VLSI design and they are also used for development of faster algorithms for optimisation problems.

In Section 2 the problem statement is discussed along with some basic results. Results related to UUSN and ILN are discussed in detail for some specific classes of graphs (including complement of complete graph, matching, paths, cycles, complete bipartite graph, complete binary trees) in Section 3. Cartesian product based method is also discussed in the same section. The final section summarises the results obtained in this paper.

2. Problem Statement

A set labeling of a graph $G(V,E)$ is a function $f : V \to \mathscr{P}(\{1,2,\ldots,k\}) - \{\phi\}$ where $k \in Z^+$ such that

- f is one one.
- $\forall u,v \in V, (u,v) \in E \Leftrightarrow f(u) \cap f(v) = \phi$.

For a uniform set labeling following additional condition is required.

- $\forall u,v \in V, |f(u)| = |f(v)| = c$ where $c \in Z^+$

Uniform Universe Size Number (UUSN) of a graph is the least positive integer k such that a uniform set labeling of G exists.

Individual label number(ILN) of a graph is the smallest size of the largest label over all set labelings (not necessarily uniform) of the vertices with uniques sets such that adjacency coincides with disjointness.

2.1 General Results on UUSN and ILN

Here, we present general bounds on UUSN and ILN .

Theorem 2.1. ILN(G) $<$UUSN(G), *where G has at least* 2 *vertices.*

Proof. Consider a valid and uniform set labeling of any graph G, optimal in terms of the universe size. Clearly, the number of elements used in total is UUSN(G). If the underlying set is used as label then exactly 1 label can be generated since it is impossible to use any of the subsets of the underlying set as a label of any other vertex (due to uniform cardinality constraint in this particular case). In this particular labeling, no vertex has more than UUSN$(G)-1$ elements in its label. This proves the result. $\square$

Theorem 2.2. UUSN$(G) \geq \lfloor log_2 n \rfloor + 1$ *and* ILN$(G) \geq 1$, *where G has n vertices.*

Proof. With the use of $\lfloor \log_2 n \rfloor$ elements, it is possible to generate at most $n-1$ non-empty subsets. Using these subsets, at most $n-1$ vertices can be labeled since repetitions of labels is not allowed. Here, $|V(G)| = n$ and therefore at least 1 additional element is required in order to assign non-empty as well as unique label to the n^{th} vertex. Therefore value of UUSN is at least $\lfloor \log_2 n \rfloor + 1$ for any given graph.

Since an empty set is not allowed to be used in any of the set labelings, the largest individual label size will be at least 1 in all possible set labelings of the given graph G. $\square$

Theorem 2.3. UUSN$(G+v)$ =UUSN$(G)+c$, *if* $d(v) = n(G)$. *Here,* $\forall u \in V(G), |f(u)| = c$ *and* $c \in Z^+$

Proof. Here the vertex v is adjacent to all other vertices and hence it is not possible to reuse any of the UUSN(G) elements for the labeling of vertex v. In order to obtain uniform labeling for $G+v$, $|f(v)|$ must be c. Therefore, exactly c additional elements are required for obtaining uniform labeling of $G+v$. $\square$

Theorem 2.4. UUSN$(K_n) = n$

Proof. UUSN$(K_1) = 1$ and from the application of Theorem 2.3 exactly $(n-1)$ times iteratively starting with K_1. $\square$

Theorem 2.5. $\mathrm{ILN}(G+v)=\mathrm{ILN}(G)$, *if* $d(v)=n(G)$.

Proof. Here the vertex v is adjacent to all other vertices and hence it is not possible to reuse any of the k elements used for valid set labeling for the labeling of vertex v.In order to obtain valid set labeling for $G+v$, it is sufficient to assign a singleton set {k+1} as a label for vertex v. $\mathrm{ILN}(G)$ is at least 1 (from Theorem 2.2) and hence value of ILN won't change for $G+v$. □

Theorem 2.6. $\mathrm{ILN}(K_n)=1$

Proof. $\mathrm{ILN}(K_1)=1$ and from the application of Theorem 2.5 exactly $(n-1)$ times iteratively starting with K_1. □

The following Theorem gives the universal upper bound on UUSN and ILN.

Theorem 2.7. $\mathrm{UUSN}(G)\leq n+\binom{n}{2}(n-1)$ *and* $\mathrm{ILN}(G)\leq n$.

Proof. We will give an algorithmic proof for the claim.

Algorithm 1: To find a valid uniform labeling for any given graph

Step 1: For any graph on n vertices start with the optimal valid labeling of the corresponding complete graph K_n with $\mathrm{UUSN}=n$.

Step 2: Delete the necessary set of edges one by one from this K_n to transform complete graph into the given graph.

After deleting each edge, the following operations are performed:

Step 2(a): Add an extra element to labels of both endpoints of that particular edge (to establish non-adjacency between endpoints).

