EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 4, Article Number 6425
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

On the Edge-Gracefulness of Water Wheel-Type Graphs and Their Splitting Graphs

Aaron D. C. Angel[1,*], John Rafael M. Antalan[1], John Loureynz F. Gamurot[2], Richard P. Tagle[1]

[1] *Faculty, Department of Mathematics and Physics, College of Science, Central Luzon State University (3120), Science City of Muñoz, Nueva Ecija, Philippines*
[2] *Alumni, Department of Mathematics and Physics, College of Science, Central Luzon State University (3120), Science City of Muñoz, Nueva Ecija, Philippines*

Abstract. A graph G with p vertices and q edges is said to be edge-graceful if its edges can be labeled from 1 through q, in such a way that the labels induced on the vertices by adding over the labels of incident edges modulo p are distinct. Lo's Theorem is a known result under this topic, which states that if a graph G with p vertices and q edges is edge-graceful, then $p \Big| \left(q^2 + q - \frac{p(p-1)}{2}\right)$. Since the introduction of the concept of edge-graceful labeling, several works on the edge-graceful labeling of specific families of graphs have been conducted. We recall the definition of water wheel graphs (WW_n) and introduces two new related graphs, the right water wheel graphs (RWW_n) and left water wheel graphs (LWW_n). Next, using Lo's Theorem, and the concepts of divisibility and Diophantine equations, we proved the non-edge-gracefulness of these graphs. Then, using the same concepts, we determine all the edge-graceful splitting graphs of WW_n, RWW_n, and LWW_n, denoted as $S(WW_n)$, $S(RWW_n)$, and $S(LWW_n)$, respectively. Finally, Python computer programs were created and used to conclude that among these graph families, only $S(WW_3)$, $S(RWW_3)$, and $S(LWW_3)$ are edge-graceful, by providing their corresponding edge-graceful labels.

2020 Mathematics Subject Classifications: 05C76, 05C78, 11D09, 11D45, 68R10

Key Words and Phrases: Graph labeling, edge-graceful labeling, Diophantine equation, water wheel graph

---

1. Introduction

Graph Theory or the study of graphs is a vast field in mathematics which offers a plethora of unique and interesting topics. One of these particular topics in graph theory is graph labeling.

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i4.6425

Email addresses: aaronangel1123@gmail.com, aaron.angel@clsu2.edu.ph (A. D. C. Angel), jrantalan@clsu.edu.ph (J. R. M. Antalan), gamurotloureynz@gmail.com (J. L. F. Gamurot), richard_tagle@clsu.edu.ph (R. P. Tagle)

In the mid-1960s, the concept of **graph labeling** was first introduced. Graph labeling is an assignment of integers to the vertices or edges, or both, of a graph under certain conditions [1]. In the last 60 years, over 200 types or variations of graph labeling have been studied and about 2500 articles have been published [2].

One type of graph labeling is graceful labeling. A **graceful labeling** or graceful numbering is a special graph labeling of a graph on m edges in which the vertices are labeled, using a subset of distinct nonnegative integers from 0 to m and each edge of the graph is uniquely labeled by the absolute difference between the labels of the vertices incident to it. If the resulting graph edge numbers run from 1 to m inclusive, it is a graceful labeling, and the graph is said to be a graceful graph [3].

On the other hand, the concepts of **edge-gracefulness** and edge-graceful graphs are defined and introduced by Lo in 1985 [4]. Edge-graceful labeling is considered as a reversal of graceful labeling because it labels the edges first, then the labels of the vertices would depend on the labels of the edges incident to them. That is, a graph G with p vertices and q edges is said to be edge-graceful if the edges can be labeled from 1 through q, in such a way that the labels induced on the vertices by adding over the labels of incident edges modulo p are distinct [3].

Since the introduction of the concept of edge-graceful labeling in 1985, several works on the edge-graceful labeling of specific families of graphs have been conducted. For instance, Kuan et al. [5] studied edge-graceful unicyclic graphs. Lee et al. [6] on the other hand, studied the edge-graceful labeling of complete graphs. The work of Small in [7] shows that regular (even) spider graphs are edge-graceful. In 1992, Cabannis et al. [8] studied the edge-gracefulness of regular graphs and trees. Fast-forward to 2017, Venkatesan and Sekar [3] attempted to prove that among all wheel graphs W_n, only W_3 is edge-graceful while a complete proof of their result was provided recently in [9].

In this paper, we used essential graph theory concepts to study the basic properties of water wheel, right water wheel, and left water wheel graphs, and their splitting graphs. In particular, their vertex and edge sets were used to apply Lo's Theorem, an initial condition that must be satisfied to conclude that a particular graph is edge-graceful. Then, using the definition of edge-graceful labeling, and some preliminary concepts in number theory, which include the concept of the Diophantine equation, this paper determined which among the said graphs satisfied Lo's Theorem. If a particular graph does not satisfy Lo's Theorem, then it automatically does not qualify to be an edge-graceful graph. Finally, by using Python computer programs that we created, this paper constructed actual examples of edge-graceful labeling of the graphs that satisfy these initial conditions, thereby showing that the graphs are indeed edge-graceful.

2. Preliminaries

For completeness, we provide all the preliminary concepts and results necessary for the clear and organized presentation of the results of this study. We refer the readers who are unfamiliar with graph theory concepts that were briefly stated in this paper, to some standard graph theory references such as Gallian's "A Dynamic Survey of Graph

Labeling" [1] and Gross's "Graph theory and its applications"[10].

2.1. Essential Graph Theory Concepts

To begin this subsection, we briefly recall some key concepts in graph theory that are essential to this paper.

Definition 1. *A (p,q)-graph $G(V,E)$ is called* ***edge-graceful*** *if there exist a bijection $f : E \to \{1,2,....,q\}$, such that the induced mapping $f^+ : V \to \mathbb{Z}_p$, defined by*

$$f^+(u) \equiv \Big(\sum_{v \in N(u)} f(uv)\Big)(mod\, p)$$

is also a bijection, where $\mathbb{Z}_p$ is the set of integers $\{0,1,2,..,(p-1)\}$. Moreover, every $f(uv)$ are the labels of the edges with endpoint u in G.

Example 1. *An example of an edge-graceful labeling of a graph is shown in* **Figure 1**. *First, the graph involved, has seven edges and seven vertices. By following* **Definition 1**, *the edges of the graph were labeled using distinct integers from 1 to 7, and the induced labels of the vertices are distinct integers from 0 to 6.*

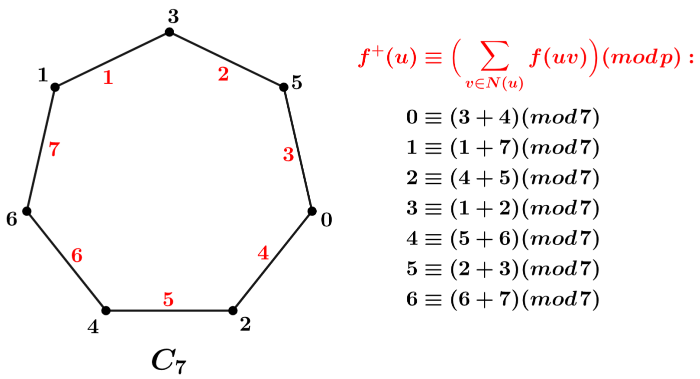

Figure 1: An Edge-graceful Labeling of Cycle Graph C_7.

Definition 2 ([11])**.** *Let $n \geq 3$ be an integer. A* ***water wheel graph*** *denoted by WW_n, is a graph with*

- *$V(WW_n) = \{u_0\} \cup \{u_i, v_i, w_i : 1 \leq i \leq n\}$; and*
- *$E(WW_n) = \{u_0u_i, u_0v_i, u_iw_i, v_iw_i : 1 \leq i \leq n\} \cup \{u_iv_{i+1} : 1 \leq i \leq n-1\} \cup \{u_nv_1\}$.*

Thus, $|V(WW_n)| = 3n + 1$ and $|E(WW_n)| = 5n$.

Example 2. *An illustration of water wheel graph WW_n is shown in* **Figure 2**.

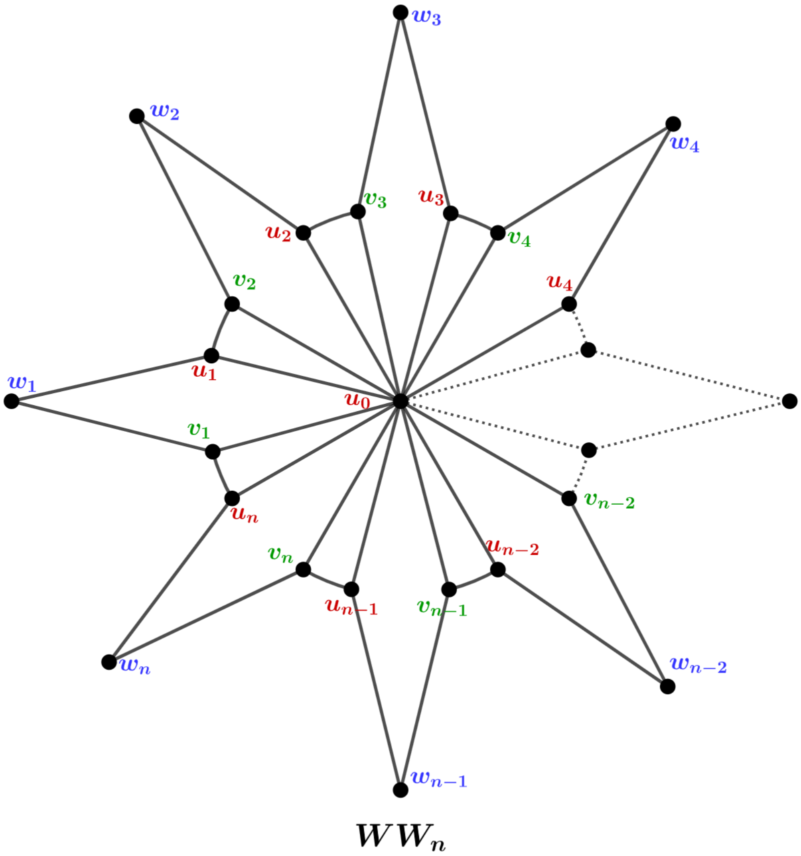

Figure 2: Water Wheel Graph WW_n.

Now, we introduce two new water wheel-related graphs, the right water wheel and left water wheel graphs.

Definition 3. *Let $n \geq 3$ be an integer. A* ***right water wheel graph*** *denoted by RWW_n, is obtained by adding edges w_nv_1 and w_iv_{i+1}, for $1 \leq i \leq n-1$, to the water wheel graph WW_n. Hence,*

- $V(RWW_n) = V(WW_n)$; *and*
- $E(WW_n) = E(WW_n) \cup \{w_iv_{i+1} : 1 \leq i \leq n-1\} \cup \{w_nv_1\}$.

Thus, $|V(RWW_n)| = 3n+1$ and $|E(RWW_n)| = 6n$.

Example 3. *An illustration of right water wheel graph RWW_n is shown in* **Figure 3**, *in the next page.*

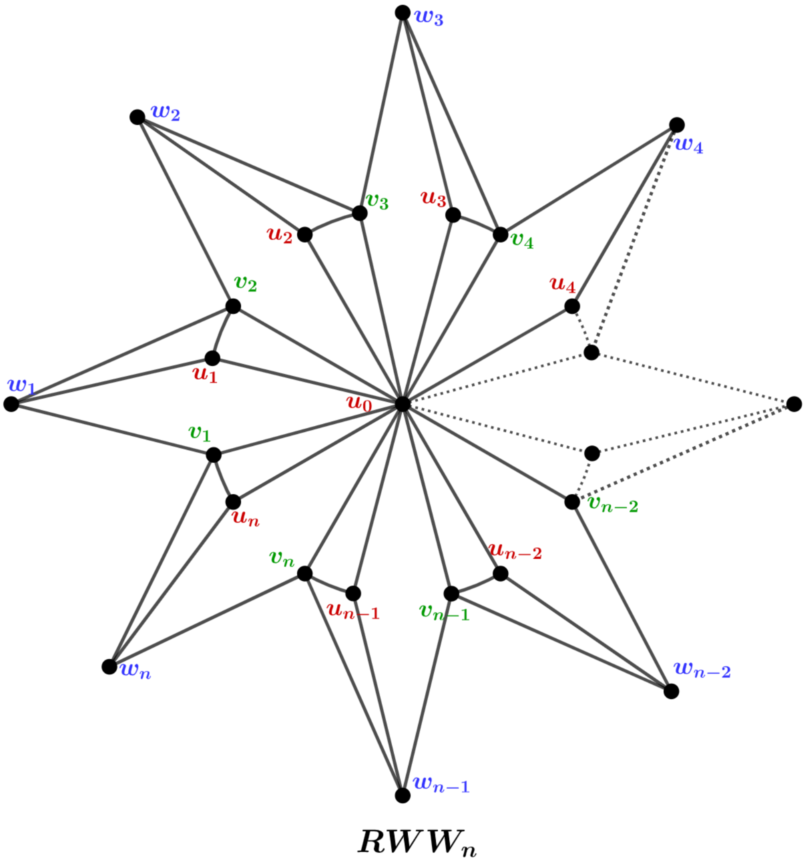

Figure 3: Right Water Wheel Graph RWW_n.

Definition 4. *Let $n \geq 3$ be an integer. A* ***left water wheel graph*** *denoted by LWW_n, is obtained by adding edges w_1u_n and w_iu_{i-1}, for $2 \leq i \leq n$, to the water wheel graph WW_n. Hence,*

- $V(RWW_n) = V(WW_n)$; *and*
- $E(WW_n) = E(WW_n) \cup \{w_iu_{i-1} : 2 \leq i \leq n\} \cup \{w_1u_n\}$.

Thus, $|V(RWW_n)| = 3n + 1$ and $|E(RWW_n)| = 6n$.

Example 4. *An illustration of left water wheel graph LWW_n is shown in* **Figure 4**, *in the next page.*

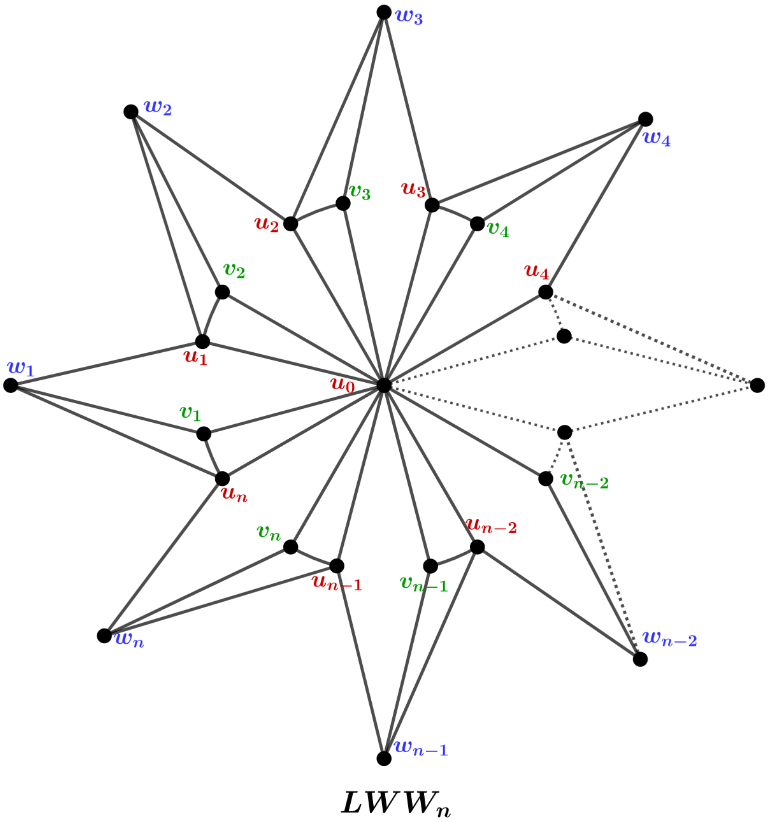

Figure 4: Left Water Wheel Graph LWW_n.

Definition 5 ([12]). *Suppose we have a graph G. The* ***splitting graph*** *of G denoted by $S(G)$ can be constructed by introducing a new vertex v' for each vertex $v \in V(G)$, and connecting v' to all the points in G that are adjacent to v.*

Example 5. *The path graph P_4 and its splitting graph $S(P_4)$ is shown in* **Figure 5**

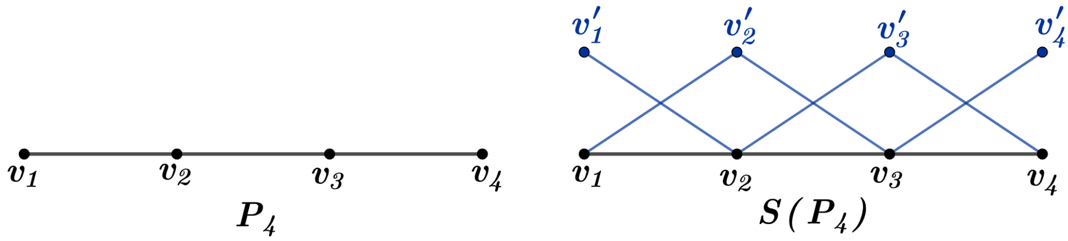

Figure 5: Splitting Graph of Path Graph P_4

Remark 1. *For convenience with the computer programs used in the study, we represent the new vertices in the splitting graph with capital letters. For example, a new vertex w_1' will be re-represented as simply vertex W_1.*

Example 6. *To illustrate* **Remark 1**, *the splitting graph $S(WW_3)$ is shown in* **Figure 6**. *The subgraph colored in black is the original water wheel graph WW_3, while the newly introduced vertices and edges are colored in blue.*

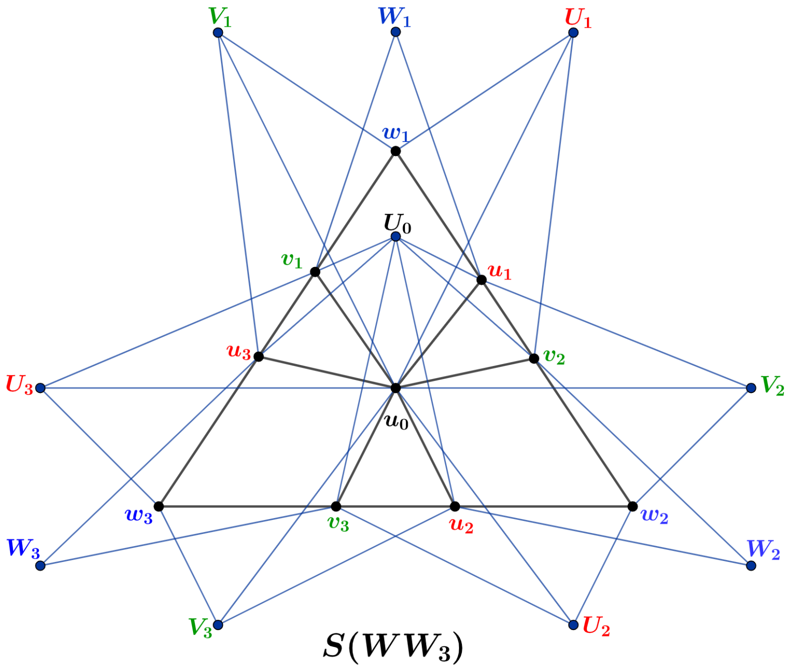

Figure 6: Splitting Graph of $S(WW_3)$

We now introduce a proposition, used in this paper, regarding the order and size of splitting graphs.

Proposition 1 ([12])**.** *Suppose that G is a (p, q)- graph, then its splitting graph $S(G)$ is a $(2p, 3q)$- graph.*

2.2. Essential Number Theory Concepts

In this subsection, we recall the important concepts and results in number theory that are used in this paper. For a more detailed discussion of basic concepts in number theory, readers may refer to Burton's "Elementary Number Theory"[13].

Definition 6. *Let a and b be two positive integers, where $a \leq b$. We say that a* **divides** *b, written in symbols by $a \mid b$, if there is a positive integer c such that $b = ac$. Otherwise, we say that a* **does not divide** *b, denoted by $a \nmid b$. Now, if $a \mid b$, it also implies the following: (i) b is a multiple of a, (ii) a is a divisor of b , and (iii) a is a factor of b.*

Definition 7. *Let p be a positive integer. Also let $a, b \in \mathbb{Z}$. We can say that a is congruent to b(modulo p) if $p \mid (a - b)$. That is if $a = b + kp$ for some $k \in \mathbb{Z}$.*

Definition 8. General Quadratic Diophantine Equation*: Diophantine equations are equations wherein only integer solutions are allowed. A general quadratic Diophantine equation is an equation of the form*

$$ax^2 + bxy + cy^2 + dx + ey + f = 0,$$

where x and y are both integers.

2.3. Essential Results Used in this Paper

This subsection discusses the Lo's theorem, which is the main theorem that we will use to obtain the results of this paper. Furthermore, some essential transformations of quadratic Diophantine equations are also discussed in this section.

Theorem 1. Lo's Theorem [4]**:** *If a graph G of p vertices and q edges is edge-graceful, then*

$$p \Big| \left(q^2 + q - \frac{p(p-1)}{2} \right).$$

A complete proof of **Theorem 1(Lo's Theorem)** was provided in [14].

Lemma 1. *[15]The quadratic Diophantine equation*

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{1}$$

where $a, b, c, d, e,$ and f are integer coefficients, and x and y are the unknown variables can be reduced to

$$X^2 - DY^2 = N, \tag{2}$$

where $X = Dy + E$, $Y = 2ax + by + d$, and $N = E^2 - DF$.
Note that in the above transformation, $D = b^2 - 4ac$,$E = bd - 2ae$, and $F = d^2 - 4af$. Moreover, observe that if X and Y is a solution to equation (2), then there are integers x and y, such that

$$x = \frac{Y - by - d}{2a} \quad \text{and} \quad y = \frac{X - E}{D}, \tag{3}$$

where x and y in equation (3) are the integer solutions for equation (1).

Remark 2. *The Diophantine equations involved in this study have $c = 0$. Thus, equation (2) will just be written as*

$$X^2 - (bY)^2 = N \tag{4}$$

Observe that we can factor the left side of equation (4) as $(X + bY)(X - bY)$. Moreover, if the pair N_1 and N_2 is a factor pair of N, then we will have linear system of equation,

$$(X + bY) = N_1 \quad \text{and} \quad (X - bY) = N_2.$$

Now, solving for X and Y, we will have,

$$X = \frac{N_1 + N_2}{2} \quad \text{and} \quad Y = \frac{N_1 - N_2}{2b}$$

Having discussed all preliminary information, we are now ready to present the results of this study.

3. Results

We begin the section by providing results on the non-edge-gracefulness of water wheel, right water wheel, and left water wheel graphs.

3.1. Non-edge-gracefulness of Water Wheel Graphs WW_n, Right Water Wheel Graphs RWW_n, and Left Water Wheel Graphs LWW_n.

Theorem 2. *For $n \geq 3$, water wheel graph WW_n is not edge-graceful.*

Proof. First, recall from **Definition 2** that water wheel graph WW_n has $3n+1$ vertices and $5n$ edges. Using **Theorem 1(Lo's Theorem)**, if a graph with p vertices and q edges is edge-graceful, then

$$p \Big| \Big(q^2 + q - \frac{p(p-1)}{2} \Big). \tag{5}$$

Thus, we substitute $p = 3n + 1$ and $q = 5n$ in equation (5). That is,

$$(3n+1) \Big| \Big((5n)^2 + 5n - \frac{(3n+1)(3n)}{2} \Big),$$

which further implies that,

$$\Big(\frac{41n^2 + 7n}{6n + 2} \Big) \in \mathbb{Z}.$$

This means,

$$\begin{aligned} 41n^2 + 7n &= (6n + 2)k \\ &= 6nk + 2k, \quad \text{for some } k \in \mathbb{Z}. \end{aligned}$$

Thus, we solve for all the possible integer values of n and k in the Diophantine equation,

$$41n^2 + 7n - 6nk - 2k = 0, \tag{6}$$

where $n, k \in \mathbb{Z}$. This Diophantine equation is of the form,

$$an^2 + bnk + ck^2 + dn + ek + f = 0,$$

where $a = 41$, $b = -6$, $c = 0$, $d = 7$, $e = -2$, and $f = 0$. Using the transformation stated in **Lemma 1**, equation (6) can be reduced to

$$\mathbf{X^2 - 36Y^2 = 13120}. \tag{7}$$

where $Y = 2an + bk + d = 82n - 6k + 7$ and $X = Dk + E = 36k + 122$.
By factoring the left side of the equation (7), we will have,

$$(X + (-6)Y)(X - (-6)Y) = 13120,$$

where $b = -6$. Therefore, by **Lemma 1** and **Remark 2**, we have the following:

$$X = \frac{N_1 + N_2}{2}, \qquad Y = \frac{N_1 - N_2}{-12}, \qquad k = \frac{X - 122}{36}, \qquad n = \frac{Y + 6k - 7}{82}, \tag{8}$$

where (N_1, N_2) are factor pairs of $N = 13120$. Using (8), we can now solve for the integer values of n and k, by considering all the factor pairs of 13120, as presented in **Table 1** in the Appendices section.
Table 1 shows that there are 4 possible solutions for the Diophantine equation $41n^2 + 7n - 6nk - 2k = 0$. These are: **(n,k)= (0,0), (1,6), (-7,-49), and (-2, -15)**. However, since we are only concern with possible values of n for water wheel graph WW_n, then by **Definition 2**, n must be a positive integer and $n \geq 3$. Thus, none of our 4 solutions satisfies both of these conditions. Hence, we have shown that **for all $n \geq 3$, water wheel graph WW_n does not satisfy the Lo's Theorem**. Therefore, we can conclude that **for all $n \geq 3$, water wheel graph WW_n is not edge-graceful**. □

Theorem 3. *For $n \geq 3$, right water wheel graph RWW_n is not edge-graceful.*

Proof. Recall from **Definition 3** that right water wheel graph RWW_n has $3n + 1$ vertices and $6n$ edges.
Similar to the proof of **Theorem 2**, we substitute $p = 3n + 1$ and $q = 6n$ in **Theorem 1(Lo's Theorem)**. That is,

$$(3n+1) \Big| \left((6n)^2 + 6n - \frac{(3n+1)(3n)}{2} \right),$$

which further implies that,

$$\left(\frac{63n^2 + 9n}{6n + 2} \right) \in \mathbb{Z}.$$

This means,

$$\begin{aligned} 63n^2 + 9n &= (6n+2)k \\ &= 6nk + 2k, \qquad \text{for some } k \in \mathbb{Z}. \end{aligned}$$

Thus, we solve for all the possible integer values of n and k in the Diophantine equation,

$$63n^2 + 9n - 6nk - 2k = 0, \tag{9}$$

where $n, k \in \mathbb{Z}$. This Diophantine equation is of the form,

$$an^2 + bnk + ck^2 + dn + ek + f = 0,$$

where $a = 63$, $b = -6$, $c = 0$, $d = 9$, $e = -2$, and $f = 0$. Using the transformation stated in **Lemma 1**, equation (9) can be reduced to

$$\mathbf{X^2 - 36Y^2 = 36288}. \tag{10}$$

where $Y = 2an + bk + d = 126n - 6k + 9$ and $X = Dk + E = 36k + 198$.
By factoring the left side of the equation (10), we will have,

$$(X + (-6)Y)(X - (-6)Y) = 36288,$$

where $b = -6$. Therefore, by **Lemma 1** and **Remark 2**, we have the following:

$$X = \frac{N_1 + N_2}{2}, \qquad Y = \frac{N_1 - N_2}{-12}, \qquad k = \frac{X - 198}{36}, \qquad n = \frac{Y + 6k - 9}{126}, \tag{11}$$

where (N_1, N_2) are factor pairs of $N = 36288$. Using (11), we can now solve for the integer values of n and k, by considering all the factor pairs of 36288, as presented in **Table 2** in the Appendices section.
Table 2 shows that there are two possible solutions for the Diophantine equation $63n^2 + 9n - 6nk - 2k = 0$. These are: **(n,k)= (0,0), and (1, 9)**. However, since we are only concern with possible values of n for right water wheel graph RWW_n, then by **Definition 3**, n must be a positive integer and $n \geq 3$. Thus, none of our two solutions satisfies both of these conditions. Hence, we have shown that **for all $n \geq 3$, right water wheel graph RWW_n does not satisfy the Lo's Theorem**. Therefore, we conclude that **for all $n \geq 3$, right water wheel graph RWW_n is not edge-graceful**. □

Theorem 4. *For $n \geq 3$, left water wheel graph LWW_n is not edge-graceful.*

Proof. Recall from **Definition 4** that left water wheel graph LWW_n, also has $3n + 1$ vertices and $6n$ edges, the same with right water wheel graph RWW_n. Since LWW_n has the same size and order as RWW_n, then the proof of **Theorem 4** will be the same as the proof of **Theorem 3**. Therefore, we already have enough evidence to conclude that **for all $n \geq 3$, no left water wheel graph LWW_n is edge-graceful**. □

We now present our results regarding the edge-gracefulness of the splitting graphs of water wheel, right water wheel, and left water wheel graphs.

3.2. On Edge-gracefulness of the Splitting Graphs of Water Wheel Graphs $S(WW_n)$, Right Water Wheel Graphs $S(RWW_n)$, and Left Water Wheel Graphs $S(LWW_n)$.

Theorem 5. *Among all splitting graphs of water wheel graphs, only $S(WW_3)$ is edge-graceful.*

Proof. First, using **Definiton 2** and **Proposition 1**, the splitting graph $S(WW_n)$ has $6n + 2$ vertices and $15n$ edges.
Hence, we substitute $p = 6n + 2$ and $q = 15n$ in **Theorem 1(Lo's Theorem)**. That is,

$$(6n + 2) \Big| \left((15n)^2 + 15n - \frac{(6n + 2)(6n + 1)}{2} \right),$$

which further implies that,

$$\left(\frac{207n^2+6n-1}{6n+2}\right) \in \mathbb{Z}.$$

This means,

$$\begin{aligned} 207n^2+6n-1 &= (6n+2)k \\ &= 6nk+2k, \quad \text{for some } k \in \mathbb{Z}. \end{aligned}$$

Thus, we solve for all the possible integer values of n and k in the Diophantine equation,

$$207n^2+6n-1-6nk-2k=0, \tag{12}$$

where $n, k \in \mathbb{Z}$. This Diophantine equation is of the form,

$$an^2+bnk+ck^2+dn+ek+f=0,$$

where $a=207$, $b=-6$, $c=0$, $d=6$, $e=-2$, and $f=-1$. Using the transformation stated in **Lemma 1**, equation (12) can be reduced to

$$\mathbf{X^2-36Y^2=596160}. \tag{13}$$

where $Y=2an+bk+d=414n-6k+6$ and $X=Dk+E=36k+792$.
By factoring the left side of the equation (13), we will have,

$$(X+(-6)Y)(X-(-6)Y)=596160,$$

where $b=-6$. Therefore, by **Lemma 1** and **Remark 2**, we have the following:

$$X=\frac{N_1+N_2}{2}, \qquad Y=\frac{N_1-N_2}{-12}, \qquad k=\frac{X-792}{36}, \qquad n=\frac{Y+6k-6}{414}, \tag{14}$$

where (N_1, N_2) are the factor pairs of $N=596160$. Using (14), we can now solve for the integer values of n and k, by considering all the factor pairs of 596160, as presented in **Table 3** in the Appendices section.
Table 3 shows that there are 2 possible solutions for the Diophantine equation $207n^2+6n-1-6nk-2k=0$. These are: **(n,k)= (3,94), and (-1,-50)**. However, since we are only concern with possible values of n for $S(WW_n)$, then only **(3,94)**, where $n=3$, is a valid solution under this conditions. Therefore, we have shown that $S(WW_3)$ **is the only splitting graph of water wheel graph that satisfies the Lo's Theorem**.
To complete this proof, an actual example of an edge-graceful labeling of $S(WW_3)$ must be provided. $S(WW_3)$ is a very large graph with 20 vertices and 45 edges. By permutation, there is a total of 45! possible labeling for its edges. Thus, the researchers created and used a Python computer program to generate an edge-graceful labeling for the said graph. One of the results from the computer program is the labeling shown in **Figure 7**.

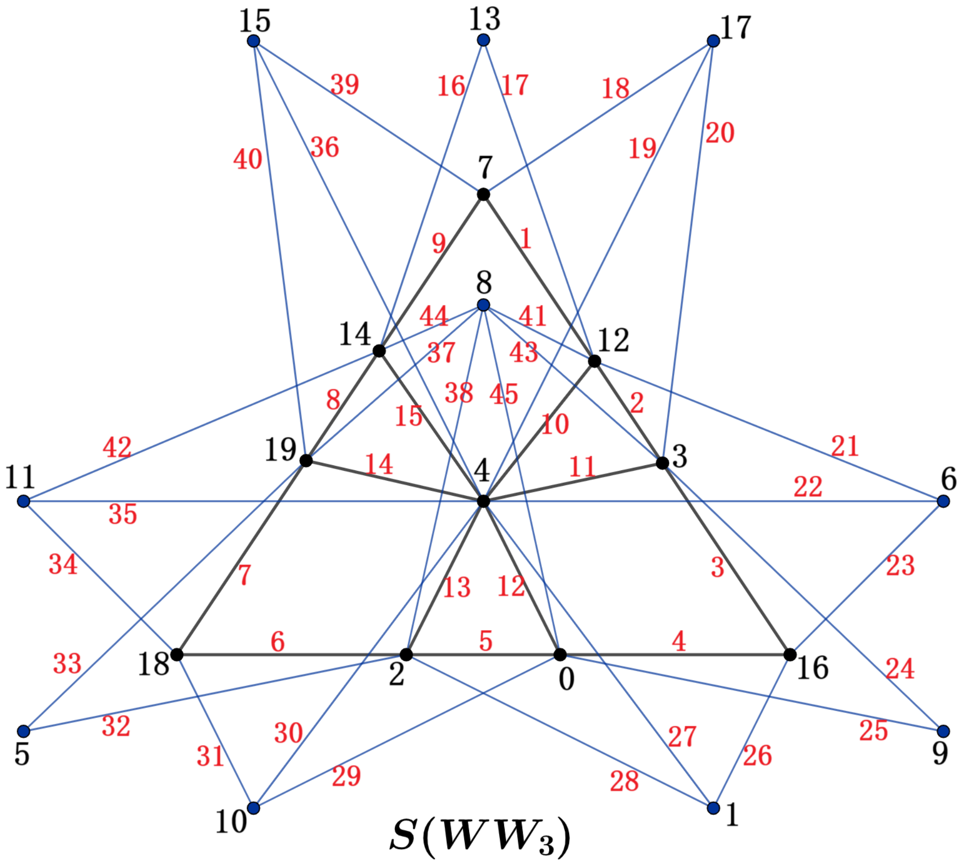

Figure 7: An Edge-graceful Labeling of $S(WW_3)$

As shown in **Figure 7**, the 45 edges of $S(WW_3)$ are labeled using distinct integers from 1 to 45. Then, by **Definition 1**, the 20 vertices are successfully labeled using distinct integers from 0 to 19. Therefore, the labeled graph in **Figure 7** is indeed an edge-graceful labeling of $S(WW_3)$.
Since all the needed conditions are satisfied, then we can now conclude that $S(WW_3)$ **is the only edge-graceful splitting graph of water wheel graphs**. $\square$

Theorem 6. *Among all splitting graphs of right water wheel graphs, only $S(RWW_3)$ is edge-graceful.*

Proof. First, using **Definiton 3** and **Proposition 1**, the splitting graph $S(RWW_n)$ has $6n+2$ vertices and $18n$ edges.
Hence, we substitute $p = 6n+2$ and $q = 18n$ in **Theorem 1(Lo's Theorem)**. That is,

$$(6n+2)\Big|\left((18n)^2 + 18n - \frac{(6n+2)(6n+1)}{2}\right),$$

which further implies that,

$$\left(\frac{306n^2 + 9n - 1}{6n + 2}\right) \in \mathbb{Z}.$$

This means,

$$\begin{aligned} 306n^2 + 9n - 1 &= (6n + 2)k \\ &= 6nk + 2k, \quad \text{for some } k \in \mathbb{Z}. \end{aligned}$$

Thus, we solve for all the possible integer values of n and k in the Diophantine equation,

$$306n^2 + 9n - 1 - 6nk - 2k = 0, \tag{15}$$

where $n, k \in \mathbb{Z}$. This Diophantine equation is of the form,

$$an^2 + bnk + ck^2 + dn + ek + f = 0,$$

where $a = 306$, $b = -6$, $c = 0$, $d = 9$, $e = -2$, and $f = -1$. Using the transformation stated in **Lemma 1**, equation (15) can be reduced to

$$\mathbf{X^2 - 36Y^2 = 1321920}. \tag{16}$$

where $Y = 2an + bk + d = 612n - 6k + 9$ and $X = Dk + E = 36k + 1170$.
By factoring the left side of the equation (16), we will have,

$$(X + (-6)Y)(X - (-6)Y) = 1321920,$$

where $b = -6$. Therefore, by **Lemma 1** and **Remark 2**, we have the following:

$$X = \frac{N_1 + N_2}{2}, \qquad Y = \frac{N_1 - N_2}{-12}, \qquad k = \frac{X - 1170}{36}, \qquad n = \frac{Y + 6k - 9}{612}, \tag{17}$$

where (N_1, N_2) are the factor pairs of $N = 1321920$. Using (17), we can now solve for the integer values of n and k, by considering all the factor pairs of 1321920, as presented in **Table 4** in the Appendices section.
Table 4 shows that there are 2 possible solutions for the Diophantine equation $306n^2 + 9n - 1 - 6nk - 2k = 0$. These are: **(n,k)= (3,139), and (-1,-74)**. However, since we are only concern with possible values of n for $S(RWW_n)$, then only **(3,139)**, where $n = 3$, is a valid solution under this conditions. Therefore, we have shown that $S(RWW_3)$ **is the only splitting graph of right water wheel graph that satisfies the Lo's Theorem**. Similar to **Theorem 5**, $S(RWW_3)$ is a very large graph with 20 vertices and 54 edges. By permutation, there is a total of 54! possible labeling for its edges. Thus, the researchers created and used a Python computer program to generate an edge-graceful labeling for the said graph. One of the results from the computer program is the labeling shown in **Figure 8**.

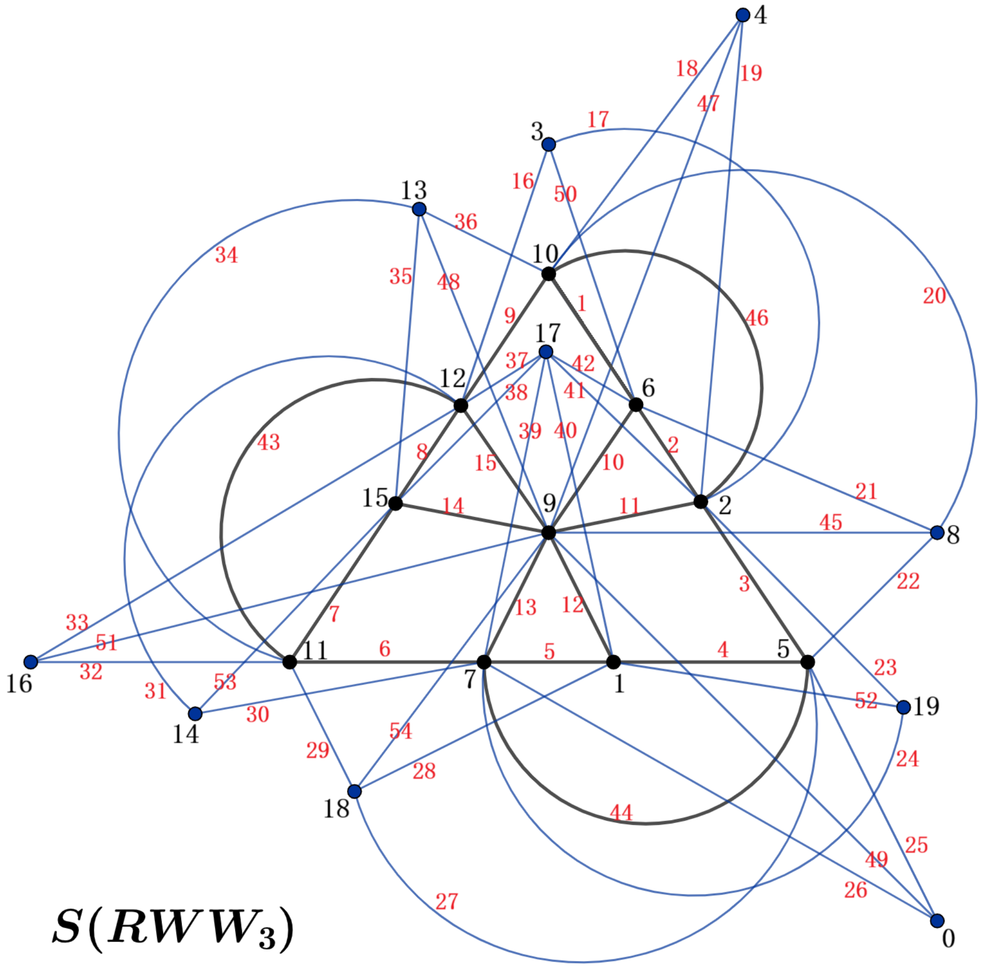

Figure 8: An Edge-graceful Labeling of $S(RWW_3)$

As shown in **Figure 8**, the 54 edges of $S(RWW_3)$ are labeled using distinct integers from 1 to 54. Then, by **Definition 1**, the 20 vertices are successfully labeled using distinct integers from 0 to 19. Therefore, the labeled graph in **Figure 8** is indeed an edge-graceful labeling of $S(RWW_3)$.

Since all the needed conditions are satisfied, then we can now conclude that $S(RWW_3)$ **is the only edge-graceful splitting graph of right water wheel graphs**. □

Theorem 7. *Among all splitting graphs of left water wheel graphs, only* $S(LWW_3)$ *is edge-graceful.*

Proof. First, using **Definiton 4** and **Proposition 1**, the splitting graph $S(LWW_n)$ also has $6n + 2$ vertices and $18n$ edges, the same as $S(RWW_n)$. Since the order and size of $S(RWW_n)$ and $S(LWW_n)$ are the same, then the proof of **Theorem 7** will just be similar to the proof of **Theorem 6**.

With the proof of **Theorem 6**, we have also shown that $S(LWW_3)$ **is the only splitting graph of left water wheel graph that satisfies the Lo's Theorem**.
Similar to the preceding theorems, an actual edge-graceful labeling of $S(LWW_3)$ must be provided, in order to complete this proof. The same with $S(RWW_3)$, splitting graph $S(LWW_3)$ also has 20 vertices and 54 edges. Thus, a Python computer program was created and used by the researchers to generate an edge-graceful labeling of the said graph. One of the results from the computer program is the labeling shown in **Figure 9**.

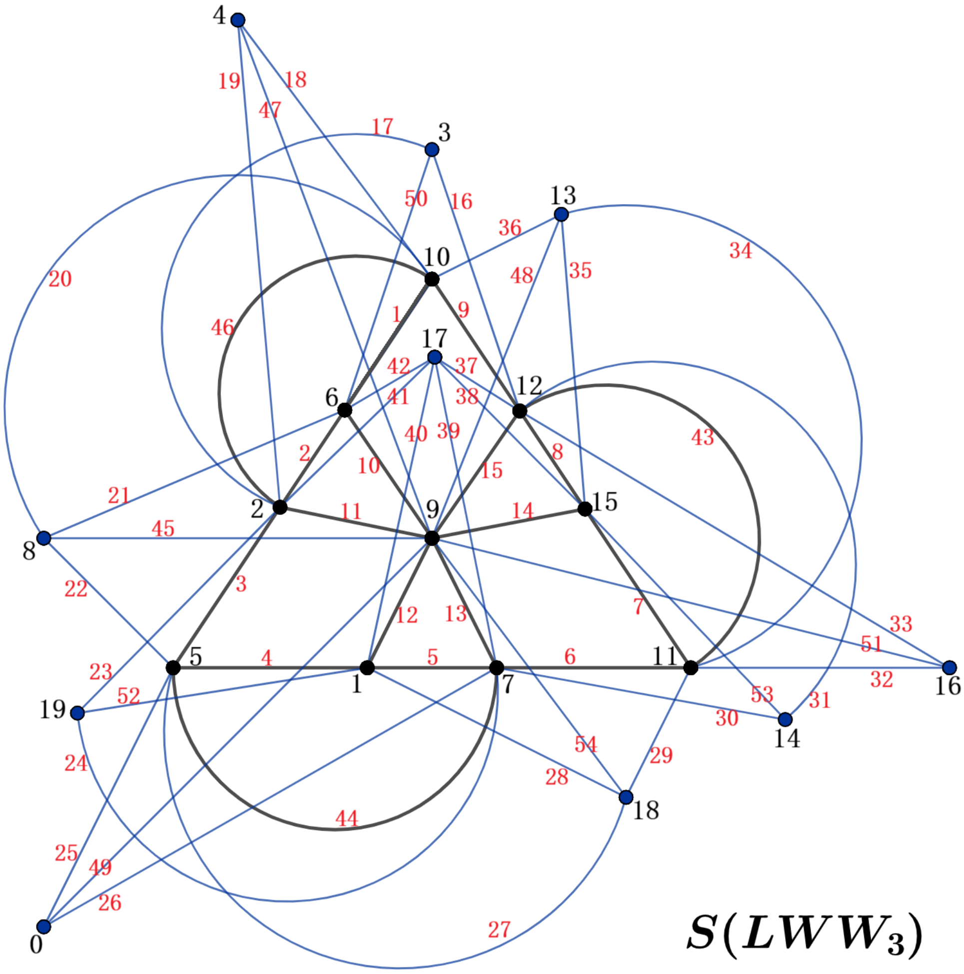

Figure 9: An Edge-graceful Labeling of $S(LWW_3)$

As shown in **Figure 9**, the 54 edges of $S(LWW_3)$ are labeled using distinct integers from 1 to 54. Then, by **Definition 1**, the 20 vertices are successfully labeled using distinct integers from 0 to 19. Therefore, the labeled graph in **Figure 9** is indeed an edge-graceful labeling of $S(LWW_3)$.
Since all the needed conditions are satisfied, then we can now conclude that $S(LWW_3)$ **is the only edge-graceful splitting graph of left water wheel graphs**. □

4. Conclusion

In this paper, two new graphs are introduced, namely the right water wheel graph RWW_n and the left water wheel graph LWW_n. Using Lo's Theorem, and the concepts of divisibility and Diophantine equations, we have proven the non-edge gracefulness of water wheel graphs WW_n, right water wheel graphs RWW_n, and left water wheel graphs LWW_n. Using similar concepts and custom-developed Python computer programs, we have also successfully investigated the edge-gracefulness of the splitting graphs of water wheel, right water wheel, and left water wheel graphs. In Theorem 5, we proved that $S(WW_3)$ is the only edge-graceful splitting graph of water wheel graphs. In Theorem 6, we showed that $S(RWW_3)$ is the only edge-graceful splitting graph of right water wheel graphs. Finally, we proved in Theorem 7 that $S(LWW_3)$ is the only edge-graceful splitting graph of left water wheel graphs.

Acknowledgements

The completion of this paper would not be possible without the help and support of the following people and organizations: Central Luzon State University (CLSU), CLSU, College of Science, Department of Mathematics and Physics, Department of Science and Technology-Science Education Institute (DOST-SEI), so our heartfelt gratitude to them. We also express our gratitude to various reviewers for their comments and suggestions that helped to improve the overall content of this paper.

References

[1] Joseph A. Gallian. A dynamic survey of graph labeling. *Electronic Journal of Combinatorics*, 6(25):4–623, 2022.

[2] S. Ramachandran and T. Gnanaseelan. Prime labeling on some cycle related graphs. *Advances and Applications in Mathematical Sciences*, 21(12):6711–6719, 2022.

[3] S. Venkatesan and P. Sekar. Not all wheel graphs were edge-graceful. *Journal Homepage: http://www.ijesm.co.in*, 6:5, 2017.

[4] S.-P. Lo. On edge-graceful labelings of graphs. *Congressus Numerantium*, 50:231–241, 1985.

[5] Q. Kuan, S. Lee, J. Mitchem, and A. Wang. On edge-graceful unicyclic graphs. *Congressus Numerantium*, 61:65–74, 1988.

[6] L. Lee, S. Lee, and G. Muty. On edge-graceful labelings of complete graphs: solutions of lo's conjecture. *Congressus Numerantium*, 62:225–233, 1988.

[7] D. Small. Regular (even) spider graphs are edge-graceful. *Congressus Numerantium*, 74:247–254, 1990.

[8] S. Cabaniss, R. Low, and J. Mitchem. On edge-graceful regular graphs and trees. *Ars Combinatoria*, 34:129–142, 1992.

[9] A. Angel, J. L. Gamurot, J. R. Antalan, and R. Tagle. On edge-gracefulness of wheel

graphs. *Communications in Combinatorics, Cryptography & Computer Science*, pages 213–219, 2024.

[10] Jonathan L. Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.

[11] Mohammed Aljohani and Salama Nagy Daoud. The edge odd graceful labeling of water wheel graphs. *Axioms*, 14(1):5, 2024.

[12] E. Sampathkumar and H. B. Walikar. On the splitting graph of a graph. *Karnatak University Science*, 25(13):13–16, 1980.

[13] David Burton. *Ebook: Elementary number theory*. McGraw Hill, 2010.

[14] Aaron DC Angel, John Rafael M. Antalan, John Loureynz F. Gamurot, and Richard P. Tagle. Edge-graceful usual fan graphs, 2024. arXiv preprint arXiv:2412.08338.

[15] Bal Bahadur Tamang. *On the study of quadratic Diophantine equations*. PhD thesis, Tribhuvan University Kathmandu, 2021.

Appendices

This section shows all the numerical tables and computer programs that are used for this study.

We begin by presenting the numerical tables used in this study. These tables show the complete set of solutions to the Diophantine equations that are involved in the study. Each table corresponds to a particular Diophantine equation associated with the main theorems, providing detailed numerical evidence that supports the theoretical results established in the earlier sections.

Presented below, is the complete table of solutions for the Diophantine equation $41n^2 + 7n - 6nk - 2k = 0$, relevant to Theorem 2.

Table 1: Solutions for $41n^2 + 7n - 6nk - 2k = 0$

N_1	N_2	X	Y	n	k	N_1	N_2	X	Y	n	k
1	13120	6560.5	1093.25	26.33	178.85	-1	-13120	-6560.5	-1093.25	-27	-185.63
2	6560	3281	546.5	13	87.75	-2	-6560	-3281	-546.5	-13.67	-94.53
4	3280	1642	273	6.33	42.22	-4	-3280	-1642	-273	**-7**	**-49**
5	2624	1314.5	218.25	5	33.13	-5	-2624	-1314.5	-218.25	-5.67	-39.90
8	1640	824	136	3	19.5	-8	-1640	-824	-136	-3.67	-26.28
10	1312	661	108.5	2.33	14.97	-10	-1312	-661	-108.5	-3	-21.75
16	820	418	67	1.33	8.22	-16	-820	-418	-67	**-2**	**-15**
20	656	338	53	**1**	**6**	-20	-656	-338	-53	-1.67	-12.78
32	410	221	31.5	0.5	2.75	-32	-410	-221	-31.5	-1.17	-9.53
40	328	184	24	0.33	1.72	-40	-328	-184	-24	-1	-8.5
41	320	180.5	23.25	0.32	1.63	-41	-320	-180.5	-23.25	-0.98	-8.40
64	205	134.5	11.75	0.08	0.35	-64	-205	-134.5	-11.75	-0.75	-7.13
80	164	122	7	**0**	**0**	-80	-164	-122	-7	-0.67	-6.78
82	160	121	6.5	-0.01	-0.03	-82	-160	-121	-6.5	-0.66	-6.75
13120	1	6560.5	-1093.25	-0.33	178.85	-13120	-1	-6560.5	1093.25	-0.34	-185.63
6560	2	3281	-546.5	-0.33	87.75	-6560	-2	-3281	546.5	-0.34	-94.53
3280	4	1642	-273	-0.33	42.22	-3280	-4	-1642	273	-0.34	-49
2624	5	1314.5	-218.25	-0.32	33.13	-2624	-5	-1314.5	218.25	-0.34	-39.90
1640	8	824	-136	-0.32	19.5	-1640	-8	-824	136	-0.35	-26.28
1312	10	661	-108.5	-0.31	14.97	-1312	-10	-661	108.5	-0.35	-21.75
820	16	418	-67	-0.30	8.22	-820	-16	-418	67	-0.37	-15
656	20	338	-53	-0.29	6	-656	-20	-338	53	-0.37	-12.78
410	32	221	-31.5	-0.27	2.75	-410	-32	-221	31.5	-0.40	-9.53
328	40	184	-24	-0.25	1.72	-328	-40	-184	24	-0.41	-8.5
320	41	180.5	-23.25	-0.25	1.63	-320	-41	-180.5	23.25	-0.42	-8.40
205	64	134.5	-11.75	-0.20	0.35	-205	-64	-134.5	11.75	-0.46	-7.13
164	80	122	-7	-0.17	0	-164	-80	-122	7	-0.50	-6.78
160	82	121	-6.5	-0.17	-0.03	-160	-82	-121	6.5	-0.5	-6.75

Presented below, is the complete table of solutions for the Diophantine equation $63n^2 + 9n - 6nk - 2k = 0$, relevant to Theorem 3.

Table 2: Solutions for $63n^2 + 9n - 6nk - 2k = 0$

N_1	N_2	X	Y	n	k	N_1	N_2	X	Y	n	k
1	36288	18144.5	3023.92	47.67	498.51	-1	-36288	-18144.5	-3023.92	-48.33	-509.51
2	18144	9073	1511.83	23.67	246.53	-2	-18144	-9073	-1511.83	-24.33	-257.53
3	12096	6049.5	1007.75	15.67	162.54	-3	-12096	-6049.5	-1007.75	-16.33	-173.54
4	9072	4538	755.67	11.67	120.56	-4	-9072	-4538	-755.67	-12.33	-131.56
6	6048	3027	503.5	7.67	78.58	-6	-6048	-3027	-503.5	-8.33	-89.58
7	5184	2595.5	431.42	6.52	66.60	-7	-5184	-2595.5	-431.42	-7.19	-77.60
8	4536	2272	377.33	5.67	57.61	-8	-4536	-2272	-377.33	-6.33	-68.61
9	4032	2020.5	335.25	5	50.63	-9	-4032	-2020.5	-335.25	-5.67	-61.63
12	3024	1518	251	3.67	36.67	-12	-3024	-1518	-251	-4.33	-47.67
14	2592	1303	214.83	3.10	30.69	-14	-2592	-1303	-214.83	-3.76	-41.69
16	2268	1142	187.67	2.67	26.22	-16	-2268	-1142	-187.67	-3.33	-37.22
18	2016	1017	166.5	2.33	22.75	-18	-2016	-1017	-166.5	-3	-33.75
21	1728	874.5	142.25	1.95	18.79	-21	-1728	-874.5	-142.25	-2.62	-29.79
24	1512	768	124	1.67	15.83	-24	-1512	-768	-124	-2.33	-26.83
27	1344	685.5	109.75	1.44	13.54	-27	-1344	-685.5	-109.75	-2.11	-24.54
28	1296	662	105.67	1.38	12.89	-28	-1296	-662	-105.67	-2.05	-23.89
32	1134	583	91.83	1.17	10.69	-32	-1134	-583	-91.83	-1.83	-21.69
36	1008	522	81	**1**	**9**	-36	-1008	-522	-81	-1.67	-20
42	864	453	68.5	0.81	7.08	-42	-864	-453	-68.5	-1.48	-18.08
48	756	402	59	0.67	5.67	-48	-756	-402	-59	-1.33	-16.67
54	672	363	51.5	0.56	4.58	-54	-672	-363	-51.5	-1.22	-15.58
56	648	352	49.33	0.52	4.28	-56	-648	-352	-49.33	-1.19	-15.28
63	576	319.5	42.75	0.43	3.38	-63	-576	-319.5	-42.75	-1.10	-14.38
64	567	315.5	41.92	0.42	3.26	-64	-567	-315.5	-41.92	-1.08	-14.26
72	504	288	36	0.33	2.5	-72	-504	-288	-36	-1	-13.5
81	448	264.5	30.58	0.26	1.85	-81	-448	-264.5	-30.58	-0.93	-12.85
84	432	258	29	0.24	1.67	-84	-432	-258	-29	-0.90	-12.67
96	378	237	23.5	0.17	1.08	-96	-378	-237	-23.5	-0.83	-12.08
108	336	222	19	0.11	0.67	-108	-336	-222	-19	-0.78	-11.67
112	324	218	17.67	0.10	0.56	-112	-324	-218	-17.67	-0.76	-11.56
126	288	207	13.5	0.05	0.25	-126	-288	-207	-13.5	-0.71	-11.25
144	252	198	9	**0**	**0**	-144	-252	-198	-9	-0.67	-11
162	224	193	5.17	-0.04	-0.14	-162	-224	-193	-5.17	-0.63	-10.86
168	216	192	4	-0.05	-0.17	-168	-216	-192	-4	-0.62	-10.83
189	192	190.5	0.25	-0.08	-0.21	-189	-192	-190.5	-0.25	-0.59	-10.79
36288	1	18144.5	-3023.92	-0.33	498.51	-36288	-1	-18144.5	3023.92	-0.33	-509.51
18144	2	9073	-1511.83	-0.33	246.53	-18144	-2	-9073	1511.83	-0.34	-257.53
12096	3	6049.5	-1007.75	-0.33	162.54	-12096	-3	-6049.5	1007.75	-0.34	-173.54
9072	4	4538	-755.67	-0.33	120.56	-9072	-4	-4538	755.67	-0.34	-131.56
6048	6	3027	-503.5	-0.33	78.58	-6048	-6	-3027	503.5	-0.34	-89.58
5184	7	2595.5	-431.42	-0.32	66.60	-5184	-7	-2595.5	431.42	-0.34	-77.60
4536	8	2272	-377.33	-0.32	57.61	-4536	-8	-2272	377.33	-0.34	-68.61
4032	9	2020.5	-335.25	-0.32	50.63	-4032	-9	-2020.5	335.25	-0.35	-61.63
3024	12	1518	-251	-0.32	36.67	-3024	-12	-1518	251	-0.35	-47.67
2592	14	1303	-214.83	-0.31	30.69	-2592	-14	-1303	214.83	-0.35	-41.69
2268	16	1142	-187.67	-0.31	26.22	-2268	-16	-1142	187.67	-0.35	-37.22

2016	18	1017	-166.5	-0.31	22.75	-2016	-18	-1017	166.5	-0.36	-33.75
1728	21	874.5	-142.25	-0.31	18.79	-1728	-21	-874.5	142.25	-0.36	-29.79
1512	24	768	-124	-0.30	15.83	-1512	-24	-768	124	-0.37	-26.83
1344	27	685.5	-109.75	-0.30	13.54	-1344	-27	-685.5	109.75	-0.37	-24.54
1296	28	662	-105.67	-0.30	12.89	-1296	-28	-662	105.67	-0.37	-23.89
1134	32	583	-91.83	-0.29	10.69	-1134	-32	-583	91.83	-0.38	-21.69
1008	36	522	-81	-0.29	9	-1008	-36	-522	81	-0.38	-20
864	42	453	-68.5	-0.28	7.08	-864	-42	-453	68.5	-0.39	-18.08
756	48	402	-59	-0.27	5.67	-756	-48	-402	59	-0.40	-16.67
672	54	363	-51.5	-0.26	4.58	-672	-54	-363	51.5	-0.40	-15.58
648	56	352	-49.33	-0.26	4.28	-648	-56	-352	49.33	-0.41	-15.28
576	63	319.5	-42.75	-0.25	3.38	-576	-63	-319.5	42.75	-0.42	-14.38
567	64	315.5	-41.92	-0.25	3.26	-567	-64	-315.5	41.92	-0.42	-14.26
504	72	288	-36	-0.24	2.5	-504	-72	-288	36	-0.43	-13.5
448	81	264.5	-30.58	-0.23	1.85	-448	-81	-264.5	30.58	-0.44	-12.85
432	84	258	-29	-0.22	1.67	-432	-84	-258	29	-0.44	-12.67
378	96	237	-23.5	-0.21	1.08	-378	-96	-237	23.5	-0.46	-12.08
336	108	222	-19	-0.19	0.67	-336	-108	-222	19	-0.48	-11.67
324	112	218	-17.67	-0.19	0.56	-324	-112	-218	17.67	-0.48	-11.56
288	126	207	-13.5	-0.17	0.25	-288	-126	-207	13.5	-0.5	-11.25
252	144	198	-9	-0.14	0	-252	-144	-198	9	-0.52	-11
224	162	193	-5.17	-0.12	-0.14	-224	-162	-193	5.17	-0.55	-10.86
216	168	192	-4	-0.11	-0.17	-216	-168	-192	4	-0.56	-10.83
192	189	190.5	-0.25	-0.08	-0.21	-192	-189	-190.5	0.25	-0.58	-10.79

Presented below, is the complete table of solutions for the Diophantine equation $207n^2 + 6n - 1 - 6nk - 2k = 0$, relevant to Theorem 5.

Table 3: Solutions for $207n^2 + 6n - 1 - 6nk - 2k = 0$

N_1	N_2	X	Y	n	k	N_1	N_2	X	Y	n	k
1	596160	298080.5	49679.92	239.67	8258.01	-1	-596160	-298080.5	-49679.92	-240.33	-8302.01
2	298080	149041	24839.83	119.67	4118.03	-2	-298080	-149041	-24839.83	-120.33	-4162.03
3	198720	99361.5	16559.75	79.67	2738.04	-3	-198720	-99361.5	-16559.75	-80.33	-2782.04
4	149040	74522	12419.67	59.67	2048.06	-4	-149040	-74522	-12419.67	-60.33	-2092.06
5	119232	59618.5	9935.58	47.67	1634.07	-5	-119232	-59618.5	-9935.58	-48.33	-1678.07
6	99360	49683	8279.5	39.67	1358.08	-6	-99360	-49683	-8279.5	-40.33	-1402.08
8	74520	37264	6209.33	29.67	1013.11	-8	-74520	-37264	-6209.33	-30.33	-1057.11
9	66240	33124.5	5519.25	26.33	898.13	-9	-66240	-33124.5	-5519.25	-27	-942.13
10	59616	29813	4967.17	23.67	806.14	-10	-59616	-29813	-4967.17	-24.33	-850.14
12	49680	24846	4139	19.67	668.17	-12	-49680	-24846	-4139	-20.33	-712.17
15	39744	19879.5	3310.75	15.67	530.21	-15	-39744	-19879.5	-3310.75	-16.33	-574.21
16	37260	18638	3103.67	14.67	495.72	-16	-37260	-18638	-3103.67	-15.33	-539.72
18	33120	16569	2758.5	13	438.25	-18	-33120	-16569	-2758.5	-13.67	-482.25
20	29808	14914	2482.33	11.67	392.28	-20	-29808	-14914	-2482.33	-12.33	-436.28
23	25920	12971.5	2158.08	10.10	338.32	-23	-25920	-12971.5	-2158.08	-10.77	-382.32
24	24840	12432	2068	9.67	323.33	-24	-24840	-12432	-2068	-10.33	-367.33
27	22080	11053.5	1837.75	8.56	285.04	-27	-22080	-11053.5	-1837.75	-9.22	-329.04
30	19872	9951	1653.5	7.67	254.42	-30	-19872	-9951	-1653.5	-8.33	-298.42

32	18630	9331	1549.83	7.17	237.19	-32	-18630	-9331	-1549.83	-7.83	-281.19
36	16560	8298	1377	6.33	208.5	-36	-16560	-8298	-1377	-7	-252.5
40	14904	7472	1238.67	5.67	185.56	-40	-14904	-7472	-1238.67	-6.33	-229.56
45	13248	6646.5	1100.25	5	162.63	-45	-13248	-6646.5	-1100.25	-5.67	-206.63
46	12960	6503	1076.17	4.88	158.64	-46	-12960	-6503	-1076.17	-5.55	-202.64
48	12420	6234	1031	4.67	151.17	-48	-12420	-6234	-1031	-5.33	-195.17
54	11040	5547	915.5	4.11	132.08	-54	-11040	-5547	-915.5	-4.78	-176.08
60	9936	4998	823	3.67	116.83	-60	-9936	-4998	-823	-4.33	-160.83
64	9315	4689.5	770.92	3.42	108.26	-64	-9315	-4689.5	-770.92	-4.08	-152.26
69	8640	4354.5	714.25	3.14	98.96	-69	-8640	-4354.5	-714.25	-3.81	-142.96
72	8280	4176	684	**3**	**94**	-72	-8280	-4176	-684	-3.67	-138
80	7452	3766	614.33	2.67	82.61	-80	-7452	-3766	-614.33	-3.33	-126.61
81	7360	3720.5	606.58	2.63	81.35	-81	-7360	-3720.5	-606.58	-3.30	-125.35
90	6624	3357	544.5	2.33	71.25	-90	-6624	-3357	-544.5	-3	-115.25
92	6480	3286	532.33	2.28	69.28	-92	-6480	-3286	-532.33	-2.94	-113.28
96	6210	3153	509.5	2.17	65.58	-96	-6210	-3153	-509.5	-2.83	-109.58
108	5520	2814	451	1.89	56.17	-108	-5520	-2814	-451	-2.56	-100.17
115	5184	2649.5	422.42	1.75	51.60	-115	-5184	-2649.5	-422.42	-2.42	-95.60
120	4968	2544	404	1.67	48.67	-120	-4968	-2544	-404	-2.33	-92.67
135	4416	2275.5	356.75	1.44	41.21	-135	-4416	-2275.5	-356.75	-2.11	-85.21
138	4320	2229	348.5	1.41	39.92	-138	-4320	-2229	-348.5	-2.07	-83.92
144	4140	2142	333	1.33	37.5	-144	-4140	-2142	-333	-2	-81.5
160	3726	1943	297.17	1.17	31.97	-160	-3726	-1943	-297.17	-1.83	-75.97
162	3680	1921	293.17	1.15	31.36	-162	-3680	-1921	-293.17	-1.81	-75.36
180	3312	1746	261	1	26.5	-180	-3312	-1746	-261	-1.67	-70.5
184	3240	1712	254.67	0.97	25.56	-184	-3240	-1712	-254.67	-1.64	-69.56
192	3105	1648.5	242.75	0.92	23.79	-192	-3105	-1648.5	-242.75	-1.58	-67.79
207	2880	1543.5	222.75	0.83	20.88	-207	-2880	-1543.5	-222.75	-1.49	-64.88
216	2760	1488	212	0.78	19.33	-216	-2760	-1488	-212	-1.44	-63.33
230	2592	1411	196.83	0.71	17.19	-230	-2592	-1411	-196.83	-1.38	-61.19
240	2484	1362	187	0.67	15.83	-240	-2484	-1362	-187	-1.33	-59.83
270	2208	1239	161.5	0.56	12.42	-270	-2208	-1239	-161.5	-1.22	-56.42
276	2160	1218	157	0.54	11.83	-276	-2160	-1218	-157	-1.20	-55.83
288	2070	1179	148.5	0.5	10.75	-288	-2070	-1179	-148.5	-1.17	-54.75
320	1863	1091.5	128.58	0.42	8.32	-320	-1863	-1091.5	-128.58	-1.08	-52.32
324	1840	1082	126.33	0.41	8.06	-324	-1840	-1082	-126.33	-1.07	-52.06
345	1728	1036.5	115.25	0.36	6.79	-345	-1728	-1036.5	-115.25	-1.03	-50.79
360	1656	1008	108	0.33	6	-360	-1656	-1008	-108	**-1**	**-50**
368	1620	994	104.33	0.32	5.61	-368	-1620	-994	-104.33	-0.99	-49.61
405	1472	938.5	88.92	0.26	4.07	-405	-1472	-938.5	-88.92	-0.93	-48.07
414	1440	927	85.5	0.25	3.75	-414	-1440	-927	-85.5	-0.91	-47.75
432	1380	906	79	0.22	3.17	-432	-1380	-906	-79	-0.89	-47.17
460	1296	878	69.67	0.19	2.39	-460	-1296	-878	-69.67	-0.86	-46.39
480	1242	861	63.5	0.17	1.92	-480	-1242	-861	-63.5	-0.83	-45.92
540	1104	822	47	0.11	0.83	-540	-1104	-822	-47	-0.78	-44.83
552	1080	816	44	0.10	0.67	-552	-1080	-816	-44	-0.77	-44.67
576	1035	805.5	38.25	0.08	0.38	-576	-1035	-805.5	-38.25	-0.75	-44.38
621	960	790.5	28.25	0.05	-0.04	-621	-960	-790.5	-28.25	-0.72	-43.96
648	920	784	22.67	0.04	-0.22	-648	-920	-784	-22.67	-0.70	-43.78
690	864	777	14.5	0.01	-0.42	-690	-864	-777	-14.5	-0.68	-43.58
720	828	774	9	0	-0.5	-720	-828	-774	-9	-0.67	-43.5
736	810	773	6.17	-0.01	-0.53	-736	-810	-773	-6.17	-0.66	-43.47
596160	1	298080.5	-49679.92	-0.33	8258.01	-596160	-1	-298080.5	49679.92	-0.33	-8302.01

298080	2	149041	-24839.83	-0.33	4118.03	-298080	-2	-149041	24839.83	-0.33	-4162.03
198720	3	99361.5	-16559.75	-0.33	2738.04	-198720	-3	-99361.5	16559.75	-0.33	-2782.04
149040	4	74522	-12419.67	-0.33	2048.06	-149040	-4	-74522	12419.67	-0.33	-2092.06
119232	5	59618.5	-9935.58	-0.33	1634.07	-119232	-5	-59618.5	9935.58	-0.34	-1678.07
99360	6	49683	-8279.5	-0.33	1358.08	-99360	-6	-49683	8279.5	-0.34	-1402.08
74520	8	37264	-6209.33	-0.33	1013.11	-74520	-8	-37264	6209.33	-0.34	-1057.11
66240	9	33124.5	-5519.25	-0.33	898.13	-66240	-9	-33124.5	5519.25	-0.34	-942.13
59616	10	29813	-4967.17	-0.33	806.14	-59616	-10	-29813	4967.17	-0.34	-850.14
49680	12	24846	-4139	-0.33	668.17	-49680	-12	-24846	4139	-0.34	-712.17
39744	15	19879.5	-3310.75	-0.33	530.21	-39744	-15	-19879.5	3310.75	-0.34	-574.21
37260	16	18638	-3103.67	-0.33	495.72	-37260	-16	-18638	3103.67	-0.34	-539.72
33120	18	16569	-2758.5	-0.33	438.25	-33120	-18	-16569	2758.5	-0.34	-482.25
29808	20	14914	-2482.33	-0.33	392.28	-29808	-20	-14914	2482.33	-0.34	-436.28
25920	23	12971.5	-2158.08	-0.32	338.32	-25920	-23	-12971.5	2158.08	-0.34	-382.32
24840	24	12432	-2068	-0.32	323.33	-24840	-24	-12432	2068	-0.34	-367.33
22080	27	11053.5	-1837.75	-0.32	285.04	-22080	-27	-11053.5	1837.75	-0.34	-329.04
19872	30	9951	-1653.5	-0.32	254.42	-19872	-30	-9951	1653.5	-0.35	-298.42
18630	32	9331	-1549.83	-0.32	237.19	-18630	-32	-9331	1549.83	-0.35	-281.19
16560	36	8298	-1377	-0.32	208.5	-16560	-36	-8298	1377	-0.35	-252.5
14904	40	7472	-1238.67	-0.32	185.56	-14904	-40	-7472	1238.67	-0.35	-229.56
13248	45	6646.5	-1100.25	-0.32	162.63	-13248	-45	-6646.5	1100.25	-0.35	-206.63
12960	46	6503	-1076.17	-0.31	158.64	-12960	-46	-6503	1076.17	-0.35	-202.64
12420	48	6234	-1031	-0.31	151.17	-12420	-48	-6234	1031	-0.35	-195.17
11040	54	5547	-915.5	-0.31	132.08	-11040	-54	-5547	915.5	-0.36	-176.08
9936	60	4998	-823	-0.31	116.83	-9936	-60	-4998	823	-0.36	-160.83
9315	64	4689.5	-770.92	-0.31	108.26	-9315	-64	-4689.5	770.92	-0.36	-152.26
8640	69	4354.5	-714.25	-0.31	98.96	-8640	-69	-4354.5	714.25	-0.36	-142.96
8280	72	4176	-684	-0.30	94	-8280	-72	-4176	684	-0.36	-138
7452	80	3766	-614.33	-0.30	82.61	-7452	-80	-3766	614.33	-0.37	-126.61
7360	81	3720.5	-606.58	-0.30	81.35	-7360	-81	-3720.5	606.58	-0.37	-125.35
6624	90	3357	-544.5	-0.30	71.25	-6624	-90	-3357	544.5	-0.37	-115.25
6480	92	3286	-532.33	-0.30	69.28	-6480	-92	-3286	532.33	-0.37	-113.28
6210	96	3153	-509.5	-0.29	65.58	-6210	-96	-3153	509.5	-0.37	-109.58
5520	108	2814	-451	-0.29	56.17	-5520	-108	-2814	451	-0.38	-100.17
5184	115	2649.5	-422.42	-0.29	51.60	-5184	-115	-2649.5	422.42	-0.38	-95.60
4968	120	2544	-404	-0.29	48.67	-4968	-120	-2544	404	-0.38	-92.67
4416	135	2275.5	-356.75	-0.28	41.21	-4416	-135	-2275.5	356.75	-0.39	-85.21
4320	138	2229	-348.5	-0.28	39.92	-4320	-138	-2229	348.5	-0.39	-83.92
4140	144	2142	-333	-0.28	37.5	-4140	-144	-2142	333	-0.39	-81.5
3726	160	1943	-297.17	-0.27	31.97	-3726	-160	-1943	297.17	-0.40	-75.97
3680	162	1921	-293.17	-0.27	31.36	-3680	-162	-1921	293.17	-0.40	-75.36
3312	180	1746	-261	-0.26	26.5	-3312	-180	-1746	261	-0.41	-70.5
3240	184	1712	-254.67	-0.26	25.56	-3240	-184	-1712	254.67	-0.41	-69.56
3105	192	1648.5	-242.75	-0.26	23.79	-3105	-192	-1648.5	242.75	-0.41	-67.79
2880	207	1543.5	-222.75	-0.25	20.88	-2880	-207	-1543.5	222.75	-0.42	-64.88
2760	216	1488	-212	-0.25	19.33	-2760	-216	-1488	212	-0.42	-63.33
2592	230	1411	-196.83	-0.24	17.19	-2592	-230	-1411	196.83	-0.43	-61.19
2484	240	1362	-187	-0.24	15.83	-2484	-240	-1362	187	-0.43	-59.83
2208	270	1239	-161.5	-0.22	12.42	-2208	-270	-1239	161.5	-0.44	-56.42
2160	276	1218	-157	-0.22	11.83	-2160	-276	-1218	157	-0.44	-55.83
2070	288	1179	-148.5	-0.22	10.75	-2070	-288	-1179	148.5	-0.45	-54.75
1863	320	1091.5	-128.58	-0.20	8.32	-1863	-320	-1091.5	128.58	-0.46	-52.32
1840	324	1082	-126.33	-0.20	8.06	-1840	-324	-1082	126.33	-0.46	-52.06

1728	345	1036.5	-115.25	-0.19	6.79	-1728	-345	-1036.5	115.25	-0.47	-50.79
1656	360	1008	-108	-0.19	6	-1656	-360	-1008	108	-0.48	-50
1620	368	994	-104.33	-0.19	5.61	-1620	-368	-994	104.33	-0.48	-49.61
1472	405	938.5	-88.92	-0.17	4.07	-1472	-405	-938.5	88.92	-0.50	-48.07
1440	414	927	-85.5	-0.17	3.75	-1440	-414	-927	85.5	-0.5	-47.75
1380	432	906	-79	-0.16	3.17	-1380	-432	-906	79	-0.51	-47.17
1296	460	878	-69.67	-0.15	2.39	-1296	-460	-878	69.67	-0.52	-46.39
1242	480	861	-63.5	-0.14	1.92	-1242	-480	-861	63.5	-0.53	-45.92
1104	540	822	-47	-0.12	0.83	-1104	-540	-822	47	-0.55	-44.83
1080	552	816	-44	-0.11	0.67	-1080	-552	-816	44	-0.56	-44.67
1035	576	805.5	-38.25	-0.10	0.38	-1035	-576	-805.5	38.25	-0.57	-44.38
960	621	790.5	-28.25	-0.08	-0.04	-960	-621	-790.5	28.25	-0.58	-43.96
920	648	784	-22.67	-0.07	-0.22	-920	-648	-784	22.67	-0.59	-43.78
864	690	777	-14.5	-0.06	-0.42	-864	-690	-777	14.5	-0.61	-43.58
828	720	774	-9	-0.04	-0.5	-828	-720	-774	9	-0.62	-43.5
810	736	773	-6.17	-0.04	-0.53	-810	-736	-773	6.17	-0.63	-43.47

Presented below, is the complete table of solutions for the Diophantine equation $306n^2 + 9n - 1 - 6nk - 2k = 0$, relevant to Theorem 6.

Table 4: Solutions for $306n^2 + 9n - 1 - 6nk - 2k = 0$

N_1	N_2	X	Y	n	k	N_1	N_2	X	Y	n	k
1	1321920	660960.5	110159.9	359.7	18327.5	-1	-1321920	-660960.5	-110159.9	-360.3	-18392.5
2	660960	330481	55079.8	179.7	9147.5	-2	-660960	-330481	-55079.8	-180.3	-9212.5
3	440640	220321.5	36719.8	119.7	6087.54	-3	-440640	-220321.5	-36719.8	-120.3	-6152.5
4	330480	165242	27539.7	89.7	4557.56	-4	-330480	-165242	-27539.7	-90.3	-4622.6
5	264384	132194.5	22031.6	71.7	3639.57	-5	-264384	-132194.5	-22031.6	-72.3	-3704.6
6	220320	110163	18359.5	59.7	3027.58	-6	-220320	-110163	-18359.5	-60.3	-3092.6
8	165240	82624	13769.3	44.7	2262.61	-8	-165240	-82624	-13769.3	-45.3	-2327.6
9	146880	73444.5	12239.3	39.7	2007.63	-9	-146880	-73444.5	-12239.3	-40.3	-2072.6
10	132192	66101	11015.2	35.7	1803.64	-10	-132192	-66101	-11015.2	-36.3	-1868.6
12	110160	55086	9179	29.7	1497.67	-12	-110160	-55086	-9179	-30.3	-1562.7
15	88128	44071.5	7342.8	23.7	1191.71	-15	-88128	-44071.5	-7342.8	-24.3	-1256.7
16	82620	41318	6883.7	22.2	1115.22	-16	-82620	-41318	-6883.7	-22.8	-1180.2
17	77760	38888.5	6478.6	20.8	1047.74	-17	-77760	-38888.5	-6478.6	-21.5	-1112.7
18	73440	36729	6118.5	19.7	987.75	-18	-73440	-36729	-6118.5	-20.3	-1052.75
20	66096	33058	5506.3	17.7	885.78	-20	-66096	-33058	-5506.3	-18.3	-950.8
24	55080	27552	4588	14.7	732.83	-24	-55080	-27552	-4588	-15.3	-797.8
27	48960	24493.5	4077.8	13	647.88	-27	-48960	-24493.5	-4077.75	-13.7	-712.9
30	44064	22047	3669.5	11.7	579.92	-30	-44064	-22047	-3669.5	-12.3	-644.9
32	41310	20671	3439.8	10.9	541.69	-32	-41310	-20671	-3439.8	-11.6	-606.7
34	38880	19457	3237.2	10.3	507.97	-34	-38880	-19457	-3237.2	-10.9	-572.97
36	36720	18378	3057	9.7	478	-36	-36720	-18378	-3057	-10.3	-543
40	33048	16544	2750.7	8.7	427.06	-40	-33048	-16544	-2750.7	-9.3	-492.06
45	29376	14710.5	2444.3	7.7	376.13	-45	-29376	-14710.5	-2444.3	-8.3	-441.13
48	27540	13794	2291	7.2	350.67	-48	-27540	-13794	-2291	-7.8	-415.67
51	25920	12985.5	2155.8	6.7	328.21	-51	-25920	-12985.5	-2155.8	-7.4	-393.21
54	24480	12267	2035.5	6.3	308.25	-54	-24480	-12267	-2035.5	-7	-373.25
60	22032	11046	1831	5.7	274.33	-60	-22032	-11046	-1831	-6.3	-339.33
64	20655	10359.5	1715.9	5.3	255.26	-64	-20655	-10359.5	-1715.9	-5.96	-320.26
68	19440	9754	1614.3	5.0	238.44	-68	-19440	-9754	-1614.3	-5.6	-303.44
72	18360	9216	1524	4.7	223.5	-72	-18360	-9216	-1524	-5.3	-288.5
80	16524	8302	1370.3	4.2	198.11	-80	-16524	-8302	-1370.3	-4.8	-263.11
81	16320	8200.5	1353.3	4.1	195.29	-81	-16320	-8200.5	-1353.3	-4.8	-260.29
85	15552	7818.5	1288.9	3.9	184.68	-85	-15552	-7818.5	-1288.9	-4.6	-249.68
90	14688	7389	1216.5	3.7	172.75	-90	-14688	-7389	-1216.5	-4.3	-237.75
96	13770	6933	1139.5	3.4	160.08	-96	-13770	-6933	-1139.5	-4.1	-225.08
102	12960	6531	1071.5	3.2	148.92	-102	-12960	-6531	-1071.5	-3.9	-213.92
108	12240	6174	1011	**3**	**139**	-108	-12240	-6174	-1011	-3.7	-204
120	11016	5568	908	2.7	122.17	-120	-11016	-5568	-908	-3.3	-187.17
135	9792	4963.5	804.8	2.3	105.375	-135	-9792	-4963.5	-804.75	-3	-170.38
136	9720	4928	798.7	2.3	104.39	-136	-9720	-4928	-798.7	-2.98	-169.39
144	9180	4662	753	2.2	97	-144	-9180	-4662	-753	-2.83	-162
153	8640	4396.5	707.3	2.02	89.63	-153	-8640	-4396.5	-707.25	-2.69	-154.63
160	8262	4211	675.2	1.9	84.47	-160	-8262	-4211	-675.2	-2.58	-149.47
162	8160	4161	666.5	1.9	83.08	-162	-8160	-4161	-666.5	-2.56	-148.08
170	7776	3973	633.8	1.8	77.86	-170	-7776	-3973	-633.8	-2.45	-142.86
180	7344	3762	597	1.7	72	-180	-7344	-3762	-597	-2.33	-137

192	6885	3538.5	557.8	1.5	65.79	-192	-6885	-3538.5	-557.75	-2.21	-130.79
204	6480	3342	523	1.4	60.33	-204	-6480	-3342	-523	-2.10	-125.33
216	6120	3168	492	1.3	55.5	-216	-6120	-3168	-492	-2	-120.5
240	5508	2874	439	1.2	47.33	-240	-5508	-2874	-439	-1.83	-112.33
243	5440	2841.5	433.1	1.1	46.43	-243	-5440	-2841.5	-433.1	-1.81	-111.43
255	5184	2719.5	410.8	1.1	43.04	-255	-5184	-2719.5	-410.75	-1.75	-108.04
270	4896	2583	385.5	1	39.25	-270	-4896	-2583	-385.5	-1.67	-104.25
272	4860	2566	382.3	0.99	38.78	-272	-4860	-2566	-382.3	-1.66	-103.78
288	4590	2439	358.5	0.9	35.25	-288	-4590	-2439	-358.5	-1.58	-100.25
306	4320	2313	334.5	0.8	31.75	-306	-4320	-2313	-334.5	-1.51	-96.75
320	4131	2225.5	317.6	0.8	29.32	-320	-4131	-2225.5	-317.6	-1.46	-94.32
324	4080	2202	313	0.8	28.67	-324	-4080	-2202	-313	-1.44	-93.67
340	3888	2114	295.7	0.7	26.22	-340	-3888	-2114	-295.7	-1.39	-91.22
360	3672	2016	276	0.7	23.5	-360	-3672	-2016	-276	-1.33	-88.5
405	3264	1834.5	238.3	0.6	18.46	-405	-3264	-1834.5	-238.25	-1.22	-83.46
408	3240	1824	236	0.5	18.17	-408	-3240	-1824	-236	-1.22	-83.17
432	3060	1746	219	0.5	16	-432	-3060	-1746	-219	-1.17	-81
459	2880	1669.5	201.8	0.5	13.88	-459	-2880	-1669.5	-201.75	-1.12	-78.88
480	2754	1617	189.5	0.4	12.42	-480	-2754	-1617	-189.5	-1.08	-77.42
486	2720	1603	186.2	0.4	12.03	-486	-2720	-1603	-186.2	-1.07	-77.03
510	2592	1551	173.5	0.4	10.58	-510	-2592	-1551	-173.5	-1.04	-75.58
540	2448	1494	159	0.3	9	-540	-2448	-1494	-159	**-1**	**-74**
544	2430	1487	157.2	0.3	8.81	-544	-2430	-1487	-157.2	-0.995	-73.81
576	2295	1435.5	143.3	0.3	7.38	-576	-2295	-1435.5	-143.25	-0.96	-72.38
612	2160	1386	129	0.3	6	-612	-2160	-1386	-129	-0.92	-71
648	2040	1344	116	0.2	4.83	-648	-2040	-1344	-116	-0.89	-69.83
680	1944	1312	105.3	0.2	3.94	-680	-1944	-1312	-105.3	-0.86	-68.94
720	1836	1278	93	0.2	3	-720	-1836	-1278	-93	-0.83	-68
765	1728	1246.5	80.3	0.1	2.13	-765	-1728	-1246.5	-80.25	-0.80	-67.13
810	1632	1221	68.5	0.1	1.42	-810	-1632	-1221	-68.5	-0.78	-66.42
816	1620	1218	67	0.1	1.33	-816	-1620	-1218	-67	-0.77	-66.33
864	1530	1197	55.5	0.1	0.75	-864	-1530	-1197	-55.5	-0.75	-65.75
918	1440	1179	43.5	0.1	0.25	-918	-1440	-1179	-43.5	-0.73	-65.25
960	1377	1168.5	34.8	0.04	-0.04	-960	-1377	-1168.5	-34.75	-0.71	-64.96
972	1360	1166	32.3	0.04	-0.11	-972	-1360	-1166	-32.3	-0.70	-64.89
1020	1296	1158	23	0.02	-0.33	-1020	-1296	-1158	-23	-0.69	-64.67
1080	1224	1152	12	0	-0.5	-1080	-1224	-1152	-12	-0.67	-64.5
1088	1215	1151.5	10.6	-0.002	-0.51	-1088	-1215	-1151.5	-10.6	-0.66	-64.49
1321920	1	660960.5	-110159.9	-0.33	18327.5	-1321920	-1	-660960.5	110159.9	-0.33	-18392.51
660960	2	330481	-55079.8	-0.33	9147.53	-660960	-2	-330481	55079.8	-0.33	-9212.53
440640	3	220321.5	-36719.8	-0.33	6087.54	-440640	-3	-220321.5	36719.8	-0.33	-6152.54
330480	4	165242	-27539.7	-0.33	4557.56	-330480	-4	-165242	27539.7	-0.33	-4622.56
264384	5	132194.5	-22031.6	-0.33	3639.57	-264384	-5	-132194.5	22031.6	-0.33	-3704.57
220320	6	110163	-18359.5	-0.33	3027.58	-220320	-6	-110163	18359.5	-0.33	-3092.58
165240	8	82624	-13769.3	-0.33	2262.61	-165240	-8	-82624	13769.3	-0.34	-2327.61
146880	9	73444.5	-12239.3	-0.33	2007.63	-146880	-9	-73444.5	12239.3	-0.34	-2072.63
132192	10	66101	-11015.2	-0.33	1803.64	-132192	-10	-66101	11015.2	-0.34	-1868.64
110160	12	55086	-9179	-0.33	1497.67	-110160	-12	-55086	9179	-0.34	-1562.67
88128	15	44071.5	-7342.8	-0.33	1191.71	-88128	-15	-44071.5	7342.75	-0.34	-1256.71
82620	16	41318	-6883.7	-0.33	1115.22	-82620	-16	-41318	6883.7	-0.34	-1180.22
77760	17	38888.5	-6478.6	-0.33	1047.74	-77760	-17	-38888.5	6478.6	-0.34	-1112.74
73440	18	36729	-6118.5	-0.33	987.75	-73440	-18	-36729	6118.5	-0.34	-1052.75
66096	20	33058	-5506.3	-0.33	885.78	-66096	-20	-33058	5506.3	-0.34	-950.78

55080	24	27552	-4588	-0.33	732.83	-55080	-24	-27552	4588	-0.34	-797.83
48960	27	24493.5	-4077.8	-0.33	647.88	-48960	-27	-24493.5	4077.8	-0.34	-712.88
44064	30	22047	-3669.5	-0.33	579.92	-44064	-30	-22047	3669.5	-0.34	-644.92
41310	32	20671	-3439.8	-0.32	541.69	-41310	-32	-20671	3439.8	-0.34	-606.69
38880	34	19457	-3237.2	-0.32	507.97	-38880	-34	-19457	3237.2	-0.34	-572.97
36720	36	18378	-3057	-0.32	478	-36720	-36	-18378	3057	-0.34	-543
33048	40	16544	-2750.7	-0.32	427.06	-33048	-40	-16544	2750.7	-0.34	-492.06
29376	45	14710.5	-2444.3	-0.32	376.13	-29376	-45	-14710.5	2444.3	-0.35	-441.13
27540	48	13794	-2291	-0.32	350.67	-27540	-48	-13794	2291	-0.35	-415.67
25920	51	12985.5	-2155.8	-0.32	328.21	-25920	-51	-12985.5	2155.75	-0.35	-393.21
24480	54	12267	-2035.5	-0.32	308.25	-24480	-54	-12267	2035.5	-0.35	-373.25
22032	60	11046	-1831	-0.32	274.33	-22032	-60	-11046	1831	-0.35	-339.33
20655	64	10359.5	-1715.9	-0.32	255.26	-20655	-64	-10359.5	1715.9	-0.35	-320.26
19440	68	9754	-1614.3	-0.31	238.44	-19440	-68	-9754	1614.3	-0.35	-303.44
18360	72	9216	-1524	-0.31	223.5	-18360	-72	-9216	1524	-0.35	-288.5
16524	80	8302	-1370.3	-0.31	198.11	-16524	-80	-8302	1370.3	-0.36	-263.11
16320	81	8200.5	-1353.3	-0.31	195.29	-16320	-81	-8200.5	1353.25	-0.36	-260.29
15552	85	7818.5	-1288.9	-0.31	184.68	-15552	-85	-7818.5	1288.9	-0.36	-249.68
14688	90	7389	-1216.5	-0.31	172.75	-14688	-90	-7389	1216.5	-0.36	-237.75
13770	96	6933	-1139.5	-0.31	160.08	-13770	-96	-6933	1139.5	-0.36	-225.08
12960	102	6531	-1071.5	-0.31	148.92	-12960	-102	-6531	1071.5	-0.36	-213.92
12240	108	6174	-1011	-0.30	139	-12240	-108	-6174	1011	-0.36	-204
11016	120	5568	-908	-0.30	122.17	-11016	-120	-5568	908	-0.37	-187.17
9792	135	4963.5	-804.8	-0.30	105.38	-9792	-135	-4963.5	804.75	-0.37	-170.38
9720	136	4928	-798.7	-0.30	104.39	-9720	-136	-4928	798.7	-0.37	-169.39
9180	144	4662	-753	-0.29	97	-9180	-144	-4662	753	-0.37	-162
8640	153	4396.5	-707.3	-0.29	89.63	-8640	-153	-4396.5	707.25	-0.38	-154.63
8262	160	4211	-675.2	-0.29	84.47	-8262	-160	-4211	675.2	-0.38	-149.47
8160	162	4161	-666.5	-0.29	83.08	-8160	-162	-4161	666.5	-0.38	-148.08
7776	170	3973	-633.8	-0.29	77.86	-7776	-170	-3973	633.8	-0.38	-142.86
7344	180	3762	-597	-0.28	72	-7344	-180	-3762	597	-0.38	-137
6885	192	3538.5	-557.8	-0.28	65.79	-6885	-192	-3538.5	557.75	-0.39	-130.79
6480	204	3342	-523	-0.28	60.33	-6480	-204	-3342	523	-0.39	-125.33
6120	216	3168	-492	-0.27	55.5	-6120	-216	-3168	492	-0.39	-120.5
5508	240	2874	-439	-0.27	47.33	-5508	-240	-2874	439	-0.40	-112.33
5440	243	2841.5	-433.1	-0.27	46.43	-5440	-243	-2841.5	433.1	-0.40	-111.43
5184	255	2719.5	-410.8	-0.26	43.04	-5184	-255	-2719.5	410.75	-0.40	-108.04
4896	270	2583	-385.5	-0.26	39.25	-4896	-270	-2583	385.5	-0.41	-104.25
4860	272	2566	-382.3	-0.26	38.78	-4860	-272	-2566	382.3	-0.41	-103.78
4590	288	2439	-358.5	-0.25	35.25	-4590	-288	-2439	358.5	-0.41	-100.25
4320	306	2313	-334.5	-0.25	31.75	-4320	-306	-2313	334.5	-0.42	-96.75
4131	320	2225.5	-317.6	-0.25	29.32	-4131	-320	-2225.5	317.6	-0.42	-94.32
4080	324	2202	-313	-0.25	28.67	-4080	-324	-2202	313	-0.42	-93.67
3888	340	2114	-295.7	-0.24	26.22	-3888	-340	-2114	295.7	-0.43	-91.22
3672	360	2016	-276	-0.24	23.5	-3672	-360	-2016	276	-0.43	-88.5
3264	405	1834.5	-238.3	-0.22	18.46	-3264	-405	-1834.5	238.25	-0.44	-83.46
3240	408	1824	-236	-0.22	18.17	-3240	-408	-1824	236	-0.44	-83.17
3060	432	1746	-219	-0.22	16	-3060	-432	-1746	219	-0.45	-81
2880	459	1669.5	-201.8	-0.21	13.88	-2880	-459	-1669.5	201.75	-0.46	-78.88
2754	480	1617	-189.5	-0.20	12.42	-2754	-480	-1617	189.5	-0.46	-77.42
2720	486	1603	-186.2	-0.20	12.03	-2720	-486	-1603	186.2	-0.47	-77.03
2592	510	1551	-173.5	-0.19	10.58	-2592	-510	-1551	173.5	-0.47	-75.58
2448	540	1494	-159	-0.19	9	-2448	-540	-1494	159	-0.48	-74

2430	544	1487	-157.2	-0.19	8.81	-2430	-544	-1487	157.2	-0.48	-73.81
2295	576	1435.5	-143.3	-0.18	7.375	-2295	-576	-1435.5	143.25	-0.49	-72.38
2160	612	1386	-129	-0.17	6	-2160	-612	-1386	129	-0.5	-71
2040	648	1344	-116	-0.16	4.83	-2040	-648	-1344	116	-0.51	-69.83
1944	680	1312	-105.3	-0.15	3.94	-1944	-680	-1312	105.3	-0.52	-68.94
1836	720	1278	-93	-0.14	3	-1836	-720	-1278	93	-0.53	-68
1728	765	1246.5	-80.3	-0.13	2.13	-1728	-765	-1246.5	80.25	-0.54	-67.13
1632	810	1221	-68.5	-0.11	1.42	-1632	-810	-1221	68.5	-0.55	-66.42
1620	816	1218	-67	-0.11	1.33	-1620	-816	-1218	67	-0.56	-66.33
1530	864	1197	-55.5	-0.10	0.75	-1530	-864	-1197	55.5	-0.57	-65.75
1440	918	1179	-43.5	-0.08	0.25	-1440	-918	-1179	43.5	-0.58	-65.25
1377	960	1168.5	-34.8	-0.07	-0.04	-1377	-960	-1168.5	34.75	-0.59	-64.96
1360	972	1166	-32.33	-0.07	-0.11	-1360	-972	-1166	32.3	-0.60	-64.89
1296	1020	1158	-23	-0.06	-0.33	-1296	-1020	-1158	23	-0.61	-64.67
1224	1080	1152	-12	-0.04	-0.5	-1224	-1080	-1152	12	-0.63	-64.5
1215	1088	1151.5	-10.6	-0.04	-0.51	-1215	-1088	-1151.5	10.6	-0.63	-64.49

Next, we present the computer programs that are used in the study. These are Python computer programs that were used to generate actual edge-graceful labeling of needed particular graphs, in order to complete the proof of the main results of this study.

Provided below, is the Python program used to generate an actual edge-graceful labeling of $S(WW_3)$.

```
#Python Program for S(WW_3)
from itertools import permutations

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45]

# Function to compute u0 to V1
def compute_v_values(w1u1, u1v2, v2w2, w2u2, u2v3, v3w3, w3u3, u3v1, v1w1, u0u1, u0v2, u0u2, u0v3, u0u3, u0v1, W1v1, W1u1, U1w1, U1u0, U1v2, V2u1, V2u0, V2w2, W2v2, W2u2, U2w2, U2u0, U2v3, V3u2, V3u0, V3w3, W3v3, W3u3, U3w3, U3u0, U3v1, V1u3, V1u0, V1w1, U0v1, U0u3, U0v3, U0u2, U0v2, U0u1):
    u0 = (u0v1 + u0u1 + u0v2 + u0u2 + u0v3 + u0u3 + V1u0 + U1u0 + V2u0 + U2u0 + V3u0 + U3u0) % 20
    w1 = (v1w1 + w1u1 + V1w1 + U1w1) % 20
    u1 = (w1u1 + u1v2 + u0u1 + W1u1 + V2u1 + U0u1) % 20
    v2 = (u1v2 + v2w2 + u0v2 + U1v2 + W2v2 + U0v2) % 20
    w2 = (v2w2 + w2u2 + V2w2 + U2w2) % 20
    u2 = (w2u2 + u2v3 + u0u2 + W2u2 + V3u2 + U0u2) % 20
    v3 = (u2v3 + v3w3 + u0v3 + U2v3 + W3v3 + U0v3) % 20
    w3 = (v3w3 + w3u3 + V3w3 + U3w3) % 20
    u3 = (w3u3 + u3v1 + u0u3 + W3u3 + V1u3 + U0u3) % 20
```

```
    v1 = (u3v1 + v1w1 + u0v1 + U3v1 + W1v1 + U0v1) % 20
    U0 = (U0v1 + U0u3 + U0v3 + U0u2 + U0v2 +U0u1) % 20
    W1 = (W1v1 + W1u1) % 20
    U1 = (U1w1 + U1u0 + U1v2) % 20
    V2 = (V2u1 + V2u0 + V2w2) % 20
    W2 = (W2v2 + W2u2) % 20
    U2 = (U2w2 + U2u0 + U2v3) % 20
    V3 = (V3u2 + V3u0 + V3w3) % 20
    W3 = (W3v3 + W3u3) % 20
    U3 = (U3w3 + U3u0 + U3v1) % 20
    V1 = (V1u3 + V1u0 + V1w1) % 20
    return u0, w1, u1, v2, w2, u2, v3, w3, u3, v1, U0, W1, U1, V2, W2,
       U2, V3, W3, U3, V1

# Backtracking function to find valid permutations
def backtrack(perm, used):
    # If we have a complete permutation, verify u0 to V1
    if len(perm) == 45:
        v_values = compute_v_values(*perm)
        # Verify if u0 to V1 are distinct
        if len(set(v_values)) == 20:
            print(f"Permutation: {perm}, u0 to V1: {v_values}")
        return

    # Try to add each unused number to the permutation
    for num in numbers:
        if not used[num - 1]:  # Check if number is not used
            # Mark the number as used
            used[num - 1] = True
            perm.append(num)

            # Recursively try the next number
            backtrack(perm, used)

            # Undo the choice (backtrack)
            perm.pop()
            used[num - 1] = False

# Start the backtracking process
def find_valid_permutations():
    used = [False] * 45  # Keeps track of used numbers
    backtrack([], used)

# Start finding the valid permutations
find_valid_permutations()
```

Provided below, is the Python program used to generate an actual edge-graceful labeling of $S(RWW_3)$.

```
    #Python Program for S(RWW_3)
from itertools import permutations

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
    18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
    34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
    50, 51, 52, 53, 54]

# Function to compute u0 to V1
def compute_v_values(w1u1, u1v2, v2w2, w2u2, u2v3, v3w3, w3u3, u3v1,
    v1w1, u0u1, u0v2, u0u2, u0v3, u0u3, u0v1, W1v1, W1v2, U1w1, U1v2,
    V2w1, V2u1, V2w2, W2v2, W2v3, U2w2, U2v3, V3w2, V3u2, V3w3, W3v3,
    W3v1, U3w3, U3v1, V1w3, V1u3, V1w1, U0v1, U0u3, U0v3, U0u2, U0v2,
    U0u1, w3v1, w2v3, w1v2, V1u0, U3u0, W3u3, V3u0, U2u0, W2u2, V2u0,
    U1u0, W1u1):
    u0 = (u0v1 + u0u1 + u0v2 + u0u2 + u0v3 + u0u3 + V1u0 + U1u0 + V2u0
        + U2u0 + V3u0 + U3u0) % 20
    w1 = (v1w1 + w1u1 + w1v2 + V1w1 + U1w1 + V2w1) % 20
    u1 = (w1u1 + u0u1 + u1v2 + W1u1 + U0u1 + V2u1) % 20
    v2 = (w1v2 + u0v2 + u1v2 + v2w2 + W1v2 + U0v2 + U1v2 + W2v2) % 20
    w2 = (v2w2 + w2u2 + w2v3 + V2w2 + U2w2 + V3w2) % 20
    u2 = (w2u2 + u0u2 + u2v3 + W2u2 + U0u2 + V3u2) % 20
    v3 = (w2v3 + u2v3 + u0v3 + v3w3 + W2v3 + U2v3 + U0v3 + W3v3) % 20
    w3 = (v3w3 + w3u3 + w3v1 + V3w3 + U3w3 + V1w3) % 20
    u3 = (w3u3 + u0u3 + u3v1 + W3u3 + U0u3 + V1u3) % 20
    v1 = (u3v1 + u0v1 + v1w1 + w3v1 + U3v1 + U0v1 + W1v1 + W3v1) % 20
    U0 = (U0v1 + U0u1 + U0v2 + U0u2 + U0v3 + U0u3) % 20
    W1 = (W1v1 + W1u1 + W1v2) % 20
    U1 = (U1w1 + U1u0 + U1v2) % 20
    V2 = (V2w1 + V2u1 + V2u0 + V2w2) % 20
    W2 = (W2v2 + W2u2 + W2v3) % 20
    U2 = (U2w2 + U2u0 + U2v3) % 20
    V3 = (V3w2 + V3u2 + V3u0 + V3w3) % 20
    W3 = (W3v3 + W3u3 + W3v1) % 20
    U3 = (U3w3 + U3u0 + U3v1) % 20
    V1 = (V1w3 + V1u3 + V1u0 + V1w1) % 20
    return u0, w1, u1, v2, w2, u2, v3, w3, u3, v1, U0, W1, U1, V2, W2,
        U2, V3, W3, U3, V1

# Backtracking function to find valid permutations
def backtrack(perm, used):
    # If we have a complete permutation, verify u0 to V1
    if len(perm) == 54:
        v_values = compute_v_values(*perm)
        # Verify if u0 to V1 are distinct
        if len(set(v_values)) == 20:
            print(f"Permutation: {perm}, u0 to V1: {v_values}")
```

```
        return

    # Try to add each unused number to the permutation
    for num in numbers:
        if not used[num - 1]:  # Check if number is not used
            # Mark the number as used
            used[num - 1] = True
            perm.append(num)

            # Recursively try the next number
            backtrack(perm, used)

            # Undo the choice (backtrack)
            perm.pop()
            used[num - 1] = False

# Start the backtracking process
def find_valid_permutations():
    used = [False] * 54  # Keeps track of used numbers
    backtrack([], used)

# Start finding the valid permutations
find_valid_permutations()
```

Provided below, is the Python program used to generate an actual edge-graceful labeling of $S(LWW_3)$.

```
    #Python Program for S(LWW_3)
from itertools import permutations

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
    18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
    34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
    50, 51, 52, 53, 54]

# Function to compute u0 to V1
def compute_v_values(v1w1, u3v1, w3u3, v3w3, u2v3, w2u2, v2w2, u1v2,
    w1u1, u0v1, u0u3, u0v3, u0u2, u0v2, u0u1, W1u1, W1u3, V1w1, V1u3,
    U3w1, U3v1, U3w3, W3u3, W3u2, V3w3, V3u2, U2w3, U2v3, U2w2, W2u2,
    W2u1, V2w2, V2u1, U1w2, U1v2, U1w1, U0u1, U0v2, U0u2, U0v3, U0u3,
    U0v1, w2u1, w3u2, w1u3, U1u0, V2u0, W2v2, U2u0, V3u0, W3v3, U3u0,
    V1u0, W1v1):
    u0 = (u0v1 + u0u1 + u0v2 + u0u2 + u0v3 + u0u3 + V1u0 + U1u0 + V2u0
        + U2u0 + V3u0 + U3u0) % 20
    w1 = (v1w1 + w1u1 + w1u3 + V1w1 + U1w1 + U3w1) % 20
    u1 = (w1u1 + u1v2 + u0u1 + w2u1 + W1u1 + V2u1 + U0u1 + W2u1) % 20
    v2 = (u1v2 + v2w2 + u0v2 + U1v2 + W2v2 + U0v2) % 20
    w2 = (v2w2 + w2u2 + w2u1 + V2w2 + U2w2 + U1w2) % 20
```

```
    u2 = (w2u2 + u2v3 + u0u2 + w3u2 + W2u2 + V3u2 + U0u2 + W3u2) % 20
    v3 = (u2v3 + v3w3 + u0v3 + U2v3 + W3v3 + U0v3) % 20
    w3 = (v3w3 + w3u3 + w3u2 + V3w3 + U3w3 + U2w3) % 20
    u3 = (w3u3 + u3v1 + u0u3 + w1u3 + W3u3 + V1u3 + U0u3 + W1u3) % 20
    v1 = (u3v1 + v1w1 + u0v1 + U3v1 + W1v1 + U0v1) % 20
    U0 = (U0v1 + U0u3 + U0v3 + U0u2 + U0v2 + U0u1) % 20
    W1 = (W1u1 + W1v1 + W1u3) % 20
    U1 = (U1w1 + U1u0 + U1v2 + U1w2) % 20
    V2 = (V2u1 + V2u0 + V2w2) % 20
    W2 = (W2u1 + W2v2 + W2u2) % 20
    U2 = (U2w2 + U2u0 + U2v3 + U2w3) % 20
    V3 = (V3u2 + V3u0 + V3w3) % 20
    W3 = (W3u2 + W3v3 + W3u3) % 20
    U3 = (U3w3 + U3u0 + U3v1 + U3w1) % 20
    V1 = (V1u3 + V1u0 + V1w1) % 20
    return u0, w1, u1, v2, w2, u2, v3, w3, u3, v1, U0, W1, U1, V2, W2,
        U2, V3, W3, U3, V1

# Backtracking function to find valid permutations
def backtrack(perm, used):
    # If we have a complete permutation, verify u0 to V1
    if len(perm) == 54:
        v_values = compute_v_values(*perm)
        # Verify if u0 to V1 are distinct
        if len(set(v_values)) == 20:
            print(f"Permutation: {perm}, u0 to V1: {v_values}")
        return

    # Try to add each unused number to the permutation
    for num in numbers:
        if not used[num - 1]:  # Check if number is not used
            # Mark the number as used
            used[num - 1] = True
            perm.append(num)

            # Recursively try the next number
            backtrack(perm, used)

            # Undo the choice (backtrack)
            perm.pop()
            used[num - 1] = False

# Start the backtracking process
def find_valid_permutations():
    used = [False] * 54  # Keeps track of used numbers
    backtrack([], used)

# Start finding the valid permutations
```

```
find_valid_permutations()
```