Chapter-2

Generating the Vertex Sets with some Distance Parameter Properties in Caterpillar Graphs

1. INTRODUCTION:

In this chapter the vertices of caterpillar tree are viewed with different approach and categorized into the sets, and based on the distance parameters i.e., diameter and radius. The distance parameters have been presented with some set theory views. Here is the set of diametral vertices, is the set of central vertices and is the set of vertices which are neither central nor peripheral. Then, the cardinality of these sets has some property and helps to specify the basic characters of caterpillar tree. A linear complexity algorithm is also designed to generate these sets.

2. BASIC DEFINITIONS:

2.1 Distance:

The distance between two vertices $u$ and $v$, denoted $d(u,v)$, is the length of a shortest $u-v$ path, also called a $u-v$ geodesic.

Algorithm for Distance:

2.1.1To find the shortest distance

for $k \leftarrow 0$ to $n - 1$
for $i \leftarrow 0$ to $n - 1$
for $j \leftarrow 0$ to $n - 1$

$$D[i][j] = \min 1 \{D[i][j]$, $D[j][k]$, $D[k][i]\}$$

End $j$
End $i$
End $k$
2.2 Eccentricity:

The eccentricity $e_G(v)$ of a vertex $v$ is defined as the distance between $v$ and a vertex furthest apart from $v$, 

i.e. $e_G(v) = \max_{u \in V(G)} d_G(v, u)$

Algorithm: Eccentricity

To find eccentricity:

for $i \leftarrow 0$ to $n - 1$

Initialize $max = D[i][0]$

for $j \leftarrow 0$ to $n - 1$

if $D[i][j] > max$

$\quad max = D[i][j]$

End $j$

$\quad e[i] = max$

End $i$

2.3 Diameter:

The largest of the eccentricities of the vertices of a graph $G$ (or alternatively the largest of the distances between its vertices) is the diameter, denoted by $\text{diam}(G)$.

Let $G$ be a graph and $v$ be a vertex of $G$. The diameter of $G$ is the maximum eccentricity among the vertices of $G$.

Thus, $\text{diameter}(G) = \max\{e(v): V \in V(G)\}$.

Algorithm: Diameter

To find Diameter:

Initialize $min$ and $max = e[0]$

for $i \leftarrow 0$ to $n - 1$

if $e[i] > max$

$\quad max = e[i]$

else if $e[i] < min$

$\quad min = e[i]$

End $i$

Diameter = $max$. 
2.4 Radius:

The radius of G, denoted by rad(G), is the smallest of the eccentricities of the vertices.

Algorithm: Radius

To find the radius

Initialize \( min \) and \( max = e[0] \)
for \( i \leftarrow 0 \) to \( n - 1 \)
if \( e[i] > max \)
    \( max = e[i] \)
else if \( e[i] < min \)
    \( min = e[i] \)
end i

Radius = \( min \)

Fig: 2.1 Example of Graph with their Distance, Eccentricity, Diameter and Radius are given below:

Distance (f,c) : 2
Distance (g,c): 2
Distance (a,c): 3
Eccentricity(f):2
3. CATERPILLAR TREE:

A caterpillar is a tree in which every graph vertex is on a central stalk or only one graph edge away from the stalk (in other words, removal of its endpoints leaves a path graph; A tree is a caterpillar iff all nodes of degree $\geq 3$ are surrounded by at most two nodes of degree two or greater.

Each of the vertex is given with its eccentricity. The diameter of the graph is 6 and radius is 3. There are 3 diametral vertices and one radius vertex.

- They are the trees in which every node of degree at least three has at most two non-leaf neighbors.
- They are the trees that do not contain as a subgraph the graph formed by replacing every edge in the star graph $K_{1,3}$ by a path of length two.
- They are the connected graphs that can be drawn with their vertices on two parallel lines, with edges represented as non-crossing line segments that have one endpoint on each line.
- They are the trees whose square is a Hamiltonian graph. That is, in a caterpillar, there exists a cyclic sequence of all the vertices in which each adjacent pair of vertices in the sequence is at distance one or two from each other, and trees that are not caterpillars do not have such a sequence. A cycle of this type may be obtained by drawing the caterpillar on two parallel lines.
and concatenating the sequence of vertices on one line with the reverse of the sequence on the other line.