Step 2(b): In order to obtain uniform labeling, add exactly 1 extra element in the labels of all $(n-2)$ vertices excluding the endpoint vertices considered in Step $2(a)$. So total $(n-1)$ new elements are required including the element which is added in the labels of the two endpoints of the deleted edge.

Total number of distinct elements used after step 1 is exactly n.

Total number of additional elements used after steps $2(a)$ and $2(b)$ is at most $\binom{n}{2}(n-1)$ because complete graph has exactly $\binom{n}{2}$ edges and in the worst case, all the edges are required to delete in order to construct the given graph. Therefore, $\mathrm{UUSN}(G)\leq n+\binom{n}{2}(n-1)$.

For the upper bound calculation of ILN, consider the step 1 and $2(a)$ of the Algorithm 1 only. Here, after step 1 $\text{ILN}(G)$ is 1 (Theorem 2.6). Step $2(a)$ can be applied at most $n-1$ times for each vertex and each iteration of step 2 will increase the $\text{ILN}(G)$ by at most 1. Therefore, $\text{ILN}(G) \leq n$. $\square$

3. Results on UUSN and ILN for some special families of graphs

In this section, we dreive results (either exact or asymptotic) on UUSN and ILN on the following classes of graphs: Paths, Cycles, Wheel Graph, Complement of Complete Graph $(\overline{K_n})$, Matching (M_{2n}), Complete binary trees, Complete bipartite graphs.

Theorem 3.1. *For matching,* $\text{UUSN} = O(\log_2 2n)$ *and* $\text{ILN} = O(\log_2 2n)$

Proof. Consider matching graph with n edges. $(|V| = 2n)$. Assume $|U| = k$ where U is the underlying universal set. Our objective is to minimise the value of k. Requirements for obtaining uniform labeling of the given graph can be summarized as below.

(1) Using $|k|$, we should be able to generate at least $2|n|$ non-empty subsets.
(2) For each subset p, a unique subset q must exist such that $p \cap q = \phi$. (So that q can be assigned as the label of unique vertex-neighbor of vertex with label p)
(3) Each subset must have non-empty intersection with all the remaining $2n-2$ subsets.
(4) Each subset must have the same cardinality.

If we consider only subsets of cardinality $k/2$ then each subset will have exactly one disjoint subset and it will have non-empty intersection with all the remaining $2n-2$ subsets (pigeonhole principle). Additionally, each subset will have the same cardinality of $k/2$. In order to fulfill the first requirement, $\binom{k}{k/2} \geq 2n$. By Stirling's approximation, UUSN=k=$O(\log_2 2n)$

Here, size of each individual label is $k/2$. Therefore, $\text{ILN} = O(\log_2 2n)$. $\square$

Theorem 3.2. $\text{UUSN}(\overline{K_n}) \leq 1.5(1 + \lceil \log_2 n \rceil)$ *and* $\text{ILN}(\overline{K_n}) = 2$

Proof. Consider A as underlying universal set with cardinality k. Here disjoint subsets are not allowed for labeling of independent set. For all S (where $S \subset A$), at most 1 subset can be used for labeling from each pair $(S, A-S)$. So the total available sets for labeling are reduced by half. Select the subset having higher cardinality in each pair and if cardinalities are same then make arbitary selection. By doing this, each subset will have cardinality at least $\frac{|k|}{2}$. All these selected subsets will have non-empty intersection (pigenhole principle) and hence they can be used to assign labels to vertices. Note: Empty set won't be used in labeling since in the pair (ϕ, A), A has higher cardinality. So, in summary total at most 2^{k-1} vertices of the graph can be labelled using A. In order to obtain uniform labeling, at most $\frac{|k|}{2}$ additional elements are required since the least possible cardinality is $\frac{|k|}{2}$ and greatest cardinality is k (since A is used as label and $|A| = k$). So after adding at most $\frac{|k|}{2}$ elements in each label, all the labels will have uniform cardinality of k. Therefore, total number of labels required is $1.5k$ in order to obtain uniform labeling of 2^{k-1} vertices.

So using $1.5k$ labels, it is possible to do valid and uniform labeling of $(\overline{K_n})$ where $n \in \{2^{k-2}+1, 2^{k-2}+2, \cdots, 2^{k-1}\}$ which proves the result for UUSN.

ILN is at least 2 for $\overline{K_n}$. This is because at least 1 element must be common in the labels of every vertex pair and in order to avoid repetition of labels, at least one more element is required to be added in one of the individual label.

Valid-ILN-labeling-$\overline{K_n}$

(1) For each $v_i \in \overline{K_n}$ assign $\{i\}$ as its label.
(2) Add $\{i+1\}$ in the labels of all $v_i \in \overline{K_n}$.

