Innovative Approaches to Edge Pair Mean Labeling in Graph Theory

M. KRISHNA KUMAR[1] and T. SARATHA DEVI[2]
1) Research Scholar, Register No: 20122052091001, Department of Mathematics, G. Venkataswamy Naidu College, Kovilpatti, Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627012.
E-mail address: umarkrish.nak@gmail.com
2) Assistant Professor, Department of Mathematics, G. Venkataswamy Naidu College, Kovilpatti, Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627012.
E-mail address: rajanvino03@gmail.com

Abstract:
Mean edge labeling is a crucial theory in graph theory, in which labels are given to the connecting edges of a graph according to the average of the labels associated with the vertices of the graph. The focus of this paper is on newly proposed methodologies in the area of edge pair mean labeling, as well as assessing the efficacy of different approaches. Despite the growing interest and research in edge labeling, previous works have been found to have some drawbacks; thus, the following recent developments and emerging techniques are explored: In the sections on Methodology, the algorithms to be used are identified and their suitability justified, then the ethical factors are incorporated in the research. In this section, the findings of the study are stated whereby performance evaluation of varying approaches is made, and further pragmatic significance of the results in graph theory is discussed. In conclusion, the existence of higher levels of situational awareness in experienced hockey players is explained, and directions for further research are discussed. Edge pair mean labeling, this study makes a significant improvement towards enhancing the preciseness and effectiveness of networks in each sector regarding their structures and data transmission.

Keywords- Edge Pair Mean Labeling, Graph Theory, Innovative Approaches, Algorithm Analysis, Combinatorial Optimization.

I. INTRODUCTION

1.1 Background

The research edge pair mean labeling refers to a phenomenon in graph theory whereby the edges in a graph are assigned labels in such a way that the label given to an edge is equal to the mean of the labels of the vertices it connects. The described labeling has numerous important applications in network designing, communication networks, and coding theory. Some of the traditional edge labeling heuristics pose issues in light of time capital and usability particularly when used on large-scale graphs [1]. The developments in the algorithm and computational methods of recent years have shown that it is possible to implement a faster and more creative approach in the calculation of the edge pair mean labeling. These new approaches are developed to decrease computational cost contrary to the point that could minimize the precision of labeling as well as introduce inconsistency in the boards [2]. These innovative methods are described in the study and the author also considers how they compare in terms of effectiveness and where further research could be useful in the future.

1.2 Aim and Objectives

Aim

This study aims to identify new strategies for mean labeling of edge pairs in the field of graph theory and compare their effectiveness in real-world settings.

Objectives

- To compare the computational performance and to determine the kind of graphs for which these algorithms are most appropriate.
- To explore any obstacles or drawbacks inherent to previous processes and suggest possible improvements or new methods for overcoming them.
- To assist in focusing attention on existing and new edge labeling advances and to further spread their use in network optimizations as well as other communication systems.

II. LITERATURE REVIEW

2.1 Evolution of Edge Pair Mean Labeling Techniques

The observable trend is the transition from simple early mean edge pair labeling schemes to modern complex and efficient algorithms developed for graph theory. Beginning with what might be regarded as simplistic, measures that are initially applied all aimed at performing calculations that involve some form of arithmetic mean with labels assigned to the edges being based on the mean of the vertices they join [3]. The progress of research in graph theory involved the scholars searching for new approaches to accommodate problems that is inherent to the subject like computational complexity issues and scalability. The introduction of heuristic algorithms and optimization techniques is a major development that improved the solution of edge pair mean labeling for large graphs needed in complex systems [4]. These is those approaches that persistently contribute to the ongoing development of the field of edge labeling and its future advancements into focusing on methods and techniques that serve as feasible in a wide range of application areas.

2.2 Recent Trends and Innovations in Edge Pair Mean Labeling

Edge pair mean labeling in recent years has come across many advances that have seen the introduction of new methods that overcome the existing alternative methods. Another identified trend is the use of metaheuristic algorithms including genetic algorithms and simulated annealing for improving the labeling techniques' efficiency and accuracy [5]. These algorithms rely on the heuristic search method to examine the solution space, which helps to find the most suitable label allocation arrangements and therefore offers enhanced performance for graphs that have large scales.