They are the trees whose line graphs contain a Hamiltonian path; such a path may be obtained by the ordering of the edges in a two-line drawing of the tree. More generally the number of edges that need to be added to the line graph of an arbitrary tree so that it contains a Hamiltonian path (the size of its Hamiltonian completion) equals the minimum number of edge-disjoint caterpillars that the edges of the tree can be decomposed into.

4. SET REPRESENTATION OF VERTICES OF CATERPILLAR TREE:

A graph G is defined as the set of vertices and edges, i.e. \( G = (V, E) \). Where \( V \) is the set of vertices and \( E \) is the set of edges. The set \( V \) can be further divided into different subsets with some properties assigned to the vertices. For example, the subset of \( V \) may be a set of dominating vertices or a set of independent vertices.

In this paper the vertices of caterpillar tree \( T \) are viewed with different perspective and categorized into the sets \( D, R \) and \( X \) based on the distance parameters, diameter and radius. The distance parameters have been presented with some set theory views. The details and examples are as follows.

Let be a caterpillar tree with vertices. Every caterpillar tree \( T \) has some diametral (peripheral) vertices and central vertices. In addition to these, some of the caterpillars also have the vertices which are neither central nor diametral. Let us call these vertices as \(-\)vertices. Based on these three properties, vertex set of caterpillar can be categorized. The vertices of can be grouped into three sets, and by considering the distance parameters diameter and radius.

Here \( D \) is the set of all diametral vertices. It is represented as:

\[
D = \{ v \mid v \text{ is diametral vertex} \}
\]

\( R \) is the set of all central vertices. Represented by:

\[
R = \{ v \mid v \text{ is central vertex} \}
\]
$X$ is the set of all vertices which are neither central nor diametral. We call these vertices as $x$-vertices. Represented by:

$$X = \{ v \mid v \text{ is a } x-\text{vertex} \}$$

The sum of cardinality of each set gives us total number of vertices in $T$. It is clearly seen that using the above sets we can write:

$$|D| + |R| + |X| = n \quad \Rightarrow (1)$$

From the Figure 3, see that $n = 9$, and the set of vertices is $V = \{ a, b, c, d, e, f, g, h, i \}$. Each of the vertex is given with its eccentricity. The diameter of the graph is 6 and radius is 3. There are 3 diametral vertices, 1 central vertex and 5 vertices whose eccentricity neither equals to diameter nor equals to radius. Therefore $|D| = 3$, $|R| = 1$ and, $|X| = 5$. Expression 1 can be easily verified.

There are different conditions for these sets and their cardinality. These are dealt in detail in further sections.

3 SOME BASIC PROPERTIES OF THESE SETS:

Fig: 2. 3 Caterpillar Tree with Distance, Eccentricity, Diameter and Radius

4.1. When $|D| = n$ or $|R| = n$:

When $|D| = n$ or $|R| = n$, it means that all the vertices are present either in the set $D$ or in the set $R$ i.e., diameter=radius. Hence $T$ is self centered. Of course in this case $|X| = 0$. The only caterpillars that are self centered are $K_1$ and $K_2$. In Figure 4 the eccentricity of each vertex is specified.
4.2. When \(|R| = 1\) or \(|R| = 2\):

When \(|R| = 1\), it means that there is only one central vertex, and hence the \(T\) is central or unicentral. And while \(|R| = 2\), the \(T\) is bicentral. In [2] it has been proved that a tree can have at the most two central vertices. So \(1 \leq |R| \leq 2\). In this case \(|D| \neq 0\) and \(|X| \neq 0\).

5. VERTICES ON THE SPINE OF CATERPILLAR:

The path induced by removing the end vertices of \(T\) is called spine of caterpillar [1, 2]. This path consists of central and \(x\)-vertices and \(-\)vertices occur in pairs of same eccentric values. It can be seen in Figure 3 that the spine consists of path \(c,d,e,g,h\) and these are the vertices with eccentric vertices 5, 4, 3, 4, 5 respectively. Except the vertex \(d\), the central vertex, the other vertices are the \(x\)-vertices.