After the 2nd step, the cardinality of each individual label is 2 which proves the result for ILN. □

Theorem 3.3. UUSN $(K_{s,t}) \leq 1.5(2 + \lceil \log_2 s \rceil + \lceil \log_2 t \rceil) + |\lceil \log_2 s \rceil - \lceil \log_2 t \rceil|$
ILN $(K_{s,t}) = 2$

Proof. Complete bipartite graph consists of two independent sets. From the valid and uniform labeling of one independent set nothing can be used in the second independent set. So labeling of these two independent sets must be entirely disjoint. Now cardinality of each vertex label in first partite set is $1 + \lceil \log_2 s \rceil$ whereas cardinality of each vertex label in second partite set is $1 + \lceil \log_2 t \rceil$. In order to make cardinality of each label same, $|\lceil \log_2 s \rceil - \lceil \log_2 t \rceil|$ must be added to all the vertex labels to one of the smaller partite set (the partite set which has smaller individual label size for all vertices). This proves the result for UUSN. From Theorem 3.2, it is possible to label both partite sets with ILN 2 and ILN will remain 2 for the whole graph if disjoint sets are used for the labeling of the both partite sets. This proves the result for ILN. □

Theorem 3.4. UUSN$(P_n) = O(logn)$.

Add-edge Procedure:

Input: Valid labeling of any given path (P_n) using exactly k labels.

Output: Valid labeling of P_{n+2} using exactly $k+3$ labels.

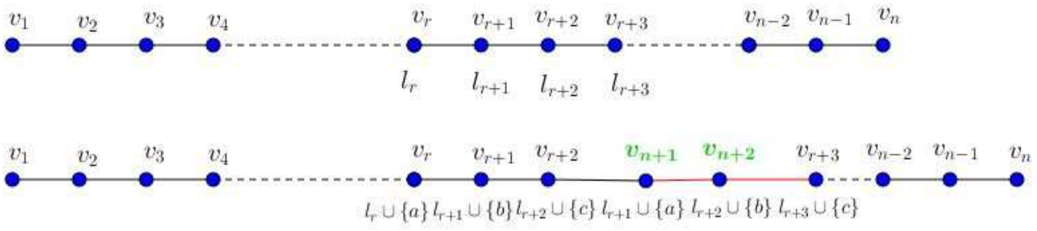

FIGURE 1. Summary of add-edge procedure

Step 1: Identify P_4 in the given P_n. Say, vertices of P_4 are v_r, v_{r+1}, v_{r+2} and v_{r+3} and corresponding labels are l_r, l_{r+1}, l_{r+2} and l_{r+3} respectively.

Step 2: Add 2 new vertices v_{n+1} and v_{n+2}. In order to construct P_{n+2}, add 3 new edges (v_{r+2}, v_{n+1}), (v_{n+1}, v_{n+2}), (v_{n+2}, v_{r+3}).

Step 3: $l(v_{n+1}) = l_{r+1}$

$l(v_{n+2}) = l_{r+2}$

Step 4: Three pairs (v_r, v_{n+1}), (v_{r+1}, v_{n+2}) and (v_{r+2}, v_{r+3}) are non-adjacent. In order to preserve non-adjacency add 3 distinct new elements say a, b and c to the labels of these 3 pairs respectively. The final labeling is valid. It preseves adjacency as as well non-adjacency for all pairs of vertices.

Lemma 1. *For the given P_n, add-edge procedure can be applied for at most $\frac{n-1}{3}$.*

Proof. Add-edge procedure can be applied to each edge-disjoint P_4-subgraph of the given P_n. Every 1 out of 3 edges of P_4 can be used for the procedure. Total number of edges in P_n is $n-1$. Therefore, for the given P_n, at most $\frac{n-1}{3}$ times procedure add-edge can be applied. □

Proof of Theorem 3.4:

We will give an algorithmic proof for the claim.

Algorithm-Valid-Uniform-Pathlabeling:

Input: Valid and uniform labeling of P_n using exactly k labels.

Output:Valid and uniform labeling of P_{n+2i} using exactly $k+3$ labels where $0 < i \leq \frac{n-1}{3}$ and $i \in N^+$

Step 1: Apply add-edge procedure on the given P_n for the first 4 vertices i.e. $r = 1$.

Step 2: For $j \geq 1$, (where $j \in N^+$)

If $3j+1 < n$ and $3j+4 \leq n$ then apply add-edge procedure for $r = 3j+1$ with the following modifications in the step 4 of the procedure:

(Observation: v_{3j+1} participates in exactly two add-edge procedure, in iteration j as well as in iteration $j-1$. Therefore, before starting of iteration j, the label of v_{3j+1} already has an additional element p due to iteration $j-1$ where $p \in \{a,b,c\}$ For $j = 1$, $p = \{c\}$ from step 1. The label of v_{3j+1} won't change during the j^{th} iteration and $p \in l_{3j+1}$.)

(1) Add p in the label of newly added vertex v which is the neighbour of v_{3j+3}.
(2) Use remaining two elements $\{a,b,c\} - \{p\}$ in any order, for the remaining two pairs.

Step 3: After j^{th} iteration of the addedge procedure, the cardinality of the labels of the first $3j+1$ vertices will be increased by exactly 1. So the first $3j+1$ vertices of the path has a uniform labeling. The label of the $(3j+1)^{th}$ vertex will certainly contain exactly one of the 3 elements namely a,b,c. WLOG a is used in the label of the $(3j+1)^{th}$ vertex.

Step 4: Partition the remaining vertices $V \setminus \{V_1, V_2, \ldots, V_{3j+1}\}$ into 2 sets: A: Even numbered vertices and B: Odd numbered vertices.

Add b to all labels of A and add c to labels of B. After this step, the cardinality of the remaining vertices are also increased by 1. In general, the cardinality of each vertex is increased by

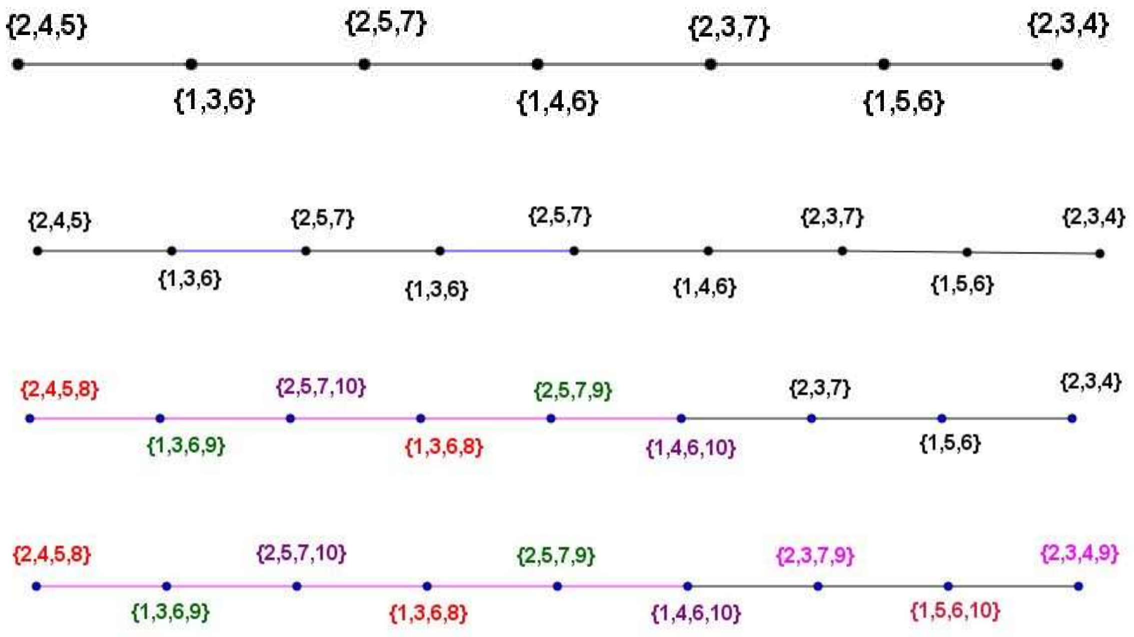

FIGURE 2. UUSN labeling- P_7 to P_9

exactly 1. So the final labeling is uniform and valid. No additional elements are required in this procedure.

Upper bound calculation:

So at most $\lfloor \frac{n-1}{3} \rfloor$ edges can be used and they will generate at most $\dfrac{2n-2}{3}$ new vertices.

Recurrence: $T(5n/3) = T(n) + 3$

UUSN : $O(\log n)$ □

Note: We can generate paths for all values of n by applying the add-edge procedure repeatedly. One can generate all P_{2x+1} by repeatedly applying the add-edge procedure on a valid and uniform labeling of P_7, similarly all P_{2x} can be generated by application of the add-edge procedure on a valid and uniform labeling of P_6. Here $x > 3$ and $x \in \mathcal{N}$.

Theorem 3.5. $\text{UUSN}(C_n) = O(\log n)$. *and* $\text{UUSN}(W_n) = O(\log n)$.

Proof. For cycles results are slightly better for some values of n because C_n contains exactly one more edge as compared to P_n and hence it is possible to apply add-edge procedure exactly 1 more time as compared to P_n (specifically when n is multiple of 4).

