International Journal of Mathematics Trends and Technology
ISSN: 2231-5373/ https://doi.org/10.14445/22315373/IJMTT-V72I7P102
Volume 72 Issue 7, 10-13, July 2026

Original Article

Graph-Theoretic Verification of Lucky and Proper Lucky Labeling Invariants

S. Akila Devi[1], S. Vembu[2], G. Jasmine[3], S. Christy Deva Suba[4]

[1]*Department of Mathematics, P.S.R. Engineering College (Autonomous), Sivakasi, Tamil Nadu, India.*
[2]*Department of Mathematics, Idhaya College for Women (Autonomous), Kumbakonam, Tamil Nadu, India.*
[3]*Department of Mathematics, The American College, Madurai, Tamil Nadu, India.*

[1]*Corresponding Author : apakilamaths@gmail.com*

Received: 24 May 2026 Revised: 29 June 2026 Accepted: 15 July 2026 Published: 30 July 2026

Abstract *- Lucky labeling is a kind of neighborhood summation graph labeling that has been studied for various classes of graphs extensively. However, relatively few efforts have been made to develop algorithms for the systematic verification of lucky and proper lucky labelings. This paper presents an algorithm for verifying lucky and proper lucky labelings of complete multipartite graphs with up to three partitions. The proposed procedure constructs the graph from the given partition sizes, assigns vertex labels according to the prescribed labeling scheme, computes the corresponding neighborhood sums, and verifies the conditions required for both lucky and proper lucky labelings. An illustrative example is provided to demonstrate the correctness of the algorithm.*

Keywords *- Lucky Labeling, Proper Lucky Labeling, Complete Multipartite Graph, Vertex Labels, Algorithm.*

1. Introduction

Graph labeling is an important branch of graph theory that has been extensively studied because of its rich mathematical structure and wide range of applications. In a graph labeling, integers are assigned to the vertices, edges, or both, subject to prescribed conditions. Over the years, numerous labeling techniques have been introduced, leading to significant developments in graph theory and its applications. Comprehensive accounts of graph labeling and its various subclasses are available in the survey of Gallian [3] and the standard graph theory texts by Bondy and Murty [1] and West[2].

Among the many graph labeling schemes, lucky labeling has attracted considerable attention since it was introduced by Czerwiński et al.[4]. Let $G=(V, E)$ be a simple graph and let $f: V(G)\rightarrow N$ be a vertex labeling. For each vertex $v\in V(G)$, the neighborhood sum is defined by $S(v)=\sum_{u\in N(v)} f(u)$, where $N(v)$ denotes the open neighborhood of v. A labeling is called a lucky labeling if adjacent vertices receive distinct neighborhood sums. The minimum value of the largest label used in such a labeling is called the lucky number of the graph.

Proper lucky labeling extends this concept by requiring the assigned vertex labels themselves to form a proper vertex labeling while preserving the lucky labeling condition. This labeling has been investigated for several graph classes, including Bloom graphs and related families, where the corresponding proper lucky numbers have been determined [5]. These studies demonstrate that neighborhood-sum-based labelings provide an effective framework for analysing structural properties of graphs.

Complete multipartite graphs form an important class of graphs in combinatorics because of their well-defined partition structure and numerous applications. Owing to their structural characteristics, they provide a suitable setting for studying neighborhood-sum-based labelings. However, determining lucky and proper lucky labelings requires repeated computation of neighborhood sums together with systematic verification of the defining conditions. Although these computations are straightforward for small graphs, they become increasingly tedious as the sizes of the partitions increase.

Motivated by these observations, this paper presents an algorithm for verifying lucky and proper lucky labelings of complete multipartite graphs with up to three partitions. The proposed algorithm constructs the graph corresponding to the prescribed partition sizes, assigns vertex labels according to the required labeling scheme, computes the neighborhood sums of all vertices, and verifies the defining conditions of both lucky and proper lucky labelings. An illustrative example is included to demonstrate the correctness of the proposed procedure.

2. Pseudocode

Input: Partition sizes $P = \{A, B, C\}$ (up to three partitions)
Output: Lucky Number $\eta(G)$ and Proper Lucky Number $\eta_{pl}(G)$
Step 1: Begin
Step 2: Construct the complete multipartite graph G from the given partition sizes.
Step 3: Assign vertex labels according to the number and sizes of partitions (1,2, or 3 partitions).
Step 4: For each vertex v, compute $S(v) = \text{sum of labels of neighbors.}$
Step 5: Set isLucky=True and isProperLucky=True.
Step 6: For every edge uv, if $S(u) = S(v)$, set both flags to False.
Step 7: For every adjacent pair (u, v), if $f(u) = f(v)$, set isProperLucky=False.
Step 8: If isLucky=True, $\eta(G)$ =maximum assigned label; otherwise, Undefined.
Step 9: If isProperLucky=True, $\eta_{pl}(G)$ =maximum assigned label; otherwise, Undefined.
Step 10: Return $\eta(G)$, $\eta_{pl}(G)$.

3. Verification Algorithm

This section presents an algorithm for verifying lucky and proper lucky labelings of complete multipartite graphs with up to three partitions.

Algorithm 1. Verification of Lucky and Proper Lucky Labeling for Complete Multipartite Graphs

Input: *Partition sizes $P = \{A, B, C\}$ (up to three partitions)*
Output: *Lucky Number $\eta(G)$ and Proper Lucky Number $\eta_{pl}(G)$*
Step 1: Construct the complete multipartite graph G.
Step 2:
If the number of partitions is 1, then
  For each vertex $v_i \in A$
    Assign label $f(v_i) = i + 1$
  End For
Else if the number of partitions is 2, then
  Let partition sizes be α and β
  If $\alpha \neq \beta$
    Assign label 1 to all vertices.
  Else
    Assign label 1 to the vertices of partition A
    Assign label 2 to the vertices of partition B
  End If
Else if the number of partitions is 3, then
  Let partition sizes be α, β, γ
  If $\alpha = \beta = \gamma$
    Assign labels 1, 2 and 3 to partitions A, B and C, respectively.
  Else
    Assign labels:
      $f(A) = 1$; $f(B) = 2$; $f(C) = 1$
  End If
End If
Step 3:
For each vertex $v \in V(G)$
  Compute induced weight
  $S(v) = \sum_{u \in N(v)} f(u)$
End For
Step 4:

```
Initialize
   isLucky ← True
   isProperLucky ← True
Step 5:
For each edge uv ∈ E(G)
   If S(u) = S(v)
      isLucky ← False
      isProperLucky ← False
   End If
End For
Step 6:
For each adjacent vertex pair (u,v)
   If f(u) = f(v)
      isProperLucky ← False
   End If
End For
Step 7:
If isLucky = True
   η(G) ← max{f(v): v ∈ V(G)}
Else
   η(G) ← Undefined
End If
Step 8:
If isProperLucky = True
   η_pl(G) ← max{f(v): v ∈ V(G)}
Else
   η_pl(G) ← Undefined
End If
Step 9:
Return η(G), η_pl(G).
```

The correctness and applicability of the proposed algorithm are illustrated through the following example.

4.1. Illustrative Example

In this section, the proposed verification algorithm is illustrated using a complete multipartite graph. The example demonstrates the construction of the graph, the assignment of vertex labels, the computation of neighborhood sums, and the verification of the lucky and proper lucky labeling conditions.

Consider the complete tripartite graph: $G = K_{2,2,2}$,

whose three-partite sets are $V_1 = \{v_1, v_2\}$, $V_2 = \{v_3, v_4\}$, $V_3 = \{v_5, v_6\}$.

Following Algorithm [alg:verification], assign the labels

$$f(v) = \begin{cases} 1, & v \in V_1, \\ 2, & v \in V_2, \\ 3, & v \in V_3. \end{cases}$$

The neighborhood sum of each vertex is obtained by adding the labels of all adjacent vertices. Since every vertex is adjacent to all vertices outside its own partition, the neighborhood sums are

$$S(v_1) = S(v_2) = 10,$$

$S(v_3) = S(v_4) = 8$, *and*

$$S(v_5) = S(v_6) = 6.$$

Since adjacent vertices belong to different partitions, every edge joins two vertices having distinct neighborhood sums. Hence, the assigned labeling satisfies the lucky labeling condition. Moreover, adjacent vertices also receive distinct labels.

Therefore, the labeling is proper. This example confirms that the proposed algorithm correctly verifies both the lucky labeling and the proper lucky labeling of the graph.

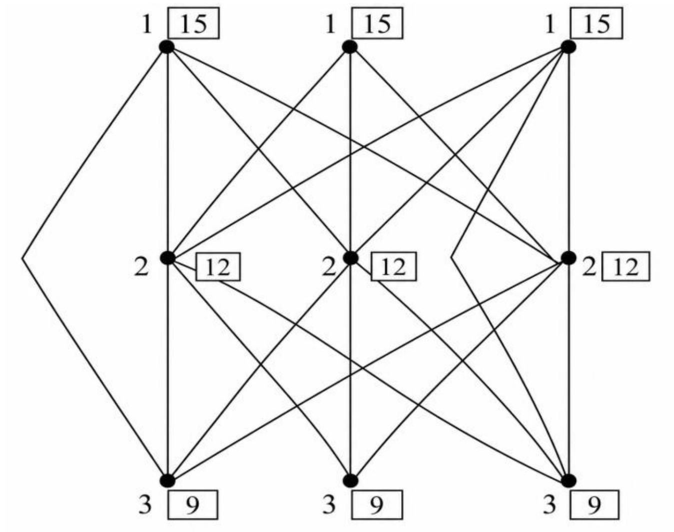

Fig. 1 Example for Lucky and proper lucy labeling of a complete tripartite graph on 3 vertices

5. Conclusion

This paper presented an algorithm for verifying lucky and proper lucky labelings of complete multipartite graphs with up to three partitions. The proposed method systematically constructs the graph, assigns vertex labels according to the prescribed labeling scheme, computes the neighborhood sums, and verifies the defining conditions of both lucky and proper lucky labelings. An illustrative example was included to demonstrate the correctness of the algorithm. The proposed approach provides a structured procedure for determining the lucky number and the proper lucky number of complete multipartite graphs. It may serve as a useful computational tool for investigating neighborhood-sum-based labelings of related graph classes.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] John Adrian Bondy, and Uppaluri Siva Ramachandra Murty, *Graph Theory*, Springer Publishing Company, Incorporated, 2008. [Google Scholar]

[2] Douglas Brent West, *Introduction to Graph Theory*, 2nd ed., Prentice Hall, 2001. [Google Scholar]

[3] Joseph A. Gallian, "A Dynamic Survey of Graph Labeling," *Electronic Journal of Combinatorics*, vol. 6, no. 25, pp. 4-623, 2022. [Google Scholar] [Publisher Link]

[4] Sebastian Czerwiński, Jarosław Grytczuk, and Wiktor Żelazny, "Lucky Labelings of Graphs," *Information Processing Letters*, vol. 109, no. 18, pp. 1078-1081, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[5] Chiranjilal Kujur, D. Antony Xavier, and S. Arul Amirtha Raja, "Lucky Labeling and Proper Lucky Labeling for Bloom Graph," *IOSR Journal of Mathematics*, vol. 13, no. 2, pp. 52-59, 2017. [CrossRef] [Google Scholar]