From Figure 3, we observe that all the vertices on the spine, except the central vertices, belong to the set \(X\). The eccentric values of vertices are such that it ranges from maximum eccentricity to minimum one, i.e. from diameter to radius. The number of paired spinal \(x\)-vertices can be computed by decrementing the diameter value by 1 till it is equal to radius. The same is represented as,

\[
\begin{align*}
&d, \quad d - 1, d - 2, \ldots, r + 1, \quad r, \quad r + 1, \ldots, d - 1, d \\
&\text{spine vertices of unicentral caterpillar}
\end{align*}
\]

\[
\begin{align*}
&d, \quad d - 1, d - 2, \ldots, r + 1, \quad r, r, \quad r + 1, \ldots, d - 1, d \\
&\text{spine vertices of bicentral caterpillar}
\end{align*}
\]

By ignoring central and diametral vertices we get the eccentricity of \(x\)-vertices on the spine,
The number of pairs of spinal x-vertices, $S_p$, (except Central vertices) can be obtained by,

\[
S_p = \sum_{i=r+1}^{d-1} 1
\]

\[
= d - 1 - (r + 1) + 1
\]

\[
= d - r - 1 + 1
\]

\[
S_p = d - r - 1
\]

And, therefore the total number of spinal vertices including central vertices, can be obtained by the following expression,

\[
\text{No. of Spine vertices are }= 2S_p + |R| = -(2).
\]

From the Figure 3, see that,

\[
d = 6, r = 3 \text{ and } |R| = 1
\]

\[
S_p = 6 - 3 - 1 = 2.
\]

Therefore,

\[
\text{no. of spine vertices } = 2 * 2 + 1 = 5.
\]

6. ALGORITHM TO GENERATE THE SETS D, R AND X :

The sets of diametral vertices, central vertices and x-vertices can be generated by the following method. First we need to find the eccentricity of each vertex and then have to find the maximum (diameter) and minimum (radius) of the eccentricities. The eccentricity of the vertices can be obtained by Floyd’s algorithm [3], which takes $O(n^3)$ if we use matrix. Finding the maximum and minimum of the eccentricities can be done in $O(n)$ time by using divide and conquer Minimum and Maximum algorithm [4, 5]. The following algorithm has been developed which determines the sets $D, R$ and $X$ and their cardinality in $O(n)$ time.

Let $T[n][n]$ be the input caterpillar tree, $n$ be the number of vertices of $T$. Let the arrays $D, R, X$ and $e[]$ represent the sets $D, R, X$ and eccentricity of each vertex. Let
$D_{count}$, $R_{count}$ and $X_{count}$ be the variables which will hold the cardinality of each set. Let the variables $diam$ and $rad$ hold the value of diameter and radius respectively.

6.1. Algorithm

Algorithm Generate_Sets(int $n$, int $e[]$)
{
    int $D_{count} = 0$, $R_{count} = 0$, $X_{count} = 0$;
    for $i = 1$ to $n$
    {
        if $e[i] = diam$
        {
            $D[D_{count}] = e[i]$;
            $D_{count}++$;
        }
        else if $e[i] = rad$
        {
            $R[R_{count}] = e[i]$;
            $R_{count}++$;
        }
        else
        {
            $X[X_{count}] = e[i]$;
            $X_{count}++$;
        }
    }
}
6.2. Analysis of Algorithm:

Since the algorithm has one for loop, the time complexity can be given as following:

\[ T(n) = \sum_{i=1}^{n-1} 1 \]
\[ = n - 1 + 1 \]
\[ = n \]

Therefore

\[ T(n) = \mathcal{O}(n) \]

And hence the algorithm is linear.

7. SOME PROPOSITIONS:

Here are some propositions which can be easily observed by the given conditions. The caterpillar can be easily characterised as self centered and almost self centered by the following corollaries. [6, 7 and 8].