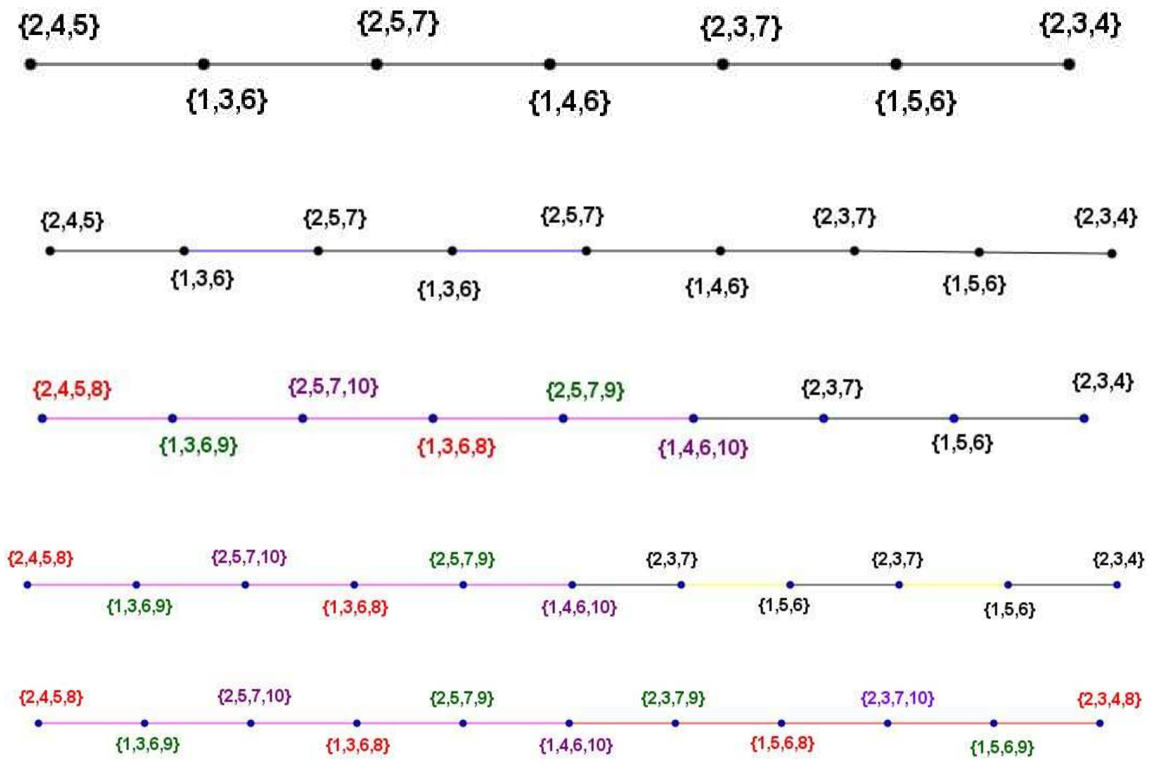

FIGURE 3. UUSN labeling- P_7 to P_{11}

Wheel graph consists of C_n and one additional vertex with degree n. Therefore, $\text{UUSN}(W_n) = \text{UUSN}(C_n) + c$. (from Theorem 2.3) $\square$

Theorem 3.6. $\text{ILN}(P_n) = O(logn)$.

Proof. Procedure addedge may increase the cardinality of individual labels by at most one. Hence New ILN= Old ILN+ 1 (After i^{th} iteration, where $i \geq 1$) If we obtain P_n using addedge procedure, then $\text{ILN}(P_n) = \text{UUSN}(P_n)/3$. So $\text{ILN}(P_n)$= $O(\log n)$ $\square$

The same idea is also applicable on cycles as well as wheel graph.

Theorem 3.7. *$\text{ILN}(C_n) = O(logn)$ and $\text{ILN}(W_n) = O(logn)$.*

Theorem 3.8. *$\text{UUSN}(BT_n) = O(logn)$ and $\text{ILN}(BT_n) = O(logn)$. Where BT=Complete binary tree and n denotes the total number of vertices of the BT.*

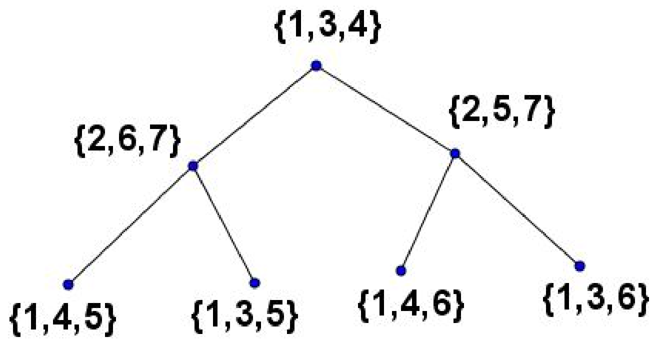

FIGURE 4. Input

Proof. We prove this result by an iterative algorithm: Valid-Uniform-TreeLabeling. The algorithm is explained below:

Input: A valid as well as uniform labeling of the complete binary tree of height h using exactly K labels.

Base case: $\text{UUSN}(BT_7) = 7$. Base case is shown in Fig. 4 with underlying labeling set: $\{1,2,3,4,5,6,7\}$.

Output: A valid and uniform labeling of complete binary tree of height $h+1$ using exactly $K+10$ labels. The ten new labels are a,b,c,d,e,f,g,h,i,j.

Step 1: For all newly added vertices in level $L+1$, find their corresponding ancestors in level $L-1$.

For all v (where v is a level $L+1$ vertex) ,

Label(v)= Label(ancestor(v) in level $L-1$) . (The levels L+1 and L-1 are highly similar with respect to adjacency with the remaining levels. Both of these levels are adjacent to level L and non-adjacent to $1,2,3,\ldots,L-3$.)

Observation after step 1:

1. All the vertex-labels which are present in level $L+1$ will have non-empty intersection because corresponding ancestors are either same or having non-empty intersection. This is desirable because all vertices of level $L+1$ are non adjacent.

2. Layer $L+1$ and $L-1$ are non adjacent and they have non-empty intersection after this step.

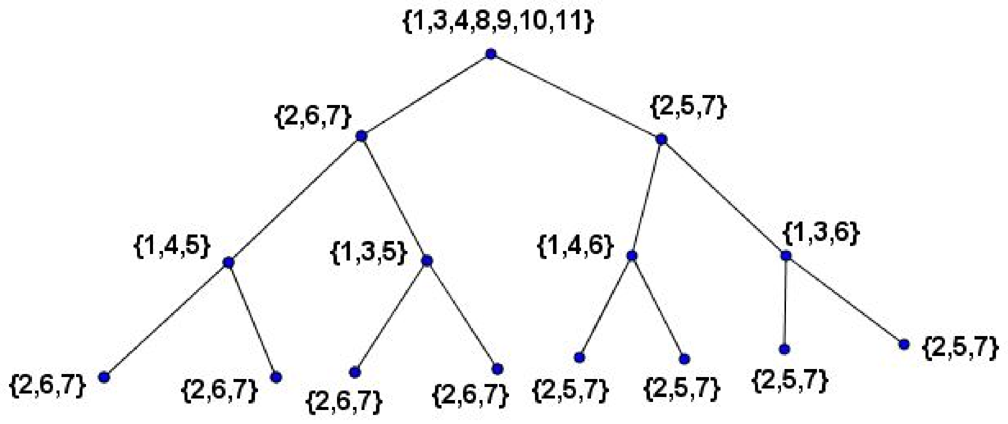

FIGURE 5. After Step 1 and 2

Step 2: The level $L-2$ is adjacent to $L-1$ but not to $L+1$.

For all v', (where v' is a level $L-2$ vertex)

$New\text{-}Label(v') = Old\text{-}Label(v') \cup \{a,b,c,d\}$

Step 3: Consider sequential ordering (left to right) of vertices which are present in level $L+1$.

For each vertex v_i,

$$\begin{aligned} New\text{-}Label(v_i) &= Old\text{-}Label(v_i) \cup \{a\} \text{ if } i \mod 4 = 1 \\ &= Old\text{-}\ Label(v_i) \cup \{b\} \text{ if } i \mod 4 = 2 \\ &= Old\text{-}\ Label(v_i) \cup \{c\} \text{ if } i \mod 4 = 3 \\ &= Old\text{-}Label(v_i) \cup \{d\} \text{ if } i \mod 4 = 0 \end{aligned}$$

Vertices of level $L-2$ and $L+1$ will preserve non-adjacency because the corresponding labels have non-empty intersection after this step.

Step 4: Consider sequential ordering (left to right) of vertices which are present in level $L : v_1, v_2, \ldots, v_q$

For each vertex v_i,

$$\begin{aligned} New\text{-}Label(v_i) &= Old\text{-}Label(v_i) \cup \{c,d\} \text{ if } i \mod 2 = 1 \\ &= Old\text{-}\ Label(v_i) \cup \{a,b\} \text{ if } i \mod 2 = 0 \end{aligned}$$

