International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

EXPLORATION OF DECAGONAL GRACEFUL LABELING IN PATH GRAPHS WITH MUSICAL APPLICATIONS

R. Senthamizh Selvi [1†], J. Saral [2†], N. Sujatha [3*†]

[1,2,3] Department of Mathematics, SRM Institute of Science and Technology, Ramapuram, Chennai, 600 089, Tamil Nadu, India.

*Corresponding author(s). E-mail(s): sujathan@srmist.edu.in;

Contributing authors: sr9004@srmist.edu.in; saralj@srmist.edu.in;

† These authors contributed equally to this work.

Abstract

This study introduces a novel labeling technique called decagonal graceful labeling. Assume graph G is simple and finite, that has m edges and n vertices. The decagonal number denoted as D_m corresponds to the m^{th} decagonal number and a decagonal graceful labeling is defined by an injective function f that assigns the vertices in set $V(G)$ to the set $\{0, 1, 2, \dots, D_m\}$. A bijective function g is naturally induced on the edge set by this vertex labeling, which assigns labels from the set $\{1, 10, 27, \dots, D_m\}$ so that $g(uv) = |f(u) - f(v)|$ for each edge $uv \in E(G)$. A graph that allows this type of labeling is termed a decagonal graceful graph. We provide a technique for producing decagonal graceful labeling for a specific group of path-related graphs and show how this labeling may be extended to graphs with any number of vertices using Python code. We also investigate the notion of a melodic graph, demonstrating how these numbers might impact and motivate musical compositions by employing decagonal numbers as edge labels, exposing intriguing relationships between graph theory and music.

Keywords: Graph Labeling, Graceful Labeling, Decagonal numbers, Decagonal Graceful Labeling.

MSC Classification: 05C78

1 Introduction

The current research examines a finite undirected graph $G = (V(G), E(G))$ with no loops or multiple edges. The variables m and n denote the size and order of G respectively.

Chartrand and Lesinak [1] are cited for terms related to graph theory. A labeling of a graph G is a projection where integers are assigned to the vertices, edges or both subject to certain limitations.Vertex, edge or total labeling are terms used to describe the labeling, depending on whether the mapping's domain is $V(G), E(G)$ or $V(G) \cup E(G)$.

The graceful labeling approach was initially proposed by Alexander Rosa [2] in 1967. Later on, Golomb [3] named it graceful labeling, although it was originally known as β - labeling. The graceful labeling of a graph G with m edges occurs if each vertex assigns a unique non-negative integer label and the absolute differences between the labels of adjacent vertices fall between 1 and m.

Received: August 08, 2025

Major findings and theories about graceful labeling, including a well-known theory regarding the graceful labeling of trees were presented by Chartrand et al. [5]. Well known and unproven is the graceful tree conjecture, which states that all trees are graceful. Nonetheless, the concept of graceful labeling was heavily used in decomposing the complete graphs into isomorphic subgraphs.

With successful applications to a variety of graph classes, including trees with up to 40 vertices (Eshghi and Azimi [6]), mathematical programming techniques have been developed to represent and solve graceful labeling problems. Michael J. Reynolds and Ljiljana Brankovic [7] provided the major computer search findings for identifying graceful labeling of trees. It is observed that every path, caterpillar, banana tree and olive tree are graceful. Certain lobsters are additionally renowned for their gracefulness, most notably fire crackers. Trees that are symmetrical, have fewer than eight leaves on their diameter and have fewer than five leaves have been considered to be graceful. An important advancement in the field of graceful tree labeling is the proof by Aldred and McKay [8] that any tree with a maximum of 27 vertices is graceful. All trees with 29 vertices were gracefully labeled with the help of an edge search technique invented by Michael Horton [9].

Deise L. de Oliveira et.al. [10] suggested strategies for winning for Alice and Bob in the graceful game based on the following graph classes: fan graphs, wheels, prisms, hypercubes, 2-powers of paths, caterpillars, trees, gear graphs, web graphs, cycles, complete bipartite graphs and paths.

Graph theory has emerged as a valuable tool in cryptography, offering innovative approaches to encryption and decryption. Auparajita Krishnaa [11] explored algorithms from graph theory such as tree traversal and graph labeling are used in cryptography to create highly hidden ciphertexts. Further it explores the use of various graph theory concepts and algorithms such as tree traversal algorithms, Kruskal's algorithm and a modified graph labeling scheme to develop encryption methods that can effectively hide the original plaintext and create highly secure ciphertexts. Perera and Wijesiri [12] proposed an alternative method in symmetric key cryptography, where graphs were transformed into matrices to serve as secret keys, enabling the generation of multiple ciphertexts from plaintext graphs. Shruthy and Maheswari [13] developed an encryption process combining Trifid Cipher with graceful labeling, utilizing Tom-Tom Code for graph labeling clues in decryption.

Vaidya and Vihol [14] investigated Fibonacci and super Fibonacci graceful labeling for cycle-related graphs, demonstrating that path unions of certain cycles are Fibonacci graceful. The concepts of super Fibonacci graceful anti magic graphs were first presented by Amuthavalli and Shanmuga Sundaram [15], who also proved that the cherry blossom flower graph, clematis flower graph and rose flower graph are examples of such graphs. The super Fibonacci graceful anti-magic labeling of such graphs was also coded in Python by them.

Lina and Asha [16] introduced odd triangular graceful labeling, where vertices are labeled with triangular numbers and edge weights form an odd sequence. Akshaya and Asha [17] extended this concept to even triangular graceful labeling, applying it to special graphs like multi-star and Golomb graphs.

Lalitha and Tamilselvi [18] introduced graceful labeling for hexagonal snakes arranged along a path, demonstrating that it satisfies α-labeling.

Branson and Sutton [19] analyzed the running time of an evolutionary algorithm for finding graceful labeling of paths, stars and complete bipartite graphs with a constant sized partition. Additionally, they compared the performance of their evolutionary algorithm to a complete constraint solver.

Nikson Simarmata et.al. [20] investigated a new approach to construct a graceful labeling for a super-rooted tree graph, which is rooted tree constructed from several rooted trees using adjacency matrices. Many families of graphs such as paths, combs, star graphs, subdivisions of stars, bistars, $S_{m,n,r}$ graphs and coconut trees, that admit polygonal sum labeling were characterized by Sureshkumar and Murugesan [21].

According to research by Anick [22], the number of graceful labelings on trees increases super exponentially with the number of edges. Additionally, the number of graceful labelings divided by the overall number of labelings exhibits an exponential function with a base of approximately 1.575. Furthermore, he proved that the logarithm of the number of graceful labelings of a tree can be accurately predicted with only five graph invariants. Kaneria and Meghpara [23] established graceful labeling for one point union for path of graphs.

