ON PRIME LABELING OF SNAKE GRAPHS

Original Research Article

Abstract

A snake graph $C^m_{(k,q)}$ is the fusion of m number of $k-$cycles, C_k, such that, for $2 \leq i \leq m$, a shared vertex called the *vertebrae*, denoted by v_i, results from the fusion where a minimal path of length q joins $v_{(i-1)}$ and v_i. In the present study, we focus on a few questions on prime labeling of snake graphs as stated in Bigham et al. when each cycle of snake does not have the same k or q values and the minimum m for which $C^m_{(2,q)}$ when k is odd and greater than 2. To find the prime labeling for snake graphs for different k values, we label the graph using the modified snake labeling when $q = 2$. To obtain a possible snake labeling, we consider the modified snake labeling for $q = 2$, by considering all the possibilities. To find the maximum value of m, we implement a Python program to obtain the smallest prime factor of $k - 2$ for all odd numbers up to any given number. We obtain a general pattern for the maximum m value considering the above results.

Keywords: Cycles; Graphs; Prime Labeling; Snake Graphs.

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

One of the most active research areas in Graph theory is Graph labeling which is an assignment of integers to the vertices or edges, or both, subject to certain conditions. This concept was introduced by Alexander Rosa in 1967 [1]. Since then, it has gathered tremendous pace with a wealth of papers written on various types of labeling of graphs such as graceful, magic, and neighborhood, to name a few [2], [3]. In the present study, we focus specifically on prime labeling [4] on snake graphs [5]. In recent years, there has been a growing interest in the study of prime labeling of graphs [6], [7], which involves assigning distinct prime numbers to the vertices of a graph in such a way that the labels of any two adjacent vertices are relatively prime. One specific type of graph that has been the

focus of such investigations is the *snake graph*, which is formed by fusing together multiple cycles with a shared vertex. In our research, we build upon the work of Bigham et al. [8] to investigate prime labeling of snake graphs with varying k and q values, where k and q represent the cycle length and minimum path length, respectively. Specifically, our study is carried out to answer a couple of questions (out of five) raised in [8].

2 Preliminaries

In this section, terminologies of the types of graphs that we consider are presented as appeared in [8].

2.1 Prime Labeling

Definition 2.1. A graph has prime labeling if the vertices, of G can be labeled distinctly with the integers $1, 2, \ldots, n$, where n is the number of vertices such that any two adjacent vertex labels are relatively prime.

2.2 Cyclic Snakes

Definition 2.2. A snake graph $C^m_{(k,q)}$ is the fusion of m number of $k-$cycles, C_k, such that, for $2 \leq i \leq m$, a shared vertex called the *vertebrae*, denoted by v_i, results from the fusion where a minimal path of length q joins $v_{(i-1)}$ and v_i.

s^i_l is the length of the channel, where i is the cycle to which the spine vertex belongs and $l-$the distance between s^i_l and v_i , for all snakes, $1 \leq i \leq m$ and $1 \leq l \leq q-1$.

b^i_j is the belly of the snake graph and is the path of $(k-q)m$ edges from v_1 to $v_(m+1)$ that does not contain any vertices on the spine. Here $i-$ is the cycle to which the belly vertex belongs and $j-$ specifies the distance between b^i_j and v_i for all snakes, $1 \leq i \leq m$ and $1 \leq l \leq k-q-1$.

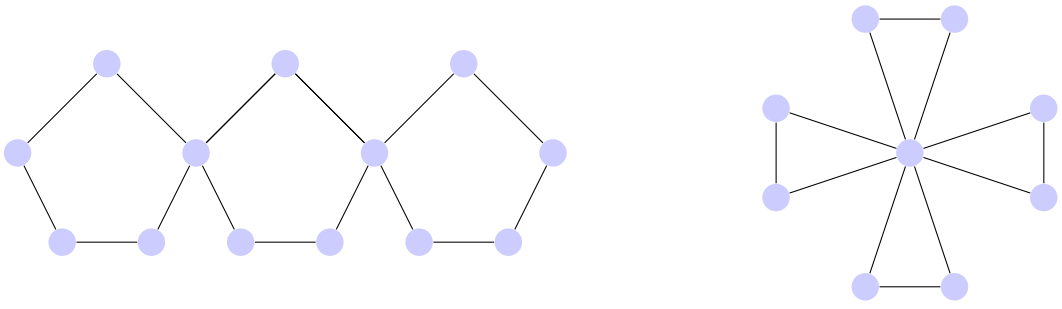

Figure 1: Two examples of snake graphs; $C^3_{5,2}$ and $C^4_{3,0}$ respectively.