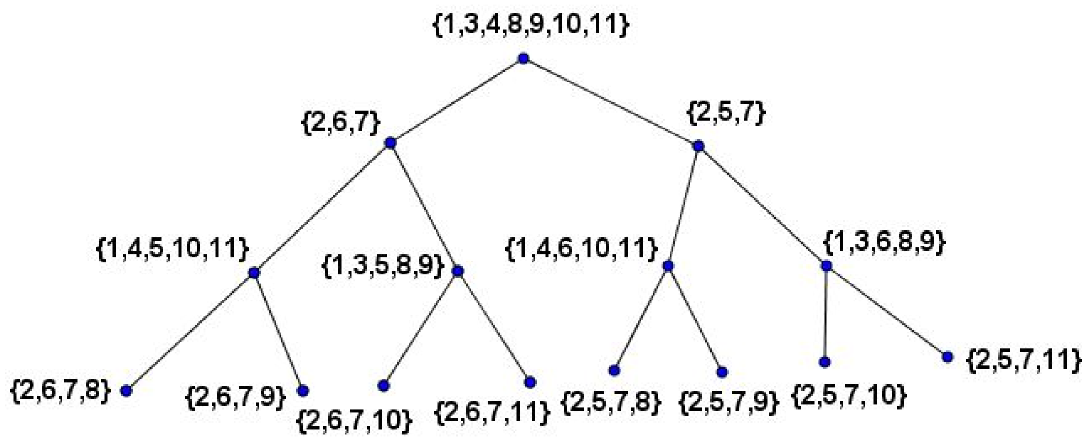

FIGURE 6. After Step 4

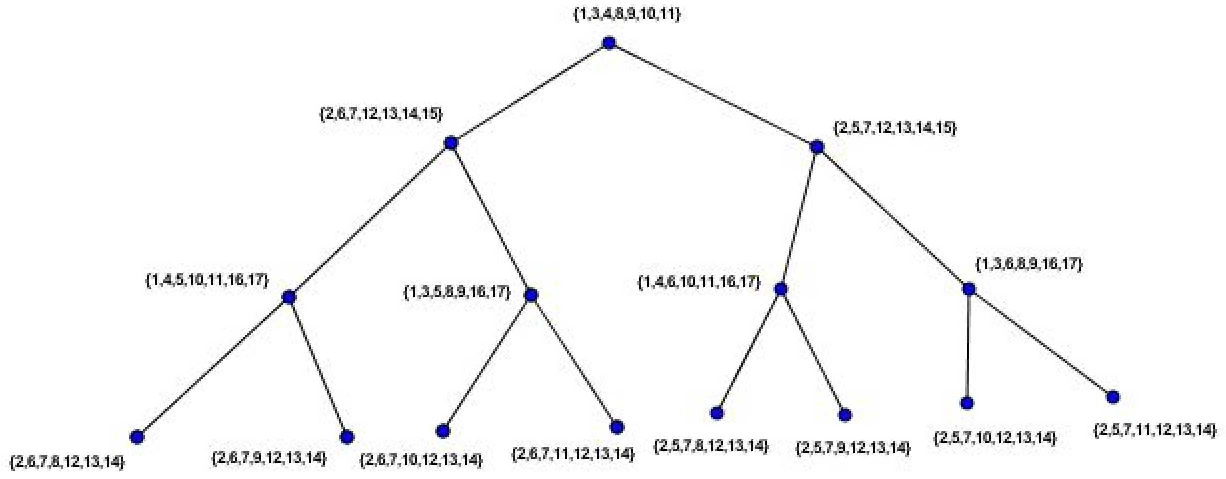

FIGURE 7. Output

Observation after step 4:

Level $L+1$ and L will preserve adjacency as well as non-adjacency.

The labeling after this step is valid and preserve adjacency as well as non-adjacency.

Step 5: Following changes are required in order to obtain the uniform labeling:

(1) Add $\{e, f, g, h\}$ to the labels of all the vertices of levels $(L-1)$, $(L-3)$, $(L-5)$, …, 1. (if L is even).

(2) Add $\{a, b, c, d\}$ to the labels of all the vertices of levels $(L-4)$, $(L-6)$, $(L-8)$, …, 2. (if L is even).

(3) Add $\{i, j\}$ to the labels of all the vertices of level L.

(4) Add $\{e, f, g\}$ to the labels of all the vertices of level $L+1$.

Note: If L is odd then consider levels $(L-1)$, $(L-3)$, $(L-5)$, …, 2 for Substep 1 and $(L-4)$, $(L-6)$, $(L-8)$, …, 1 for Substep 2 of Step 5.

For the complete binary tree, $n = 1+2+4+2^h = 2^{h+1}-1$ i.e. $h = O(logn)$. For valid and uniform labeling of each layer exactly 10 additional elements are required. Total number of layers are $O(logn)$. Therefore total number of elements required are $10 * O(logn)$ which is $O(logn)$.

Size of individual label is increased by exactly 4 after each iteration and total number of iterations are $O(logn)$. Therefore, ILN is $4 * O(logn)$ which is $O(logn)$. □

One of the possible practical application of the previous result is visulization of multiple hierarchies [7] which uses idea of treemaps [8].

3.1 Cartesian Product Based Method

Key Observation: Let A, B, C, D be sets. Then $(A \times B) \cap (C \times D) \neq \emptyset$ if and only if $A \cap C \neq \emptyset$ and $B \cap D \neq \emptyset$.

Theorem 3.9. *Let G and H be two graphs on the same vertex set V. Further, suppose $E(G) \cap E(H) = \emptyset$. Then* UUSN$(G+H) \leq$ UUSN$(G) \times$UUSN(H) *and* ILN$(G+H) \leq$ ILN$(G) \times$ILN(H).

Proof. Take optimal labelings of the vertex set using disjoint universes for the graphs G and H. Now consider the graph $G+H$. The vertex sets of G and H are identical. For each vertex v in $G+H$ give it the label $l_G(v) \times l_H(v)$. Clearly two vertices are nonadjacent only if they are nonadjacent in both G and H. In that case their labels under the two labelings will each be intersecting. From the key observation, it follows that their cartesian product new label will also intersect. Similarly for the case of non-intersection (adjacent vertices). □

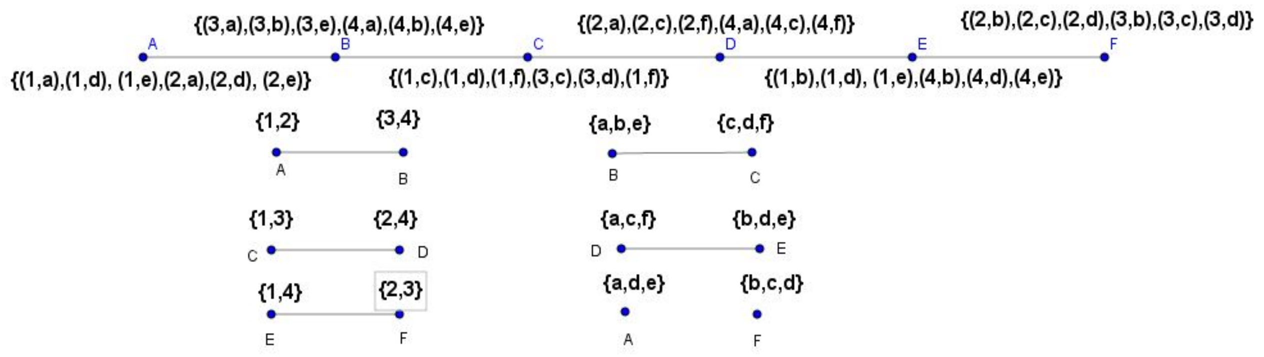
{(3,a),(3,b),(3,e),(4,a),(4,b),(4,e)}
{(2,a),(2,c),(2,f),(4,a),(4,c),(4,f)}
{(2,b),(2,c),(2,d),(3,b),(3,c),(3,d)}
A
B
C
D
E
F
{(1,a),(1,d), (1,e),(2,a),(2,d), (2,e)}
{(1,c),(1,d),(1,f),(3,c),(3,d),(1,f)}
{(1,b),(1,d), (1,e),(4,b),(4,d),(4,e)}
{1,2}
{3,4}
A
B
{a,b,e}
{c,d,f}
B
C
{1,3}
{2,4}
C
D
{a,c,f}
{b,d,e}
D
E
{1,4}
{2,3}
E
F
{a,d,e}
{b,c,d}
A
F

FIGURE 8. Cartesian Product Based Method

As a corollary we have the following theorem.

Theorem 3.10. $\text{USN}(P_n) \leq (1 + \log \frac{n}{2})^2$.

Proof. The path is the union of two disjoint matchings. Each matching has UUSN $O(\log n)$. From Theorem 3.9, we see that the graph has UUSN $O(\log n)^2$. We state this here just as an application since we have a better bound for paths (Theorem 3.4). □

4. Conclusion and Future work

We have obtained upper bound of UUSN and ILN for the complement of complete graphs, complete graphs, complete binary trees, complete bipartite graphs, paths, cycles, matching, wheel graph etc. In the future we plan to derive optimal and/or lower bound results for hypercube, harary graph etc.

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] Noga Alon, Covering graphs by the minimum number of equivalence relations, Combinatorica 6 (1986), no. 3, 201–206.

[2] VK Balakrishnan, Schaum's outline of graph theory: Including hundreds of solved problems, McGraw Hill Professional, 1997.

[3] Ya-Chen Chen, Kneser graphs are hamiltonian for n 3k, J. Comb. Theory, Ser. B 80 (2000), no. 1, 69–79.

[4] Paul Erdos, Adolph W Goodman, and Lajos Pósa, The representation of a graph by set intersections, Canad. J. Math 18 (1966), no. 106-112, 86.

[5] Michael R Gary and David S Johnson, Computers and intractability: A guide to the theory of np-completeness, 1979.

[6] CD Godsil, More odd graph theory, Discrete Math. 32 (1980), no. 2, 205–207.

[7] Mahipal Jadeja and Rahul Muthu, Labeled object treemap: A new graph-labeling based technique for visualizing multiple hierarchies, Ann. Pure Appl. Math. 13 (2017), no. 1, 49–62.

[8] Mahipal Jadeja and Kesha Shah, Tree-map: A visualization tool for large data., GSB@ SIGIR, 2015, pp. 9–13.

[9] Terry A McKee and Fred R McMorris, Topics in intersection graph theory, SIAM, 1999.

[10] Alexander Schrijver, Vertex-critical subgraphs of kneser-graphs, Mathematisch Centrum, Afdeling Zuivere Wiskunde, 1978.

[11] Saul Stahl, The multichromatic numbers of some kneser graphs, Discrete Math. 185 (1998), no. 1-3, 287–291.