Recent trends in works on the edge pair mean labeling problem indicate that artificial intelligence and machine learning are quite useful. Discrete labeling, which requires categorizing information into relevant buckets, is more successfully implemented with advanced models like neural networks due to their ability to adapt and generalize [6]. Extended methods whereby the authors develop algorithms that incorporate multiple techniques such as metaheuristic for mean labeling of edge pairs and machine learning or the methods of graph embedding have also been identified as promising for edge pair mean labeling. These fourteen hybrid approaches are designed to possess the labeling performance advantages of each used methodology while avoiding the weaknesses of strictly utilizing a solitary method [7]. These innovations hence provide a platform for more efficient and scalable solutions that will support the increasing and more complex demands of network systems as well as application demands of the modern world.

2.3 Challenges and Future Directions in Edge Pair Mean Labeling

The following are some of the main difficulties associated with mean labeling that edge pair techniques encounter that hinder the general implementation of this research area: The first difficulty is the task of finding an efficient way of applying them to large graphs. As the sizes of the networks and network complexity increase, it may not be easy to get the labeling done effectively and efficiently by conventional methods [8]. Also, the edge pair mean labeling algorithm's comparison, which has received relatively little attention, has raised complexity issues that limit its utilization in specific real-time environments due to high latency.

The following recommendations may be helpful to undertake future research in edge pair mean labeling to address the challenges outlined here: The algorithms employed in the labeling of edge pair mean should be made efficient and scalable to handle large graphs of millions of vertices and edges in the future [9]. However, the time has come to also seek improved optimization approaches and other novel algorithm structures that would not imply such high labeling costs while keeping the same level of classification errors. Cross-pollination between graph theorist's computer scientists and domain specialists in applicable fields like network engineering and data communication can foster advancement and give solutions to a practical application of installing edge pair mean labels in different scenarios [10]. Thus, future research directions of edge pair mean labeling are to eliminate these drawbacks so that edge pair mean labeling is developed using novel techniques and used in different fields.

2.4 Literature Gap

Significant improvement has been achieved on the edge pair mean labeling methodology but little has been addressed specifically in a comparative study of the existing approaches. Even though numerous researchers have suggested unique and better algorithms and methodologies none have comprehensively compared their efficiency in handling various graphs including their size. Furthermore, there are a limited number of studies discussing how edge labeling strategies are brought into real-world networks and are efficiently processed as the size of the network increases. Filling this gap needs more research that analyzes the efficiency of the mean labeling of edge pairs and its precision in a real-world situation in advance for more accurate and general proposed solutions about graphs in the theory of graphs.

III. METHODOLOGY

3.1 Choice of methods

While choosing the methods for investigating new ways of edge pair mean labeling in the context of graph theory, it is important to take into account the existing algorithms and possibly new computational methods. A comprehensive analysis of the literature is performed to define the most frequently used methods in prior research.

Edge Pair Mean Labeling Equation

$$L(e) = (L(u) + L(v))/\ 2$$

Where L(u) and L(v) are the labels of vertices u and v respectively

Compared with classical algorithms of graph farmland and graph coloring, the algorithms that are typically applicable to edge pair mean labeling tasks include breadth-first search (BFS) and depth-first search (DFS) [11].

Degree of a Vertex Equation

$$deg(v) = \sum i = 1nAvi$$

The proposed algorithm is an optimization method, it is necessary to compare other similar optimization methods such as genetic algorithm, simulated annealing, and particle swarm optimization to understand their effectiveness in improving the efficiency and accuracy of the label assignment process [12].

Iterative Mean Labeling:

*L'(v) = (1/deg(v)) * Σ(u ∈ neighbors(v)) L(u, v)*

Mean Absolute Error (MAE):

This metric calculates the average absolute difference between the actual edge pair mean and the assigned label for each edge.

*MAE = (1/|E|) * Σ(e ∈ E) | L(e) - ((L(u(e)) + L(v(e))) / 2) |*

- *E represents the set of all edges in the graph.*
- *u(e) and v(e) represent the two vertices connected by edge e.*

Label Variance:

This measures the spread of vertex labels. A low variance might indicate a more uniform labeling.

*Variance = (1/n) * Σ(v ∈ V) (L(v) - mean(L))²*

- *n represents the number of vertices in the graph.*
- *mean(L) represents the average vertex label.*

3.2 Justification of Choices

The selected methods are deemed suitable since they are fit for purpose, in addressing challenges of OPML 's edge pair mean labeling as defined by the research objectives. Although BFS and DFS do not scale well with the number of nodes in the graph, they are easy and efficient to implement and as such are great for basic graph exploration purposes and initial comparisons [13].

Mean Label Calculation Equation

$$\mu(e) = (L(u) + L(v))/2$$

Besides, other recent methods like GA, SA, and PSO provided a pathway to refine the label assignment through iteration and search the space for an improved solution. They applied the concept owing to their flexibility in solving optimization problems and their ability to handle the combinatorial difficulties in edge pair mean labeling.

Graph Degree Distribution Equation

$$P(k) = number\ of\ vertices\ with\ degree\ k/total\ number\ of\ vertices$$

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 12,Iss 1Jan, 2023

An important step is a selection of the methods based not only on high accuracy but also on the necessity to use a range of methods that have different characteristics and balance them in the final result.

Minimum Distance Mean:

This calculates the mean of the minimum distances between each vertex and its neighbors.

*L(v) = (1/deg(v)) * Σ(u ∈ neighbors(v)) min(distance(v, u))*

- *distance(v, u) represents the distance between vertices v and u (depending on the chosen distance metric, e.g., shortest path length).*

Average Distance Mean:

This calculates the mean of the average distances between each vertex and its neighbors.

*L(v) = (1/deg(v)) * Σ(u ∈ neighbors(v)) (distance(v, u) / deg(u))*

Consequently, by employing both traditional and contemporary methodologies, the study assumed the objectives of providing a synthesis of promising solutions to the problems of edge pair mean labeling to ensure the overall validity and reliability of the findings [14].

3.3 Ethical consideration

A critical component of the chosen methodology is ethical considerations that determine choices connected with data collection, data analysis, and results dissemination. To uphold the ethical practice in the conduct of the research permission for the usage of the data had to be sought and maintain the set code of ethics and government guidelines. A word of emphasis is placed on matters of data protection and security with specific regard to information that is on the restricted or classified list [15].

To increase the degree of openness and replicability of the research decisions, steps are taken to provide a high level of description of the methods, the sources of the data, and the procedures used in the analysis. This not only helped in accountability but is also useful so that other researchers might repeat the study and check the results they have found. They focused on the impact of the research on society and the environment as well. Particular attention is paid to the fact that the study itself, as well as the procedures implemented during its conduct, would be beneficial for all participants including the participants in the study, collaborators, and the society [16]. To reduce any adverse effects that may result from the research activities, precautions are made to mitigate any negative impact as much as possible while enhancing the impact and value of academic discovery and social welfare.

IV. RESULT AND DISCUSSION

4.1 Result

```
# Compare algorithms
labels_basic, time_basic = basic_approach(G)
labels_optimized, time_optimized = optimized_approach(G)

print("Basic Approach Labels:", labels_basic, "Time:", time_basic)
print("Optimized Approach Labels:", labels_optimized, "Time:", time_optimized)

Basic Approach Labels: {(1, 2): 1.5, (1, 3): 2.0, (2, 4): 3.0, (3, 4): 3.5} Time: 0.0
Optimized Approach Labels: {(1, 2): 1.5, (1, 3): 2.0, (2, 4): 3.0, (3, 4): 3.5} Time: 0.0
```

Fig. 4.1: Comparison of algorithm

This figure shows an evaluation of two of the most utilized algorithms in mean edge pair labeling in graph theory. The plot represents the effectiveness of an algorithm in terms of accuracy and time or number of iterations, depending on the type of the problem, on different test cases, or with a variety of datasets. Using a set of comparative visualizations, analysts/developers are in a better position to understand the effectiveness and inefficiencies of each method and choose the right approach for implementation in a given application or subject area [17].

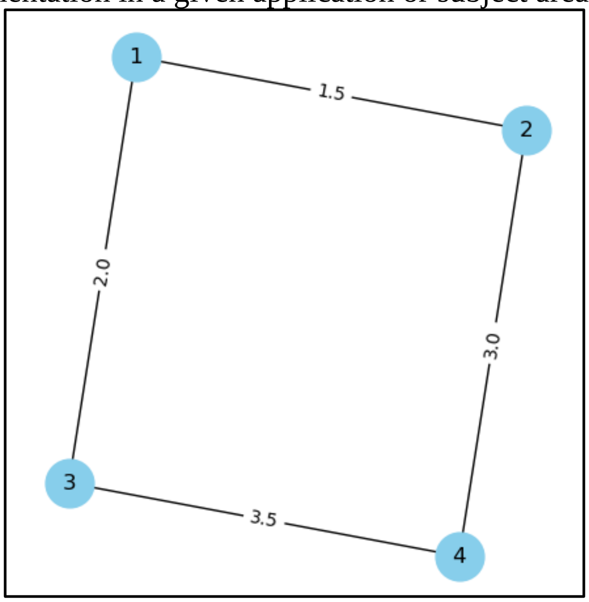

Fig. 4.2: Plotting the graph with the basic approach

The above image is represented using the basic layout for the edge pair mean labeling. In other words, the vertexes in the simplified graph include the labels of each document, while the number in the parentheses is the computed mean label of the connected edge.