7.1 Proposition 1: If \( |R| = 0 \) AND \( |X| = 0 \) AND \( |D| = n \) then the caterpillar is self centered.

Or

If \( |R| = n \) AND \( |X| = 0 \) AND \( |D| = 0 \), then the caterpillar is self centered.

7.2 Proposition 2: If \( |D| = 2 \) AND \( |R| = n - 2 \) AND \( |X| = 0 \) then the caterpillar tree is almost self centered of type \((r + 1, r)\).

7.3 Proposition 3: If \( |D| = n - 1 \) AND \( |R| = 1 \) AND \( |X| = 0 \) then the caterpillar tree is almost self centered of type \((r, r + 1)\).

8. PROGRAM:

8.1 We proposed another program which finds the eccentricity diameter and radius of graphs using DFS. We implemented it through linked list

```c
#include<stdio.h>
#define FALSE 0 
#define TRUE 1 
#define SIZE 15
```
#Typedef struc node
{
    int inf;
    int weight;
    struct node *link;
} node;
typedef struct table
{
    int visit;
    char data;
    node *nodeptr;
} table;
table *tab[SIZE]
int max=0,n,e[50],I, j;
Void dfs (int);
Void create (int)
Void main()
{
    int start, radius, center[20],直径, min;
    node *cur;
    clrscr();
    Printf("Enter the no of nodes:");
    Scanf("%d", &n);
    Create (n);
    for (i=0; i<n; i++)
    {
        e[i]=0;
        max = 0;
        for (j=0; j<n; j++)
            tab [i]-> visit = FALSE;
        tab [i]-> visit = TRUE;
        dfs(i);
        e [i]=max;
for (j=0;j<n;j++)
    tab [j]->visit=FALSE;
}
for (i=0;i<n;i++)
    printf ("n Eccentricity of %d is %d\n',i,e[i]);
Min=e[0];
Max=e[0];
for(i=1;i<n;i++)
{
    if (e[i].max)
        max =e[i];
    if (e[i]<min)
        min =e[i];
}
radius =min;
diameter =max;
printf ("Radius = %d\n", radius);
printf ("Diameter = %d\n", diameter);
}

Void create (int n)
{
    Node *newl,*temp;
    printf ("Enter the elements of the matrix below: \n");
    for (i=0; i<n; i++)
    {
        tab [i] =(table*)malloc(size of (table));
        tab[i] -> visit = FALSE;
        tab[i] -> data = 'A'+i;
        tab[i] -> nodeptr = NULL;
        top = NULL;
        for (j=0;j<n;j++)
        {
for (i=0; i<n;i++)
Printf("is there is edge from %d to %d\n",i,j);
scanf("%d",&item)
if (item== 1)
{
    Printf("Enter the Weight\n");
    Scanf("%d",&w);
}
elserew= 0;
if(item)
{
    newl= (node*) malloc (size of (node));
    new l -> info =j;
    newl -> weight = w;
    newl -> link = NULL;
    if (temp)
        temp -> link =new;
    else
        tab[i] -> nodeptr = newl;
    temp = new;
}
}
for (i=0; i<n;i++)
{
    if (e[i] = = radius)
    {
        Center[i];
        Printf("Center Vertex is %d", center [i]);
    }
}Void dfs(int u)
{ 
    node *cur;
    int k;
    k = e[i]
    cur = tab[u] -> nodeptr;
    while(cur)
    {
        if (tab[cur -> info] -> visit = = FALSE)
        {
            E[i] +=cur -> weight;
            tab[cur ->info] -> visit = TURE;
            dfs(cur -> info);
            if (max <- e[i];
            {
                (max = e[i]);
            }
            e[i] = k;
            {
                cur = cur ->link;
            }
        }
    }
}

9. CONCLUSION:

An effort has been made in this paper to characterize the vertices of caterpillar graph and generate the sets with some properties. This kind of generation of sets with some properties may help us further in dealing with problems with different views. We can refer the books [9, 10 ] for the details of set theory and further check with different set operations that can be performed on the sets D, R and X and study the different properties exhibited by these operations on these sets.
REFERENCES: