Rubbling Number of Undirected Bipartite Graph with Impetus of Python Coding

Jitendra Binwal* and Aakanksha Baber**

*Professor and Head; **Research Scholar
Department of Mathematics, Mody University of Science and Technology,
Lakshmangarh-332311, Raj., India.

** Corresponding author's email: baberaakanksha8@gmail.com
doi: https://doi.org/10.21467/proceedings.100.1

ABSTRACT

A graph rubbling and pebbling move is removable of two pebbles and addition of one pebble with their different prospect of respect of vertices. Using rubbling and pebbling move, a vertex is reachable if a pebble can be moved to the vertex. In this paper, we demonstrate the evaluation of the rubbling number of bipartite graphs with the impetus of Python coding for larger number of vertices.

Keywords: Pebbling Number, Rubbling Number, Path Graph and Bipartite Graph.

1. Introduction

Chung was first one to introduce pebbling in graphs [1]. Consider, a graph of n vertices with fixed number of pebbles distributed among its vertices. Pebbling step is the removal of 2 pebbles from a vertex and addition of 1 pebble on its adjacent vertex $[2, 3, 4, 6]$.

The $f(G)$ is the representation of pebbling number of graphs, after a sequence of pebbling moves on a graph where pebbles are randomly distributed all over the vertices can sent 1 pebble to any arbitrary vertex. For an example, $f(G) = 5$.

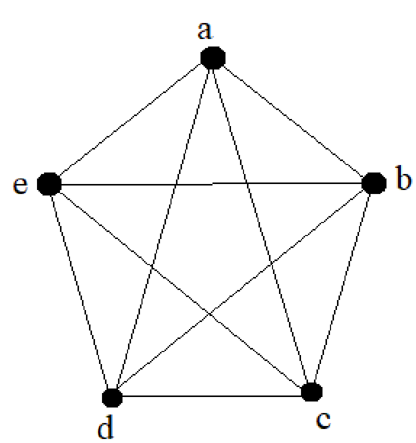


Figure 1. A undirected graph G with five vertices

1.1. Preposition [6]:

The pebbling number of the path graphs, $f(P_n) = 2^{n-1}$.

Proceedings DOI: 10.21467/proceedings.100; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-942709-6-6 (eBook)

1.2. Preposition [6]:

The pebbling number of the complete bipartite graphs,

$$f(K_{m,n}) = m + n$$

2. Graph Rubbling

Graph Rubbling is the way extension of graph pebbling. Sometimes, graph rubbling may be easier and harder than pebbling. Consider a graph, rubbling step is the removal of 1 pebble from each adjacent vertices of target vertex and placing 1 pebble on target vertex [6].

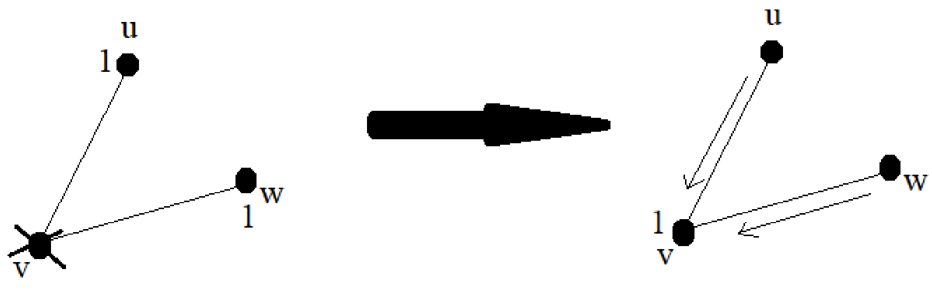


Figure 2. Rubbling Step on a graph

The $\rho(G)$ is the representation of rubbling number of graph G, after sequence of rubbling steps if target vertex is placed with 1 pebble from m number of pebbles distributed all over the graph. For an example, $\rho(G) = 4$.