Mahendran and Murugan [33] demonstrated that some graphs are pentagonal graceful. It is noted that caterpillar $P_{n-1}(1,2,\ldots,n), P_n \odot K_1$, twig graph, shrub tree, banana tree, $K_{1,n} \odot K_1$, F-tree FP_n for $n \geq 3$ and Y-tree are pentagonal graceful labeling. In addition, pentagonal graceful graphs are generated from the star $K_{1,n}$ by splitting the edges of the star for all $n \geq 1$ and from $P_n \odot K_1$ by dividing the edges of the path P_n for all $n \geq 2$. Furthermore, pentagonal graceful graphs are created when the leaves of $K_{1,n}$ are joined with the central vertex of $K_{1,n}$ for all $n \geq 1$, as well as when the pendant vertex of P_m is attached with a leaf of $K_{1,n}$ for all $n \geq 1, m \geq 2$. This shows that pentagonal graceful labeling is not limited to specific types of graphs and can be observed in various graph structures.

The n-star graph is a centred hexagonal graceful graph as demonstrated by Ramalakshmi and Syed Ali Nisaya [34].

The graphs obtained by splitting the edges of star $K_{1,n}$ for all $n \geq 1$ and $P_n \odot K_1$ as well as by identifying the leaves of $K_{1,n}$ with the central vertex of $K_{1,2}$ and F - tree FP_n for all $n \geq 3$ are all heptagonal graceful graphs, according to Akshaya and Asha [35].

Mahendran [36] showed that some graphs admits octogonal graceful labeling. It was observed that caterpillar, $P_{n-1}(1,2,\ldots,n), P_n \odot K_1$, twig graph, shrub tree, banana tree, $K_{1,n} \odot K_1, F$ - tree FP_n for $n \geq 3$ and Y-tree were octagonal graceful labeling. He presented several families of octagonal graceful graphs. One of these families is obtained by splitting the graphs from star $K_{1,n}$ for every $n \geq 1$ and the graphs from $P_n \odot K_1$ by subdividing the edges of the path P_n for all $n \geq 2$. Additionally, they showed that the graphs produced for all $n \geq 1, m \geq 2$ by

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

connecting the central vertex of $K_{1,2}$ with the leaves of $K_{1,n}$ and the graph produced for all $n \geq 1, m \geq 2$ by connecting the pendant vertex of P_m with a leaf of $K_{1,n}$ are both octagonal graceful.

Selvarani and Arthy [26] provided an overview of the application of graph theory in various domains, but they mostly focused on computer science applications.

Ongoing research continues to explore the gracefulness of various graph classes and develop new techniques for solving graceful labeling problems. The basic notion of graceful labeling in graph theory has shaped the evolution of graph labeling as a topic of study immensely. The identification of graphs with graceful labeling and the relationship between graceful labeling and graph decompositions have been the main topics of much graceful labeling research.

Applications of the graceful labeling problem includes circuit layout optimisation, communications networks and coding theory. Overall, graph-based representations offer a versatile and promising approach for analyzing and comparing symbolic music data across various Music Information Retrieval (MIR) applications.

Decagonal graceful labeling is a new labeling technique that we present in this study. Consider the graph G, which has m edges and n vertices. Decagonal graceful labeling is defined by an injective function f that corresponds the vertex set $V(G)$ to the set $\{0,1,2,\dots,D_m\}$, where D_m is the m^{th} decagonal number. This vertex labeling naturally produces a bijective function g, such that for each edge $uv \in E(G)$, $g(uv) = |f(u) - f(v)|$ and subsequent edge labels are assigned from the set $\{1, 10, 27, \dots, D_m\}$. A graph which admits decagonal graceful labeling is called a decagonal graceful graph.

For certain classes of path-related graphs, we give an approach for producing decagonal graceful labelings and we show how this labeling may be extended to graphs with an arbitrary number of vertices using Python code. Furthermore, by utilizing decagonal numbers as edge labels, we investigate the concept of a melodic graph and demonstrate how these numbers might impact and stimulate musical compositions, exposing intriguing links between graph theory and music.

2 Overview of the paper structure

The below Figure 1 represents the overview of the paper structure. It includes the following contents such as introduction, highlights of current study, preliminaries, main results, applications and conclusion.

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

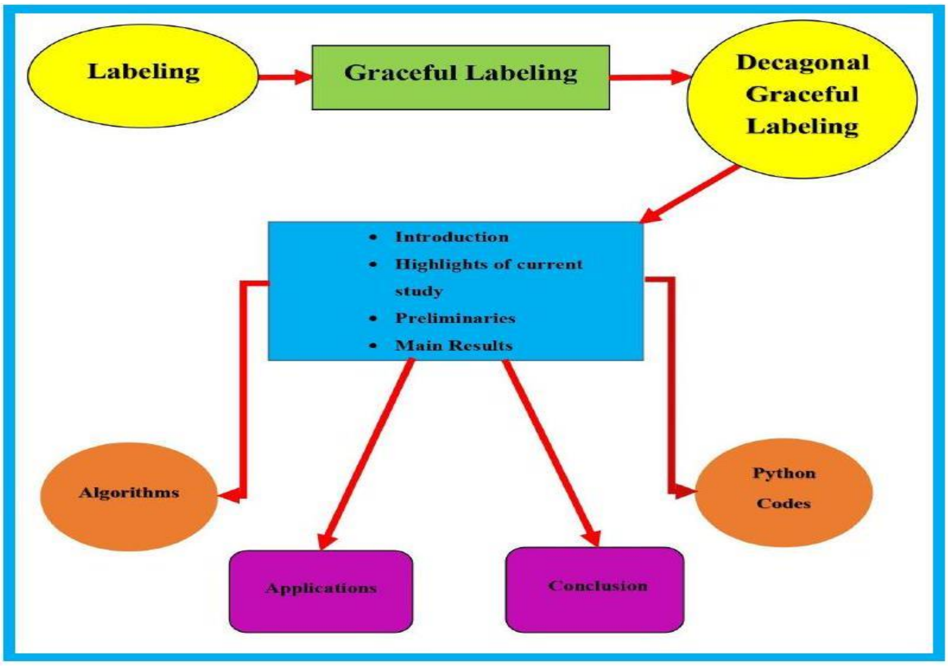


Fig. 1 Overview of the paper structure

3 Highlights of current study

The following Table 1 shows the various studies which is related to the current study. The list of table contents are Title and Author, Year of Publication and a brief description about Main findings. In the last row of table content, we have provided about the main findings of this study. These papers shows variety of graceful labeling methods applied to several types of graphs, demonstrating the increasing relevance and interest in this field of mathematical research.

Table 1 Highlights

Title & Author	Year	Main findings
Centered Hexagonal Graceful Labeling of n -star Graph, Ramalakshmi and Syed Ali Nisaya [34]	2019	The authors investigated the centered hexagonal graceful labeling of n-star graph.
Pentagonal Graceful Labeling of Some Graphs, Mahendran and Murugan [33]	2021	The authors demonstrated the constructions of pentagonal graceful labeling for some families of graphs.
Octagonal Graceful Labeling of Some Special Graphs, Mahendran [36]	2021	The authors investigated the octagonal graceful labeling of several graphs in this study.

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Heptagonal Graceful Labeling on Simple Graphs, Akshaya and Asha [35]	2022	The authors examined the heptagonal graceful labeling of several graphs in this study.
Decagonal Graceful Labeling using Python: A Computational Approach, Senthamizh Selvi et. al	Yet to Publish	The following are the paper's key findings: - Decagonal graceful labelings of several graphs related to paths are determined. - Python code is written to build decagonal graceful labels for graphs connected by paths. - This Python code allowed us to verify the labeling accuracy. - We also presented the notion using a real-world application, notably melodic graphs.

4 Preliminaries

Definition 1. A triple graph G consists of an edge set $E(G)$, a vertex set $V(G)$ and a relation that identifies two vertices (which need not be different) for each edge known as its endpoints.

Definition 2. A graph that excludes loops and multiple edges is said to be simple.

Definition 3. Assigning labels to vertices or edges of a graph while meeting specific requirements or characteristics is known as graph labeling.

Definition 4. Graceful labeling of a graph G with m number of edges involves assigning each vertex of the graph a unique, non-negative integer label so that the absolute differences between the labels of neighbouring vertices are all distinct and fall from 1 to m.

Definition 5. A path is a basic network with vertices that can be arranged in a way that makes two vertices adjacent only if they appear consecutively in the list. A path with n vertices is represented by P_n.

Definition 6. By joining exactly two pendant vertices to the vertices $n-1$ and n of P_n, a path P_n can be utilized to construct a F-tree on $n+2$ vertices, or FP_n.

Definition 7. A Y-tree on $n+1$ vertices, denoted by Y_n, is constructed from a path P_n by attaching exactly one pendant vertex to the $(n-1)^{\text{th}}$ vertex of P_n.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Definition 8. A star graph is a type of tree with one central node connected directly to all other nodes, but those other nodes are not connected to each other. A star graph is denoted by $K_{1,n}$ and it has $n+1$ vertices.

Definition 9. Decagonal numbers are those that have the form $4m^2-3m$ for all $m \geq 1$. Decagonal numbers starting at $1, 10, 27, 52, 85, 126, 175, 232, 297, 370$ and so on.

Definition 10. With m edges and n vertices, let us consider the graph G. An injective function f, where D_m is the m^{th} decagonal number, defines decagonal graceful labeling by mapping the vertex set $V(G)$ to the set $\{0,1,2,\ldots,D_m\}$. A bijective function g is generated by this vertex labeling, such that for every edge $uv \in E(G)$, $g(uv)=|f(u)-f(v)|$. Subsequent edge labels are assigned from the set $\{1,10,27,\ldots,D_m\}$. Decagonal graceful labeling is a function f that admits unique sequential decagonal numbers $\{D_1, D_2, \ldots, D_m\}$. A graph that allows for this kind of labeling is referred to as a decagonal graceful graph.

5 Remark

The analytical validity of edge label uniqueness is established for graphs that meet the decagonal graceful requirements for $n \geq 2$. For arbitrary graphs, computational verification of uniqueness might be required.

6 Main Results

Theorem 1. Let $G = P_n$. Then G admits decagonal graceful labeling for $n \geq 2$.
Proof. Let $V(G)=\{v_1, v_2, \ldots, v_n\}$ and $E(G)=\{v_i v_{i+1} : 1 \leq i \leq n-1\}$ represent the vertex set and edge set of G. Since $|V(G)|=n$ and $|E(G)|=n-1$, we define the function $f: V(G) \to \{0, 1, 2, \ldots, D_{n-1}\}$ as follows:

$$f(v_n)=0$$

Case (i): i is an even number and i is bounded by 1 and $n-1$, including both end points.

$$f(v_{n-i})=\frac{i(8n-4i-7)}{2}$$

Case (ii): i is an odd number and i is bounded by 1 and $n-1$, including both end points.

$$f(v_{n-i})=\frac{8n^2-14n+4i^2+7i-8in+3}{2}$$

Thus, the resulting edge labels are distinct decagonal numbers, ensuring that G possesses a decagonal graceful labeling for $n \geq 2$. □

The following Figure 2 gives the decagonal graceful labeling of $G = P_6$.

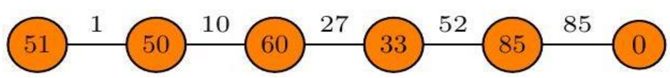


Fig. 2 Path P_6

The following Algorithm 1 gives the decagonal graceful labeling of $G = P_n$ for $n \geq 2$.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Algorithm 1 Algorithm to obtain decagonal graceful labeling of Path graph P_n, for $n \geq 2$.
Input: Set of vertices n.
Output: Set of vertex labels

1: $v \leftarrow \emptyset$;

2: ***for*** *i* ∈ (0, *n*) ***do***

3: ***if*** *i is even* ***then***

4: $v \leftarrow v \cup \left\{\frac{i(8n-4i-7)}{2}\right\}$

5: ***else***

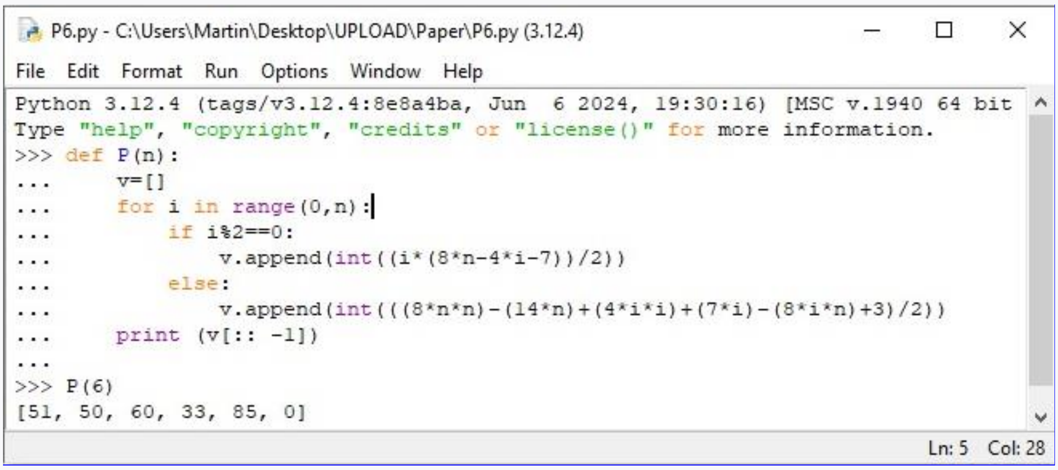
6: $v \leftarrow v \cup \left\{\frac{8n^2-14n+4i^2+7i-8in+3}{2}\right\}$

7: ***end if***

8: ***end for***

9: ***return*** *reverse v*

The following Figure 3 shows that how to obtain decagonal graceful labeling of $G = P_n$ for $n \geq 2$ using Python code.



```
P6.py - C:\Users\Martin\Desktop\UPLOAD\Paper\P6.py (3.12.4)
File  Edit  Format  Run  Options  Window  Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit
Type "help", "copyright", "credits" or "license()" for more information.
>>> def P(n):
...     v=[]
...     for i in range(0,n):
...         if i%2==0:
...             v.append(int((i*(8*n-4*i-7))/2))
...         else:
...             v.append(int(((8*n*n)-(14*n)+(4*i*i)+(7*i)-(8*i*n)+3)/2))
...     print (v[:: -1])
...
>>> P(6)
[51, 50, 60, 33, 85, 0]
Ln: 5  Col: 28
```

Fig. 3 Python code to receive decagonal graceful labeling of $G = P_n$ for $n \geq 2$.

From Table 2, we can observe the following:

- The columns in Table 2 indicate the number of vertices in P_n.
- Each row corresponds to a specific vertex of P_n.

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- The values inside Table 2 represent the labels assigned to the respective vertices.

Table 2 Decagonal graceful labeling for $G = P_n$ up to 20 vertices.

Number of Vertices	Vertex Labels of Pn																			
	n	n-1	n-2	n-3	n-4	…	…	…	…	…	…	…	…	…	…	…	…	3	2	1
2	0	1																		
3	0	10	9																	
4	0	27	17	18																
5	0	52	25	35	34															
6	0	85	33	60	50	51														
7	0	126	41	93	66	76	75													
8	0	175	49	134	82	109	99	100												
9	0	232	57	183	98	150	123	133	132											
10	0	297	65	240	114	199	147	174	164	165										
11	0	370	73	305	130	256	171	223	196	206	205									
12	0	451	81	378	146	321	195	280	228	255	245	246								
13	0	540	89	459	162	394	219	345	260	312	285	295	294							
14	0	637	97	548	178	475	243	418	292	377	325	352	342	343						
15	0	742	105	645	194	564	267	499	324	450	365	417	390	400	399					
16	0	855	113	750	210	661	291	588	356	531	405	490	438	465	455	456				
17	0	976	121	863	226	766	315	685	388	620	445	571	486	538	511	521	520			
18	0	1105	129	984	242	879	339	790	420	717	485	660	534	619	567	594	584	585		
19	0	1242	137	1113	258	1000	363	903	452	822	525	757	582	708	623	675	648	658	657	
20	0	1387	145	1250	274	1129	387	1024	484	935	565	862	630	805	679	764	712	739	729	730

Theorem 2. Let $G = FP_n$. Then G admits decagonal graceful labeling for $n \geq 3$.
Proof. Let the vertex and edge sets of G be represented as $V(G) = \{v_1, v_2, \dots, v_n, u, v\}$ and $E(G) = \{v_i v_{i+1}: 1 \leq i \leq n-1\} \cup \{uv_n, vv_{n-1}\}$. Since $|V(G)| = n+2$ and $|E(G)| = n+1$, we define the function $f: V(G) \to \{0, 1, 2, \dots, D_{n+1}\}$ as follows:

$$f(u) = 1$$
$$f(v) = 4n^2 + 5n - 9$$
$$f(v_n) = 0$$
$$f(v_{n-1}) = 4n^2 + 5n + 1$$
$$f(v_{n-2}) = 8n + 1$$

Case (i): j is an even number and j is bounded by 4 and $n-1$ including both end points.

$$f(v_{n-j}) = \frac{8nj - 4j^2 + 9j}{2}$$

Case (ii): j is an odd number and j is bounded by 3 and $n-1$ including both end points.

$$f(v_{n-j}) = \frac{8n^2 - 8nj + 18n + 4j^2 - 9j + 7}{2}$$

Thus, the resulting edge labels are distinct decagonal numbers, ensuring that G possesses a decagonal graceful labeling for $n \geq 3$. □

The following Figure 4 gives the decagonal graceful labeling of $G = FP_7$.

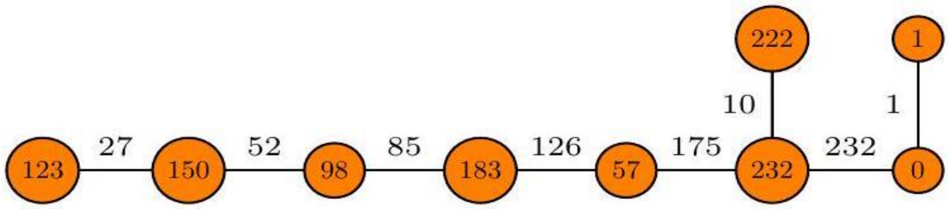


Fig. 4 FP_7

The following Algorithm 2 gives the decagonal graceful labeling of $G = FP_n$ for any $n \geq 3$.

Algorithm 2 Algorithm to obtain decagonal graceful labeling of FP_n, for $n \geq 3$.
Input: Set of vertices n.
Output: Set of vertex labels

1: $v \leftarrow \emptyset$;

2: ***for*** $i \in (0, n)$ ***do***

3: ***if*** $i = n - 1$ ***then***

4: $v \leftarrow v \cup \{4n^2 + 5n - 9\}$

5: $v \leftarrow v \cup \{1\}$

6: $v \leftarrow v \cup \{0\}$

7: ***for*** $i \in (1, n)$ ***do***

8: ***if*** *j is even* ***then***

9: $v \leftarrow v \cup \left\{\frac{8nj-4j^2+9j}{2}\right\}$

10: ***else***

11: $v \leftarrow v \cup \left\{\frac{8n^2-8nj+18n+4j^2-9j+7}{2}\right\}$

12: ***end if***

13: ***end for***

14: ***end if***

15: ***end for***

16: ***return*** *reverse v*

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The following Figure 5 shows that how to obtain decagonal graceful labeling of $G = FP_n$ for $n \geq 3$ using Python code.

```
IDLE Shell 3.12.4
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def FP(n):
...     v=[]
...     for i in range(0,n):
...         if i==n-1:
...             v.append(int((4*n*n+5*n-9)))
...             v.append(1)
...             v.append(0)
...     for j in range(1,n):
...         if j%2==0:
...             v.append(int((8*n*j-4*j*j+9*j)/2))
...         else:
...             v.append(int((8*n*n-8*n*j+18*n+4*j*j-9*j+7)/2))
...     print(v[::-1])
...
>>> FP(7)
[123, 150, 98, 183, 57, 232, 0, 1, 222]
>>>
Ln: 19  Col: 0
```

Fig. 5 Python code to receive decagonal graceful labeling of $G = FP_n$ for $n \geq 3$.

From Table 3, we can observe the following:

- The columns in Table 3 indicate the number of vertices in FP_n.
- Each row corresponds to a specific vertex of FP_n.
- The values inside Table 3 represent the labels assigned to the respective vertices.

Table 3 Decagonal graceful labeling for $G = FP_n$ up to 20 vertices.

Number of Vertices	Vertex Labels of FPn																					
	n	n-1	n-2	n-3	n-4	…	…	…	…	…	…	…	…	…	…	…	…	3	2	1	u	v
3	0	52	25																		1	42
4	0	85	33	60																	1	75
5	0	126	41	93	66																1	116
6	0	175	49	134	82	109															1	165
7	0	232	57	183	98	150	123														1	222
8	0	297	65	240	114	199	147	174													1	287
9	0	370	73	305	130	256	171	223	196												1	360
10	0	451	81	378	146	321	195	280	228	255											1	441
11	0	540	89	459	162	394	219	345	260	312	285										1	530
12	0	637	97	548	178	475	243	418	292	377	325	352									1	627
13	0	742	105	645	194	564	267	499	324	450	365	417	390								1	732
14	0	855	113	750	210	661	291	588	356	531	405	490	438	465							1	845
15	0	976	121	863	226	766	315	685	388	620	445	571	486	538	511						1	966
16	0	1105	129	984	242	879	339	790	420	717	485	660	534	619	567	594					1	1095
17	0	1242	137	1113	258	1000	363	903	452	822	525	757	582	708	623	675	648				1	1232
18	0	1387	145	1250	274	1129	387	1024	484	935	565	862	630	805	679	764	712	739			1	1377
19	0	1540	153	1395	290	1266	411	1153	516	1056	605	975	678	910	735	861	776	828	801		1	1530
20	0	1701	161	1548	306	1411	435	1290	548	1185	645	1096	726	1023	791	966	840	925	873	900	1	1691

Theorem 3. Let $G = Y_n$. Then G admits decagonal graceful labeling for $n \geq 3$.

Proof. Let the vertex set and edge set of G be represented as $V(G) = \{v_1, v_2, \dots, v_n, u\}$ and $E(G) = \{v_i v_{i+1} : 1 \leq i \leq n-1\} \cup \{uv_n\}$. Since $|V(G)| = n+1$ and $|E(G)| = n$, we define the function $f : V(G) \to \{0, 1, 2, \dots, D_n\}$ as follows:

$$f(u) = 10$$
$$f(v_n) = 1$$
$$f(v_{n-1}) = 0$$
$$f(v_{n-2}) = 4n^2 - 3n$$

Case (i): j is an even number

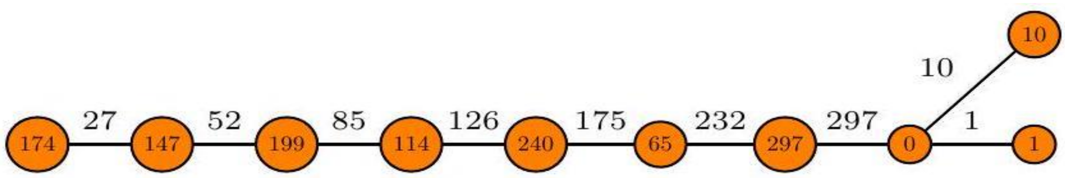
$$f(v_{n-j}) = \frac{8n^2 - 8nj + 10n + 4j^2 - 9j + 2}{2}$$

Case (ii): j is an odd number

$$f(v_{n-j}) = \frac{8nj - 8n - 4j^2 + 9j - 5}{2}$$

Thus, the resulting edge labels are distinct decagonal numbers, ensuring that G possesses a decagonal graceful labeling for $n \geq 3$. □

The following Figure 6 gives the decagonal graceful labeling of $G = Y_9$.



International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Fig. 6 Y_9

The following Algorithm 3 gives the decagonal graceful labeling of $G = Y_n$ for any $n \geq 3$.

Algorithm 3 Algorithm to obtain decagonal graceful labeling of Y_n, for $n \geq 3$.
Input: Set of vertices n.
Output: Set of vertex labels

1: $v \leftarrow \emptyset$;

2: ***for*** $i \in (0, n)$ ***do***

3: ***if*** $i = n - 1$ ***then***

4: $v \leftarrow v \cup \{10\}$

5: $v \leftarrow v \cup \{1\}$

6: ***for*** $j \in (1, n)$ ***do***

7: ***if*** $i = n - 1$ ***then***

8: $v \leftarrow v \cup \left\{\frac{8n^2-8nj+10n+4j^2-9j+2}{2}\right\}$

9: ***else***

10: $v \leftarrow v \cup \left\{\frac{8nj-8n-4j^2+9j-5}{2}\right\}$

11: ***end if***

12: ***end for***

13: ***end if***

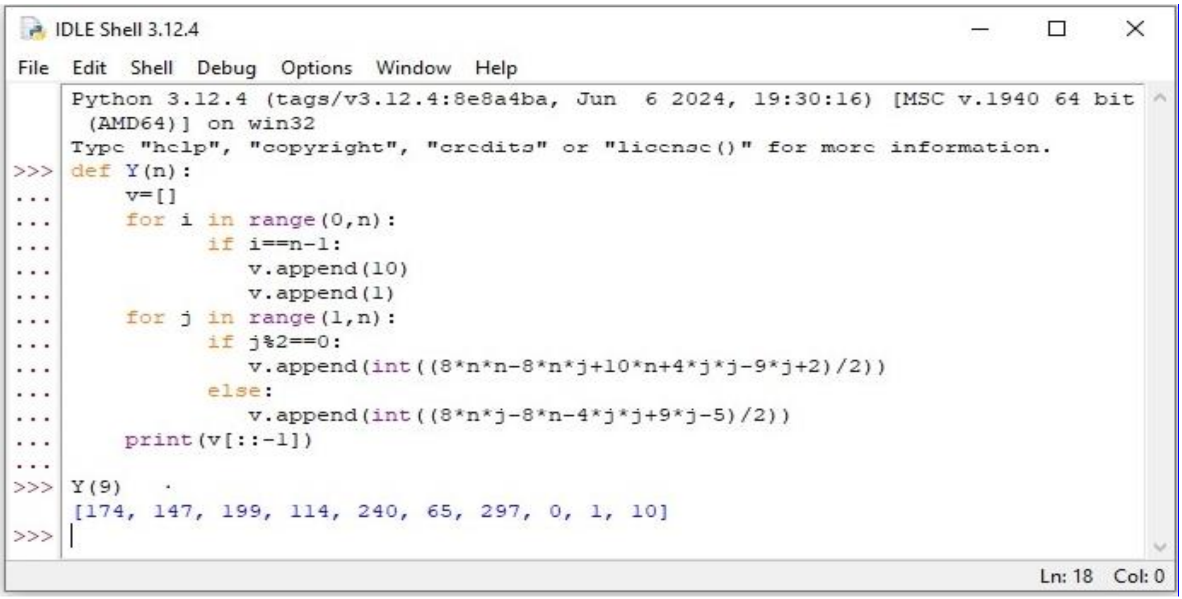
14: ***end for***

15: ***return*** *reverse v*

The following Figure 7 gives that Python code of the decagonal graceful labeling of $G = Y_n$ and from Tabel 4, we can observe the following:

- The columns in Table 4 indicate the number of vertices in Y_n.
- Each row corresponds to a specific vertex of Y_n.
- The values inside Table 4 represent the labels assigned to the respective vertices.

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



```
IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit
 (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def Y(n):
...     v=[]
...     for i in range(0,n):
...             if i==n-1:
...                 v.append(10)
...                 v.append(1)
...     for j in range(1,n):
...             if j%2==0:
...                 v.append(int((8*n*n-8*n*j+10*n+4*j*j-9*j+2)/2))
...             else:
...                 v.append(int((8*n*j-8*n-4*j*j+9*j-5)/2))
...     print(v[::-1])
...
>>> Y(9)
[174, 147, 199, 114, 240, 65, 297, 0, 1, 10]
>>>
Ln: 18  Col: 0
```

Fig. 7 Python code to receive decagonal graceful labeling of $G = Y_n$ for $n \geq 3$.

Table 4 Decagonal graceful labeling for $G = Y_n$ up to 20 vertices.

Number of Vertices	Vertex Labels of Yn																				
	n	n-1	n-2	n-3	n-4	…	…	…	…	…	…	…	…	…	…	…	…	3	2	1	u
3	1	0	27																		10
4	1	0	52	25																	10
5	1	0	85	33	60																10
6	1	0	126	41	93	66															10
7	1	0	175	49	134	82	109														10
8	1	0	232	57	183	98	150	123													10
9	1	0	297	65	240	114	199	147	174												10
10	1	0	370	73	305	130	256	171	223	196											10
11	1	0	451	81	378	146	321	195	280	228	255										10
12	1	0	540	89	459	162	394	219	345	260	312	285									10
13	1	0	637	97	548	178	475	243	418	292	377	325	352								10
14	1	0	742	105	645	194	564	267	499	324	450	365	417	390							10
15	1	0	855	113	750	210	661	291	588	356	531	405	490	438	465						10
16	1	0	976	121	863	226	766	315	685	388	620	445	571	486	538	511					10
17	1	0	1105	129	984	242	879	339	790	420	717	485	660	534	619	567	594				10
18	1	0	1242	137	1113	258	1000	363	903	452	822	525	757	582	708	623	675	648			10
19	1	0	1387	145	1250	274	1129	387	1024	484	935	565	862	630	805	679	764	712	739		10
20	1	0	1540	153	1395	290	1266	411	1153	516	1056	605	975	678	910	735	861	776	828	801	10

Theorem 4. Let $G = K_{1,n}$. Then G admits decagonal graceful labeling for all n. **Proof.** Let the vertex set and edge set of G be defined by $V(G) = \{v_1, v_2, \ldots, v_n, u\}$ and $E(G) = \{uv_i : 1 \leq i \leq n\}$. Since $|V(G)| = n + 1$ and $|E(G)| = n$, we define the function $f: V(G) \to \{0, 1, 2, \ldots, D_n\}$ as follows:

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$f(u) = 0$$
$$f(v_i) = 4i^2 - 3i$$

Thus, the resulting edge labels are distinct decagonal numbers, ensuring that G possesses a decagonal graceful labeling for all n. □

The following Figure 8 gives the decagonal graceful labeling of $G = K_{1,8}$.

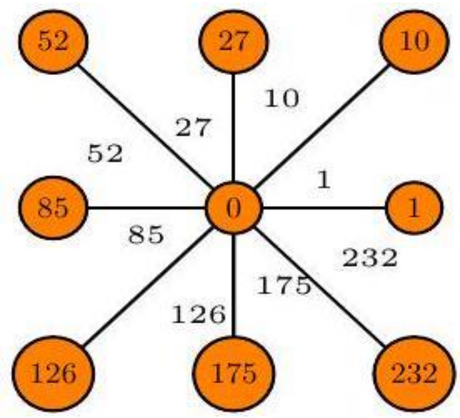


Fig. 8 Star $K_{1,8}$

The following Algorithm 4 gives the decagonal graceful labeling of $G = K_{1,n}$ for all n.

Algorithm 4 Algorithm to obtain decagonal graceful labeling of the graph $K_{1,n}$.
Input: Set of vertices n.
Output: Set of vertex label

1: $v \leftarrow \emptyset$;
2: ***for*** $i \in (0, n)$ ***do***
3: ***if*** $i = n - 1$ ***then***
4: $v \leftarrow v \cup \{0\}$
5: ***for*** $i \in (1, n + 1)$ ***do***
6: $v \leftarrow v \cup \{4i^2 - 3i\}$
7: ***end for***
8: ***end if***
9: ***end for***
10: ***return*** *reverse v*

The following Figure 9 shows that how to obtain decagonal graceful labeling of $G = K_{1,n}$ for all n using Python code and from Table 5, we can observe the following:

- The columns in Table 5 indicate the number of vertices in $K_{1,n}$.
- Each row corresponds to a specific vertex of $K_{1,n}$.
- The values inside Table 5 represent the labels assigned to the respective vertices.

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
IDLE Shell 3.12.4
File  Edit  Shell  Debug  Options  Window  Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def Star(n):
...     v=[]
...     for i in range(0,n):
...             if i==n-1:
...                 v.append(0)
...     for i in range(1,n+1):
...                 v.append(int((4*i*i-3*i)))
...     print (v[::-1])
...
>>> Star(8)
[232, 175, 126, 85, 52, 27, 10, 1, 0]
>>> |
                                                              Ln: 14  Col: 0
```

Fig. 9 Python code of the decagonal graceful labeling of $G = K_{1,n}$.

Table 5 Decagonal graceful labeling for $G = K_{1,n}$ up to 20 vertices.

Number of Vertices	Vertex Labels of K1,n																				
	n	n-1	n-2	n-3	n-4	…	…	…	…	…	…	…	…	…	…	…	…	3	2	1	u
1	1																				0
2	1	10																			0
3	1	10	27																		0
4	1	10	27	52																	0
5	1	10	27	52	85																0
6	1	10	27	52	85	126															0
7	1	10	27	52	85	126	175														0
8	1	10	27	52	85	126	175	232													0
9	1	10	27	52	85	126	175	232	297												0
10	1	10	27	52	85	126	175	232	297	370											0
11	1	10	27	52	85	126	175	232	297	370	451										0
12	1	10	27	52	85	126	175	232	297	370	451	540									0
13	1	10	27	52	85	126	175	232	297	370	451	540	637								0
14	1	10	27	52	85	126	175	232	297	370	451	540	637	742							0
15	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855						0
16	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855	976					0
17	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855	976	1105				0
18	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855	976	1105	1242			0
19	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855	976	1105	1242	1387		0
20	1	10	27	52	85	126	175	232	297	370	451	540	637	742	855	976	1105	1242	1387	1540	0

7 Application

Recent research has explored the application of network science to represent and analyze melodic relationships in music. A network model of 5-pitch sequences from improvisational music was built by Merseal et al. [27], who discovered that in listening tests, perceived relatedness was connected with melodic distance within the network.

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

By using this method, one can uncover hidden network patterns in improvised tonal music and show how language and music are both complicated systems. Other studies have applied graph-based representations to analyze melodic similarity and model musical solos Ferretti [29]. These approaches represent notes as nodes connected by edges based on sequential play, allowing for the extraction of network metrics that characterize musical pieces. Such network-based models of music have potential applications in various areas, including music classification, identification and automatic generation. Overall, these studies demonstrate the utility of network science in understanding and representing the complex relationships within musical compositions. Simonetta et al. [25] extend this concept, incorporating harmonic and contextual information through layered reductions of melodies. Their method outperforms systems lacking such structural information in music retrieval tasks. These approaches aim to capture both horizontal (contrapuntal) and vertical (harmonic) aspects of music structure.

Melodies can be graphed for use in composition and study of music, with nodes representing musical notes and edges denoting note transitions. The assignment of meaningful transitions between these nodes, which can be represented as intervals or connections between notes, is a fundamental difficulty in algorithmic composition and melody production. The edges of these graphs can be labelled with a systematic yet random sequence provided by decagonal numbers, a subset of polygonal numbers. Decagonal numbers can be used to create a mathematical foundation that can result in new melodic patterns, which could improve algorithmic composition or music analysis creative processes.

7.1 Melodic Graph Structure

A melodic graph can be defined as follows:

- Nodes: Each node represents a musical pitch or note. The set of possible pitches can be drawn from a scale (e.g., chromatic, diatonic).
- Edges: Edges between the nodes represent transitions between these notes, typically characterized by intervals (e.g., a perfect fifth, major third, etc.).
- Edge Labels: The edge labels in this graph are defined by decagonal numbers, where each decagonal number describes a particular interval or transition.

7.2 Using Decagonal Numbers for Edge Labels

The formula for the m^{th} decagonal number is $D_m = 4m^2 - 3m$. These fast-growing integers offer a non-linear sequence that can be associated with different musical aspects. There are multiple ways to apply the decagonal number in the context of edge labels:

- Interval Mapping: The time interval between two successive notes can be specified using the decagonal number. Decagonal integers, for example, may be mapped modulo 12 to provide an interval between nodes that is equivalent to musical intervals within an octave.

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Rhythmic Mapping: Longer rhythmic durations are indicated by greater decagonal numbers on the edge labels, which can also represent rhythmic values. This might be especially helpful for creating polyrhythmic frameworks or erratic rhythms.

7.3 Utilization in Algorithmic Synthesis

A composer or program can systematically introduce non-linear transitions in melodies by employing decagonal numbers as edge names. This offers a number of advantages:

- Non-linearity: Non-linear leaps between pitches are introduced by the quadratical growth of diagonal integers. This can depart from conventional stepwise or scalar motion and produce more erratic and distinctive melodic contours.

- Structural Integrity: The melody maintains some underlying structural coherence even as it introduces wider or unexpected intervals since it uses a well-defined mathematical sequence (the decagonal integers).

- Creative Exploration: Decagonal number sequences can be adjusted to investigate a variety of melodic possibilities in algorithmic composition. A composer can create a variety of melodic structures by varying the starting point n or by combining decagonal numbers.

7.4 An illustration of a Decagonal Number Melodic Graph

Consider a simple graph where:

- Nodes: Represent notes in a C Major scale (C, D, E, F, G, A, B).

- Edges: Represent transitions between notes.

- Edge Labels: Decagonal numbers, defining the interval (measured in semitones) between the notes.

For example, the edges between consecutive notes might be labeled with the first few decagonal numbers:
$D_1 = 1, D_2 = 10, D_3 = 27, D_4 = 52, D_5 = 85$

Mapping these numbers modulo 12 to fit within an octave (as intervals in semitones):

- $D_1 \text{mod} 12 = 1$ (Minor second)

- $D_2 \text{mod} 12 = 10$ (Minor seventh)

- $D_3 \text{mod} 12 = 3$ (Minor third)

- $D_4 \text{mod} 12 = 4$ (Major second)

- $D_5 \text{mod} 12 = 1$ (Minor second)

These intervals would then define the edges in the melodic graph, creating transitions such as:

- C to $C\#$ (Minor second)

- $C\#$ to $B\flat$ (Minor seventh)

- $B\flat$ to $D\flat$ (Minor third)

- $D\flat$ to F (Major third)

Decagonal numbers define this series of intervals, which result in a tune that is both musically and mathematically coherent.

Fig. 10 Melodic Graph

7.5 Application in Musical Analysis

Decagonal numbers can also be used in music analysis to examine the compositional structure of already-written works:

- Interval Analysis: One can investigate if specific artists implicitly employed mathematical sequences by examining the intervals in a melody and looking for the existence of decagonal number sequences.

- Pattern Recognition: Decagonal numbers exhibit fast growth, therefore their presence in a melody may signal passages with abrupt pitch shifts or leaps that are connected to particular composing strategies.

7.6 Computational Complexity

Another pivotal application pertains to algorithmic smoothness. Crafting melodies through decagonal numeral trajectories is computationally sleek, owing to the lucid and straightforward formulation of decagonal sequences. This clarity renders them optimal for instantaneous melodic synthesis, especially when fused with other mathematical frameworks like stochastic progression chains or generative harmonic lattices.

8 Conclusion

In this work, we proposed a new labeling technique known as decagonal graceful labeling and demonstrated that certain families of trees exhibit this property. We also developed an

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

algorithm to generate decagonal graceful labeling for these types of graphs. Additionally, we implemented Python code to facilitate labeling for the identified families. To further illustrate the concept, we presented a Melodic Graph, using decagonal numbers as edge labels, to showcase how these numbers can influence musical composition through their distinct characteristics. We hope that this work will inspire researchers to explore new labeling methods in other graph structures. We observed that C_n is decagonal graceful only when n is congruent to 0 mod 4 or 3 mod 4. Clearly, C_{10} is not decagonal graceful. Future directions of this study may include extending these results to cycle-related graphs for all n.

9 Declaration of Competing Interest

The authors disclose that they have no competing interests.

10 Data Availability Statement

No data associated in this article.

References

[1] Chartrand, G., Lesinak, L.: Graphs and Digraphs. 6th Edition, CRC Press (2016).

[2] Rosa, A.: On certain valuations of the vertices of a graph. Theory of Graphs (Internat.Symposium, Rome, July 1966), Gordon and Breach, N.Y. and Dunod Paris. (1967),349-355.

[3] Golomb, S.W.: How to number a graph, in Graph Theory and Computing. R.C. Read, ed., Academic Press, New York, (1972), 23-37.

[4] Gallian, J.A.: A dynamic survey of graph labeling. Electronic J. Comb. (2022).

[5] Chartrand, G., Egan, C., Zhang, P.: Graceful Labelings. Springer Briefs in Mathematics, (2019),7-19.

[6] Eshghi, K., Azimi, P.: Applications of mathematical programming in graceful labeling of graphs. J. Appl. Math. 1, (2004), 1-8.

[7] Brankovic, L., Reynolds, M.J.: Computer search for graceful-like labeling. Electronic Journal of Graph Theory and Applications, 1 (2022), 19-336.

[8] Aldred, R.E.L., McKay, B.D.: Graceful and harmonious labeling of trees. Bull. Inst. Combin. Appl. 23 (1998), 69-72.

[9] Horton, M.: Graceful trees: Statistics and Algorithms. Honours thesis, University of Tasmania. (2003).

[10] De Oliveira, D.L., Artigas, D., Dantas, S., Frickes, L., Luiz, A.G.: Graceful Game on Some Graph Classes. RAIRO-Oper. Res. 58 (2024), 423-439.

[11] Krishnaa, A.: Some Algorithms of Graph Theory in Cryptology. Indian Journal of Advanced Mathematics. 4(1) (2024).

[12] Perera, P.A.S.D; Wijesiri, G.S.: Encryption and Decryption Algorithms in Symmetric Key Cryptography using Graph Theory. Psychology and Education. 58(1) 3420-3427.

[13] Shruthy, V.N., Maheswari, V.: An efficient encryption process with graceful

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

labelling-A hybrid approach. AIP Conference Proceedings. 2385 (2022).
[14] Vaidya, S.K., Vihol, P.L.: Fibonacci and Super Fibonacci Labeling of some graphs. Studies in Mathematical Sciences. 2(2) (2011), 24-35.
[15] Amuthavalli, D., Shanmuga Sundaram, O.V.: Super fibonacci graceful anti-magic labeling for flower graphs and python coding. Tuijin Jishu/Journal of Propulsion Technology. 44(3) (2023).
[16] Selestin Lina, S., Asha, S.: Odd triangular graceful labeling on simple graphs. Malaya Journal of Matematik. 8(4) (2020).
[17] Akshaya, V., Asha, S.: Even Triangular Graceful Number on Special Graphs. Indian Journal of Science and Technology. 16(48) (2023), 4648-4656.
[18] Lalitha, P., Tamilselvi, L.: Hexagonal Snakes with Alpha Labeling Along a Path. National Conference on Recent Advancement in Physical Sciences(NCRAPS2019). Journal of Physics: Conference Series. 1504 (2020), 012009.
[19] Branson, L., Sutton, A.M.: Evolving labelings of graceful graphs. Proceedings of the Genetic and Evolutionary Computation Conference. (2022)., 195-203.
[20] Simarmata, N., Pratama Sandy, I., Sugeng, K.A.: Graceful labeling construction for some special tree graph using adjacency matrix. Electronic Journal of Graph Theory and Applications. 11(2) (2023), 343-356.
[21] Sureshkumar, R., Murugesan, S.: Centered Polygonal Sum Labeling of Graphs. International Journal of Mathematics And its Applications. 6(3) (2018).
[22] Anick, D.: Counting graceful labelings of trees: A theoretical and empirical study. Discrete Applied Mathematics. 198 (2016), 65-81.
[23] Kaneria, V.J., Meghpara, M.: Graceful Labeling for One Point Union for Path of Graphs. International Journal of Mathematics and its Applications. 3(2015), 49-54.
[24] Orio, N., Roda, A.: A Measure of Melodic Similarity based on a Graph Representation of the Music Structure. International Society for Music Information Retrieval Conference. (2009).
[25] Simonetta, F., Carnovalini, F., Orio, N., Roda, A.: Symbolic Music Similarity through a Graph-Based Representation. Audio Mostly Conference. (2018).
[26] Selavarani, S., Arthy, J.: Study on graceful labeling. Malaya of Mathematic. 5 (2) (2020).
[27] Merseal, H.M., Beaty, R., Kenett, Y.N., Lloyd-Cox, J., Manzano, O., Norgaard, M.: Representing melodic relationships using network science. Cognition. 2 (2023).
[28] Szeto, W.M., Wong, M.H.: A Graph - Theoretical Approach for Pattern Matching in Post-Tonal Music Analysis. Journal of New Music Research. 35(4) (2006). 307-321.
[29] Ferretti, S.: Guitar solos as networks. IEEE International Conference on Multimedia and Expo. (2016).
[30] Ünal, E., Georgiou, P., Narayanan, S.S., Chew, E.: Statistical Modeling and Retrieval of Polyphonic Music. IEEE Workshop on Multimedia Signal Processing. (2007).
[31] Pinto, A., Reinier H., Leuken, V., Demirci, M., Wiering, F., Veltkamp, R.: Indexing Music Collections Through Graph Spectra. International Society for Music

Information Retrieval Conference. (2007).
[32] Mahmoudzadeh, H., Eshghi, K.: A Metaheuristic Approach to the Graceful Labeling Problem of Graphs. IEEE Symposium on Swarm Intelligence. (2007).
[33] Mahendran, S., Murugan, K.: Pentagonal Graceful Labeling of Some Graphs. An International Scientific Journal. WSN 155 (2021), 98-112.
[34] Ramalakshmi, A., Syed Ali Nisaya, M.P.: Centered Hexagonal Graceful Labeling of n-star graph. PARIPEX - Indian Journal of Research. 8(10) (2019).
[35] Akshaya, V., Asha, S.: Heptagonal Graceful Labeling on Simple Graphs. Mathematical Statistician and Engineering Applications. 71(1) (2022), 420-425.
[36] Mahendran, S.: Octagonal graceful labeling of some special graphs. An International Scientific Journal. WSN 156 (2021), 87-101.