For example, Figure 1 shows couple of snake graphs. Note that in $C^3_{5,2}$ there are $m = 3$ number of 5-cycles, connected by a minimal path of length $q = 2$. In $C^4_{3,0}$ or the complete graph of the fan graph we have $m = 4$ number of 3-cycles, connected by a minimal path of length $q = 0$.

Definition 2.3. A cyclic snake labeling is defined by the bijective mapping $f(x) : V(C_(k,q)^m) \longrightarrow 1, 2, \cdots, n$; for any vertex x in $C_(k,q)^m$ and for $1 \leq i \leq m+1,\ 1 \leq j \leq k-q+1,\ 1 \leq l \leq q-1$.

$$f(x) = \begin{cases} ik - i - k + 2 & \text{if } x = v_i, \\ ik - i + 1 - q + l & \text{if } x = s_l^i, \\ ik - i - k + j + 2 & \text{if } x = b_j^i, \end{cases} \tag{2.1}$$

If f results in a prime graph, then we say f is a cyclic snake prime labeling.

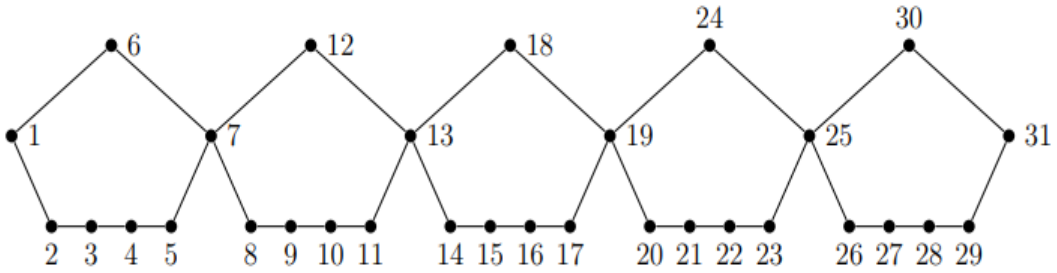

Figure 2: $C_{7,2}^5$ with the cyclic snake labelling $f(x)$.

2.3 Modified Snake Labeling

Definition 2.4. The modified cyclic snake labeling for snakes $C_{(k,q)}^m$ is defined by the following bijective function. For any vertex x in $C_{(k,q)}^m$ such that $1 \leq j \leq k-q-1$, and $1 \leq l \leq q-1$, if $gcd(m, k-2) = 1$, we label x by the original f labeling, so $F(x) = f(x)$. If $gcd(m, k-2) \neq 1$, we have

$$f(x) = \begin{cases} ik - i - k + 2 & \text{if } x = v_i, \\ ik - i - k + 3 & \text{if } x = s_1^i, \\ ik - i - k + 3 + j & \text{if } x = b_j^i, \end{cases} \tag{2.2}$$

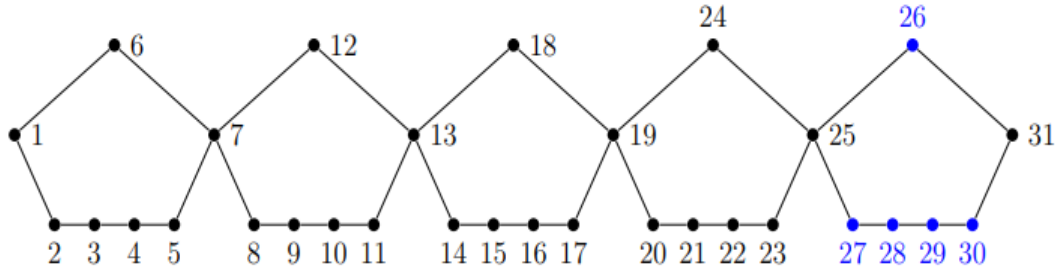

Figure 3: $C_{7,2}^5$ with the modified cyclic snake labelling $F(x)$.

By observing Figure 2 and Figure 3, the differences between $f(x)$ and $F(x)$ are highlighted in blue.

Theorem 2.1. *For every snake graph, $C^m_{(k,2)}$ for k is odd, the cyclic snake labeling is prime if and only if m is less than the smallest prime factor of $k-2$.*

3 Materials and Methods

Now, we will state the two questions as stated in [8] and followed those by providing the solutions.

Question 1 : Is there a predictable prime labeling for a snake graph such that each cycle of the snake does not necessarily have the same k or q values?

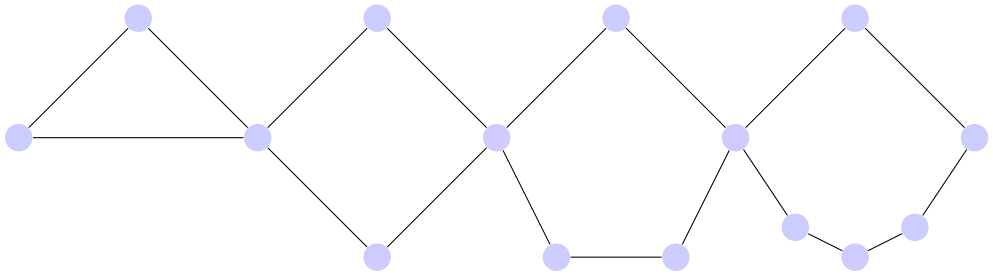

Figure 4: The snake graph with different k values.

Question 2 : What is the maximum m for which $C^m_{(k,2)},\ k$ is odd and greater than or equal to 3 has a cyclic snake prime labeling where either the spine vertices or belly vertices of each cycle are labeled first?

4 Results and Discussion

Now, we will provide a detailed study on the pendant number of line graphs, total graphs, and complement graphs as suggested in [8].

4.1 Prime Labeling for Snake Graph for different k values.

For Question 1, we can use the modified Snake Labeling because each cycle q is equal to 2. When spine vertices are labeled first and then labeled belly vertices of each circle.

Spine vertices:

$$k = 3 \qquad s_1^1 = 3 \times 1 - 1 - 3 + 3 = 2$$
$$k = 4 \qquad s_1^2 = 4 \times 2 - 2 - 4 + 3 = 5$$
$$k = 5 \qquad s_1^3 = 5 \times 3 - 3 - 5 + 3 = 10$$
$$k = 6 \qquad s_1^4 = 6 \times 4 - 4 - 6 + 3 = 17$$

Belly vertices:

$$
\begin{aligned}
k = 4 \quad & b_1^2 = 4 \times 2 - 2 - 4 + 3 + 1 = 6 \\
k = 5 \quad & b_1^3 = 5 \times 3 - 3 - 5 + 3 + 1 = 11 \\
& b_2^3 = 5 \times 3 - 3 - 5 + 3 + 2 = 12 \\
k = 6 \quad & b_1^4 = 6 \times 4 - 4 - 6 + 3 + 1 = 18 \\
& b_2^4 = 6 \times 4 - 4 - 6 + 3 + 2 = 19 \\
& b_3^4 = 6 \times 4 - 4 - 6 + 3 + 3 = 20
\end{aligned}
$$

Then we labele v_i as follows:

$$
\begin{aligned}
v_1 &= 1 \times 3 - 1 - 3 + 2 = 1 \\
v_2 &= 2 \times 4 - 2 - 4 + 2 = 4 \\
v_3 &= 3 \times 5 - 3 - 5 + 2 = 9 \\
v_4 &= 4 \times 5 - 4 - 5 + 2 = 13 \\
v_5 &= 5 \times 6 - 5 - 6 + 2 = 21
\end{aligned}
$$

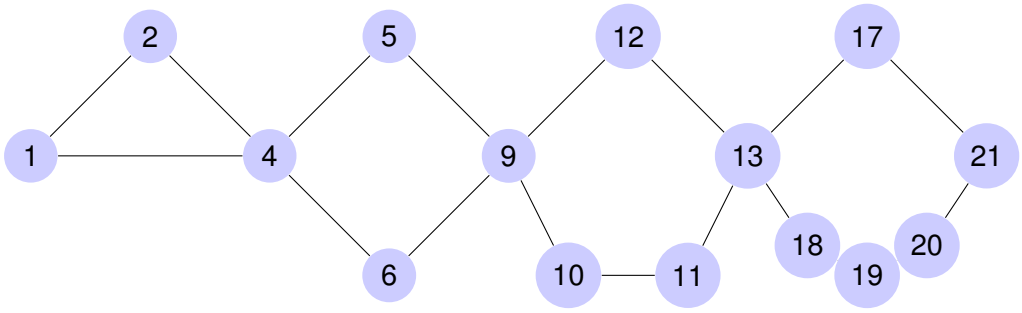

Figure 5: The labeled snake graph with different k values

4.2 Maximum value of m for Snake Graphs to have a cyclic snake prime labeling.

For Question 2, using Theorem 1 which states that when k is odd we can get a prime labeling for every snake graph $C^m_{(k,q)}$ if and only if m is less than the smallest prime factor of $k-2$ and the result obtaining by the python code to find a smallest prime factor of $k-2$ for each k values, we try to get a general pattern for m. We get the following results for m values:

1. If k is a prime number, then the max m is $k-2$.
2. max $m = 3$ when $k = 5 + 6n \; \forall \; n \in \mathbb{Z}^+$.
3. max $m = 5$ when $k = \begin{cases} 7 + 10n & \text{if } k \mod 3 \neq 0, \\ \text{otherwise} \end{cases} \forall \; n \in \mathbb{Z}^+$
4. For max $m \geq 7$ the given sequence does not have a clear pattern or a discernible mathematical rule. Therefore, it is not possible to determine the general formula for the sequence.

5 Conclusions

In Question 1, we have 15 vertices but we obtain 21 as a vertebra when labeling the graph using modified Snake Labeling for $q = 2$. Also, note that there are some integers missing, for instance $3, 7, 8, 14, 15$, and 16. When we consider all possibilities in order, the cycles $C^1_{4,2}$ and $C^1_{5,2}$ cannot be ordered as the 4th and 5th respectively as we would end up obtaining the same value as the vertebrae. Also, when we change the order of the cycles, again we end up obtaining the same value twice for the spine and belly vertices. Moreover, there is no predictable prime labeling for a snake graph that guarantees that each cycle of the snake has different k or q values. This is due to the fact that the prime labeling of a snake graph is dependent on the degrees of the vertices in the graph. If the degrees of the vertices in different cycles of the snake are different, then the prime labeling for each cycle will necessarily be different as well.

In general, the prime labeling of a snake graph is not unique, and many different types of prime labeling satisfy the underlying conditions. However, for a given snake graph, there may be a unique prime labeling that satisfies certain additional constraints or conditions. Therefore, it is not possible to predict prime labeling for a snake graph that guarantees different k or q values for each cycle of the snake. The prime labeling will depend on the specific degrees of the vertices in the graph, which can vary from cycle to cycle. Note that there is a possibility that if we consider the maximum m that we obtain in question 2 and add more cycles to Figure 3, the prime labeling of snake graphs can be obtained.

Furthermore, for Question 2, since we consider the smallest prime numbers, we also get a prime number for m. When max $m \geq 7$, the given sequence of numbers does not have an obvious or straightforward pattern. Therefore we modified the Python code to obtain the maximum m value when we were given the required k value.

References

[1] A. Rosa et al. On certain valuations of the vertices of a graph. In *Theory of Graphs (Internat. Symposium, Rome*, pages 349–355, 1966.

[2] A. Solairaju and K. Chithra. Edge-odd graceful graphs. *Electronic notes in discrete mathematics*, 33:15–20, 2009.

[3] S. A. Schluchter et al. Prime labelings of generalized petersen graphs. *Involve, a Journal of Mathematics*, 10(1):109–124, 2016.

[4] T. O. Dretsky et al. On vertex prime labeling of graphs in graph theory. *Combinatories and applications. Vol. IJ Alari (Wilky. NY 1991)*, pages 299–359.

[5] G. Gajalakshmi and S. Meena. On odd prime labelings of snake related graphs. *Journal of Algebraic Statistics*, 13(1):630–634, 2022.

[6] J. A. Gallian. A dynamic survey of graph labeling. *Electronic Journal of combinatorics*, 1(DynamicSurveys):DS6, 2018.

[7] M. A. Ollis. On prime labelings of uniform cycle snake graphs. *Emerson Authors, Researchers, Creators. 1203. https://digitalcommons.emerson.edu/arc/1203*.

[8] A. Bigham et al. Prime labelings of snake graphs. *PUMP Journal of Undergraduate research*, 2, 2019.

[9] L.R.M.K.R Jayathilaka. Prime-labeling-of-snake-graphs. https://github.com/Kavindya97/Prime-Labeling-of-Snake-Graphs.git, 2023.

[10] S. Molin and K. Jee. *Hands-On Data Analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization*. Packt Publishing Ltd, 2021.

6 Appendix

We have written code, which can be found in [9] that takes as input integer and computes the minimum prime factor. Here we mainly use pandas and math libraries [10] to compute the minimum prime factor and generate what is the max$m$ for a given odd integer.

```
n = int(input("Enter the range: "))

import pandas
import math
df = pandas.DataFrame(columns=['k_val', 'k-2_val', 'min_val'])


def pval(n):
    l = []
    for i in range(5, n+1):
        if i > 1:
            for j in range(2, i):
                if i % j == 0:
                    break
            else:
                l.append(i)
    return(l)

def odd_numbers(n):
    L = []
    for i in range(5, n+1):
        if i % 2 != 0:
            L.append(i)
    return L

def val(l):
    p = []
    for n in l:
        k = n-2
        p.append(k)
    return p


def ro(s):
    d = []
    for n in s:
        l = []
        while n % 2 == 0:
```

```
            l.append(2)
            n = n/2
        for i in range(3, int(math.sqrt(n))+1, 2):
            while n % i == 0:
                l.append(i)
                n = n/i
        if n > 2:
            l.append(int(n))
        if l == []:
            pass
        else:
            d.append(min(l))
    return d



c = odd_numbers(n)
s = val(c)
k = ro(s)

df['k_val'] = c
df['k-2_val'] = s
df['min_val'] = k

#df.to_excel("Oddvalue.xlsx")

def check_for_m(n):
    primes = []
    for i in range(2, n+1):
        for j in range(2, i):
            if i % j == 0:
                break
        else:
            primes.append(i)

    for prime in primes:
        if prime == 2:
            continue

        if prime > n:
            break

        if prime in pval(n):
            l = []
            for i in df['k_val'].where(df['min_val'] == prime).values:
                if str(i) != "nan":
                    l.append(i)
            if n in l:
                return "m="+ str(prime)
            else:
                continue
```

```
    return "not found"

x = int(input("What is the odd number: "))

while int(x) % 2 == 0:
    print("Error: Input must be an odd number.")
    x = input("Please re-enter the odd number: ")

def generate_numbers(n):
    nums = [6*k+5 for k in range(1, n+1)]
    nums2= [30*k+7 for k in range(1, n+1)]
    nums3= [84*k-5 for k in range(1, n+1)]

    if x in pval(n):
        print("m="+ str (x-2))
    elif x in nums:
        print( "m=3")
    elif x in nums2:
        print( "m=5")
    elif x in nums2:
        print( "m=7")
    else:
        print(check_for_m(x))

generate_numbers(n)
```