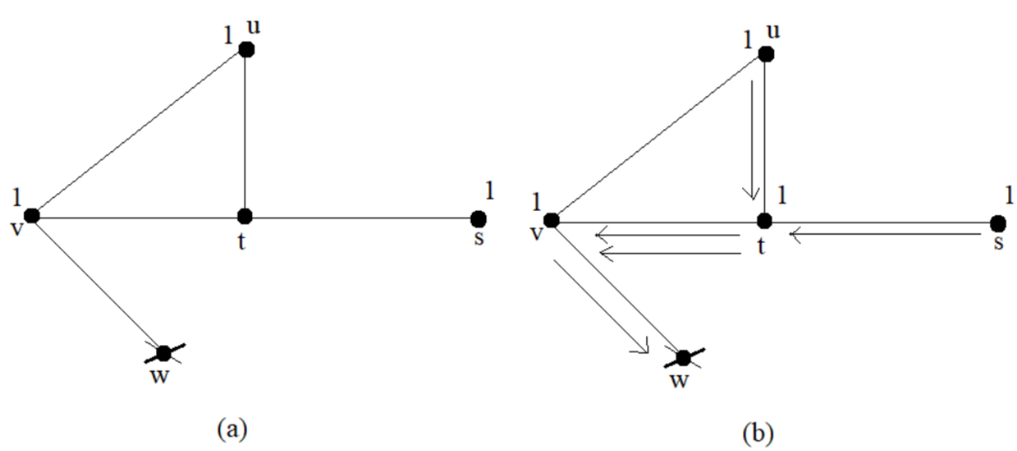


Figure 3. Rubbling number of a graph

Figure $3(a)$ shows pebble distribution of 3 pebbles where w vertex is not reachable. While, figure $3(b)$ shows that an extra pebble indicates w vertex reachable. Here, every vertex is reachable with 4 pebbles then, $\rho(G) = 4$.

2.1. Preposition [6]:

The rubbling number of the path graphs, $\rho(P_n) = 2^{n-1}$.

2.2. Preposition [6]:

The rubbling number of the complete bipartite graphs,

$$\rho(K_{m,n}) = 4$$

Proceedings DOI: 10.21467/proceedings.100

Series: AIJR Proceedings
ISBN: 978-81-942709-6-6
ISSN: 2582-3922

3. Main Theorem

Let k is the longest path in graph G that contains every vertex of G and $\bar{k}$ is the remaining vertices of G contains 1 pebble at each, which are not in k.

3.1. Conjecture [5]:

Let $G = (U, V, E)$ be an undirected bipartite graph and if there is at least one path containing every vertex of $|U|$ and $|V|$ of G then,

$$f(G) = 2^{s-1}, \text{ where } s = |U| + |V|.$$

3.2. Conjecture [5]:

If in $G = (U, V, E)$ undirected bipartite graph, there is not a path which contains all the vertices of G then, for $0 \leq \bar{k} \leq m, n$,

$$f(G) = 2^{k-1} + \bar{k}.$$

3.3. Conjecture:

On combining conjecture 3.1 and conjecture 3.2, it is clear that pebbling number of undirected bipartite graph, for $0 \leq \bar{k} \leq m, n$,

$$f(B_G) = 2^{k-1} + \bar{k}.$$

3.4. Main Theorem:

The rubbling number of undirected bipartite graph,

$$\rho(B_G) = 2^{k-1} + \bar{k} \qquad \text{for } 0 \leq \bar{k} \leq m, n$$

Proof: Case (1): In worst case scenario, when $\bar{k} = 0$.

Then, $\rho(B_G) = 2^{k-1}$, where $\mathrm{k} = |U| + |V|$.

From preposition 2.1. result is clear, rubbling is possible.

Case (2): When $\bar{k} \neq 0$ $i.e.$ $0 < \bar{k} \leq m, n$.

Using mathematical induction on $\bar{k}$.

Basic of Induction: when $\bar{k} = 1, 2$. Result is transparent.

Induction hypothesis: Suppose result is true for $(m - 2 + n)$.

Induction Test: To test whether $(m + n)$ holds result.

On combination of basic of induction and induction hypothesis, we can hold result true for $(m + n)$. Since, the graph is null graph, then each vertex will contain one pebble otherwise, rubbling concept is contradicted.

3.5. Open question: Let $G_D = (U, V, A)$ be a directed bipartite graph of arc A, if there is at least 1 path which contains every vertex of G_D i.e. $(U \cup V)$ then,

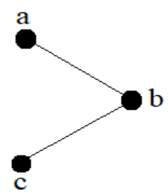
$$\rho(G_D) = 2^{s-1}, \quad s = |U| + |V|.$$

4. Python Coding

Let $G = (U, V, E)$ an undirected bipartite graph is converted into array in programming language.

1) Let the adjacency matrix of $G = (U, V, E)$ be denoted by $A(G)$.
2) Union of vertex set U and V is denoted by $(U \cup V)$.
3) Number of rows or number of columns are representation of number of vertices (i.e. $(U \cup V)$).
4) In python, length of matrix is the total number of vertices of graph G.

5) In python, a matrix is enclosed into a list, with the entry of 0 and 1 i.e. binary.
For an example,



$$A(G) = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 4. In Python, G is converted into $A(G)$ with the entry of 0 and 1 [7].

$$\therefore \quad A(G) = [[0,1,0],[1,0,1],[0,1,0]]$$

4.1. Coding for Rubbling number of undirected Bipartite Graph based on conjecture 3.1 [5,7]

Let RBG = Rubbling of Undirected Bipartite Graph.

```
>>> def power(x, s):
        res = 1
        for i in range(0, s):
            res = res * x
        return(res)
>>> def RBG(A):
        t = len(A)
        R = power(2, t-1)
        return(R)
>>> RBG([1,0,1])
4
>>> RBG([[0,1,0],[1,0,1],[0,1,0]])
4
```

For an example,

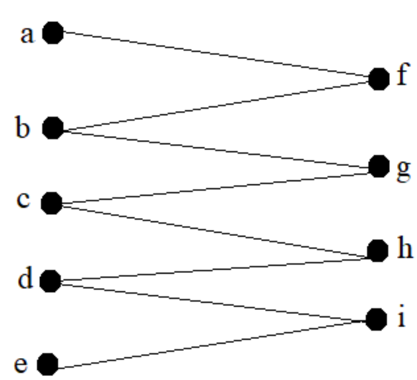


Figure 4.1.1. Bipartite Graph

```
>>> RBG(A)
256
```

Proceedings DOI: 10.21467/proceedings.100

ISBN: 978-81-942709-6-6 ISSN: 2582-3922

4.2. Coding for Pebbling number of undirected Bipartite Graph based on conjecture 3.1 $[5, 7]$

Let PBG = Pebbling of Undirected Bipartite Graph.

```
>>> def power(x, s):
        res = 1
        for i in range(0, s):
            res = res * x
        return(res)
>>> def PBG(B):
        t = len(B)
        P = power(2, t-1)
        return(P)
>>> PBG([[0,1],[1,0]])
2
```

4.3. Coding for Rubbling and Pebbling number of Undirected Bipartite Graph based on conjecture 3.1 and conjecture 3.2 $[5, 7]$

Let $G =$ total number of vertices in undirected bipartite graph and $m =$ number of vertices which are in the longest path of undirected bipartite graph.

```
>>> def power(x, s):
        res = 1
        for i in range(0, s):
            res = res * x
        return(res)
>>> def BG(G,m):
        n = G-m
        P = power(2,m-1) + n
        return(P)
>>> BG(122,3)
123
>>> BG(122,15)
16491
```

5. Concluding Remarks

Here, we have calculated the rubbling number of undirected bipartite graph with the help of Python coding. It is easy to use Python programming code to get better results. It describes the complexity of bipartite graph with the impetus of bipartite graph is widely used in the computational programming.

References

1. Deo N.: Graph Theory with Application to Engineering and Computer Science, Prentice Hall of India Private Ltd., 2001.
2. Chung F. R. K.: Pebbling in Hypercubes, SIAM J. Discrete Mathematics, 2(4), 1989, 467 − 472.
3. Gunda G.: Pebbling on Directed Graphs, University of Dayton, 2004.
4. Binwal J.: Equality of Pebbling Numbers on Directed Graphs, Journal of Indian Acad. Maths, 32(2), 2010.
5. Binwal J., Baber A.: Pebbling on Undirected Bipartite Graph, International Journal of Mathematical Sciences and Engineering Applications,12(2), 2018.
6. Sieben N.: Introduction to Graph Pebbling and Rubbling, Northern Arizona University, 2015.
7. Python 3.7 (IDLE) Programming Software.

Proceedings DOI: 10.21467/proceedings.100

Series: AIJR Proceedings
ISBN: 978-81-942709-6-6
ISSN: 2582-3922