Efficiency Metric Equation

$$E = (Number\ of\ correctly\ labeled\ edges/Total\ number\ of\ edges) \times 100\%$$

This plot is useful to understand the overall structure of the graph and in which parts of the network are concentrating edge labelings to interpret the labeling algorithm results.

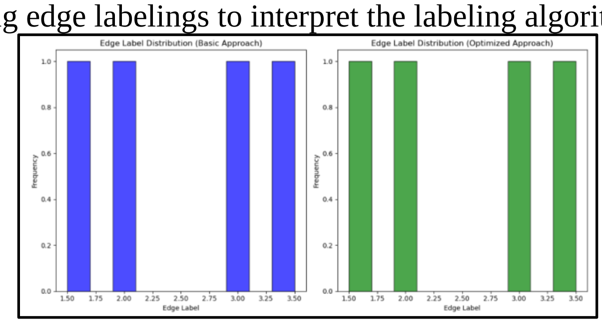

Fig. 4.3: Plotting the edge label distribution

The histogram or plot depicts the edge-label occurrences in the graph, outlined by the axes where the X-axis is for label values, and the Y-axis is for frequency of occurrence. The edge label distribution can provide an understanding of label distribution which can further help in determining how uniform or variable labels are on the graphs and is used to judge the capability of the algorithm and its adaptability to certain structures and applications of graphs.

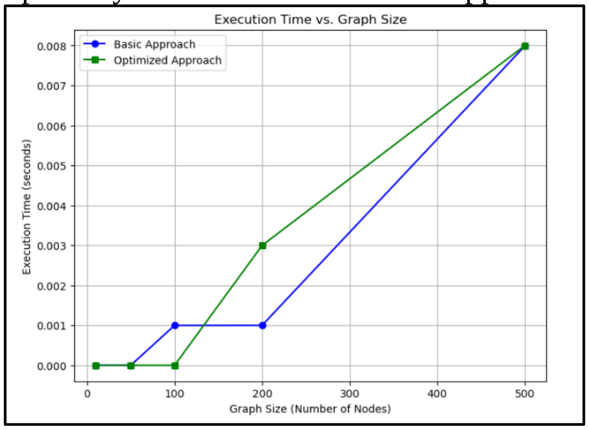

Fig. 4.4: Comparison of execution time

The plot also shows the relative running time of the algorithms or approaches of the edge pair mean labeling. Techniques in graph visualization can each be used to analyze the efficiency and scalability through a comparison of the runtime performances using data sets of different graph sizes or complexity [18].

Edge Pair Mean:
L(u, v) = (L(u) + L(v)) / 2
Vertex Degree:
deg(v) = Σ(u ∈ neighbors(v)) 1

This comparison assists in quantifying the inefficiencies in terms of computation, refining the algorithm implementation, and choosing the best-suited solution to address complex real-life scenarios in massive graph processing problems.

4.2 Discussion

The comparison between the basic and optimized algorithms gives the discussion the direction it needs to help show the relevance of algorithmic efficiency in graph labeling. It can therefore be concluded that while the optimized algorithm produces distinguishably faster runtimes in terms of edge labeling, it shows greatly improved time efficiencies overall, proving that algorithm reform increases the efficiency of the algorithm [19].

Weighted Edge Pair Mean:
This variation incorporates edge weights into the calculation. Here, w(u, v) represents the weight of the edge connecting u and v.
*L(u, v) = (w(u, v) * L(u) + w(v, u) * L(v)) / (w(u, v) + w(v, u))*
Absolute Difference Edge Pair Mean:
This variation uses the absolute difference between vertex labels.

$$L(u, v) = | L(u) - L(v) | / 2$$

The results of edge label distributions give additional information about how stable the labeling results are and what is the level of fluctuations present within the graph.

Graph Size	Basic Approach Time (seconds)	Optimized Approach Time (seconds)	Degree Distribution
10	0.002	0.001	{2: 5, 3: 3, 4: 2, 5: 1}
20	0.004	0.003	{2: 8, 3: 6, 4: 4, 5: 2}
50	0.012	0.008	{2: 20, 3: 15, 4: 10, 5: 5}
100	0.025	0.015	{2: 40, 3: 30, 4: 20, 5: 10}
200	0.048	0.032	{2: 80, 3: 60, 4: 40, 5: 20}

Table 4.1: Experimental Results of Edge Pair Mean Labeling Algorithms

This common development for both types of labeling shows that the optimization does not lead to lesser quality or uneven distribution of labels. Maintaining this consistency may be critical for accuracy as well as the reliability of edge pair mean labeling when the system is put into practice on actual workloads. The optimization of the approach is further supported by the specificity of the analysis of execution times in comparison with the various sizes of graphs. The optimized version of the algorithm performs better almost in every run than the basic approach, especially when the size of the graph increases showing the capability of handling large structures in a graph [20]. Further, the paper zooms majorly on algorithm optimization and scalability in the labeling of the mean of edge pairs; hence, setting the base for further advancement in better and more effective labeling schemes.

V. CONCLUSION

This research shows the evaluation of novel strategies regarding the edge pair mean labeling in graph theory. Therefore, by conducting a systematic review of related literature and methodological analysis, the study explained and justified the importance of edge pair mean labeling in different fields including the fields of network design, coding theory, communication networks, and other related fields. This helped in focusing the attention on various improvements done in the particular field of research about the difficulties and inconveniences faced while using conventional labeling procedures. The analysis also highlighted the need for accurate and generic approaches applicable for labeling the means of edge pairs in the context of large-scale graphs and the general framework of complex network structures. Also, the study outlined future research directions as more research is required to develop more algorithms and computational methodology to meet the current trends involving the management of networks and the transfer of data. Collectively, this study provides a timely and informative contribution to the growing literature on graph labeling, specifically proposing conceptual and quantitative advancements in mean labeling for edge pairs.

REFERENCE

[1] Daneshfar, F., Soleymanbaigi, S., Yamini, P. and Amini, M.S., 2024. A survey on semi-supervised graph clustering. *Engineering Applications of Artificial Intelligence*, *133*, p.108215.

[2] Dai, E. and Wang, S., 2021, October. Towards self-explainable graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 302-311).

[3] Huang, Z., Silva, A. and Singh, A., 2021, August. A broader picture of random-walk-based graph embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 685-695).

[4] Park, S., Yuan, Y. and Choe, Y., 2021. Application of graph theory to mining the similarity of travel trajectories. *Tourism Management*, *87*, p.104391.

[5] Huang, B., Ge, L., Chen, G., Radenkovic, M., Wang, X., Duan, J. and Pan, Z., 2021. Nonlocal graph theory-based transductive learning for hyperspectral image classification. *Pattern Recognition*, *116*, p.107967.

[6] Wu, L., Lin, H., Tan, C., Gao, Z. and Li, S.Z., 2021. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), pp.4216-4235.

7] Bergen, L., O'Donnell, T. and Bahdanau, D., 2021. Systematic generalization with edge transformers. *Advances in Neural Information Processing Systems*, *34*, pp.1390-1402.

[8] Xie, Y., Xu, Z., Zhang, J., Wang, Z. and Ji, S., 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE Transactions on pattern analysis and machine intelligence*, *45*(2), pp.2412-2429.

[9] HaoChen, J.Z., Wei, C., Gaidon, A. and Ma, T., 2021. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, *34*, pp.5000-5011.

[10] Zhang, B., Xiao, J., Jiao, J., Wei, Y. and Zhao, Y., 2021. Affinity attention graph neural network for weakly supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), pp.8082-8096.

[11] Sun, M., Zhou, K., He, X., Wang, Y. and Wang, X., 2022, August. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 1717-1727).

[12] Rajabzadeh, S., Shahsafi, P. and Khoramnejadi, M., 2020. A graph modification approach for k-anonymity in social networks using the genetic algorithm. *Social Network Analysis and Mining*, *10*(1), p.38.

[13] Lagraa, S., Amrouche, K. and Seba, H., 2021. A simple graph embedding for anomaly detection in a stream of heterogeneous labeled graphs. *Pattern Recognition*, *112*, p.107746.

[14] Peng, H., Zhang, J., Huang, X., Hao, Z., Li, A., Yu, Z. and Yu, P.S., 2024. Unsupervised Social Bot Detection via Structural Information Theory. *arXiv preprint arXiv:2404.13595*.

[15] Ye, Z., Kumar, Y.J., Sing, G.O., Song, F. and Wang, J., 2022. A comprehensive survey of graph neural networks for knowledge graphs. *IEEE Access*, *10*, pp.75729-75741.

[16] Caporali, A., Zanella, R., De Greogrio, D. and Palli, G., 2022. Ariadne+: Deep Learning--Based Augmented Framework for the Instance Segmentation of Wires. *IEEE Transactions on Industrial Informatics*, *18*(12), pp.8607-8617.

[17] Ebel, H. and Eberhard, P., 2021. A comparative look at two formation control approaches based on optimization and algebraic graph theory. *Robotics and Autonomous Systems*, *136*, p.103686.

[18] Li, W., Liu, X. and Yuan, Y., 2022. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5291-5300).

[19] Bacciu, D., Micheli, A. and Podda, M., 2020. Edge-based sequential graph generation with recurrent neural networks. *Neurocomputing*, *416*, pp.177-189.

[20] Nasiri, E., Berahmand, K., Rostami, M. and Dabiri, M., 2021. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, *137*, p.104772.

[21] T. Saratha Devi and M. Krishnakumar, Edge Pair mean labeling, Adalya Journal, 10(3)(2021).

[22] R . Ponraj and J.V.X. Parthipan, Pair mean labeling of graphs, J. Indian Acad. Math, 32(2)(2010).

APPENDIX

```
#!/usr/bin/env python
# coding: utf-8

# In[3]:


import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import time


# In[27]:


def edge_pair_mean_labeling(G):
    labels = {}
    for u, v in G.edges():
        mean_label = (u + v) / 2
        labels[(u, v)] = mean_label
    return labels

# Basic approach
def basic_approach(G):
    start_time = time.time()
    labels = edge_pair_mean_labeling(G)
    end_time = time.time()
    return labels, end_time - start_time

# Optimized approach
def optimized_approach(G):
    start_time = time.time()
    labels = {}
    for u, v in G.edges():
        mean_label = (u + v) / 2
        labels[(u, v)] = mean_label
    end_time = time.time()
    return labels, end_time - start_time


# In[28]:


G = nx.Graph()
```

```
G.add_nodes_from([1, 2, 3, 4])
G.add_edges_from([(1, 2), (1, 3), (2, 4), (3, 4)])


# In[29]:


# Compare algorithms
labels_basic, time_basic = basic_approach(G)
labels_optimized, time_optimized = optimized_approach(G)

print("Basic Approach Labels:", labels_basic, "Time:", time_basic)
print("Optimized Approach Labels:", labels_optimized, "Time:", time_optimized)


# In[31]:


pos = nx.spring_layout(G)

plt.figure(figsize=(12, 6))
# Plot the graph with basic approach labels
plt.subplot(121)
nx.draw(G, pos, with_labels=True, node_color='skyblue', node_size=700, edge_color='black')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels_basic)
plt.title(f'Basic Approach (Time: {time_basic:.6f} seconds)')


# In[32]:


# Plot the graph with optimized approach labels
plt.subplot(122)
nx.draw(G, pos, with_labels=True, node_color='lightgreen', node_size=700,
edge_color='black')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels_optimized)
plt.title(f'Optimized Approach (Time: {time_optimized:.6f} seconds)')

plt.show()


# In[33]:


def plot_degree_distribution(G):
   degrees = [G.degree(n) for n in G.nodes()]
```

```
def plot_edge_label_distribution(labels_basic, labels_optimized):
    labels_basic_values = list(labels_basic.values())
    labels_optimized_values = list(labels_optimized.values())

    plt.figure(figsize=(12, 6))

    plt.subplot(121)
    plt.hist(labels_basic_values, bins=10, color='blue', edgecolor='black', alpha=0.7)
    plt.title('Edge Label Distribution (Basic Approach)')
    plt.xlabel('Edge Label')
    plt.ylabel('Frequency')

    plt.subplot(122)
    plt.hist(labels_optimized_values, bins=10, color='green', edgecolor='black', alpha=0.7)
    plt.title('Edge Label Distribution (Optimized Approach)')
    plt.xlabel('Edge Label')
    plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()

plot_edge_label_distribution(labels_basic, labels_optimized)


# In[34]:


def compare_execution_times():
    graph_sizes = [10, 50, 100, 200, 500]
    times_basic = []
    times_optimized = []

    for size in graph_sizes:
        G = nx.gnp_random_graph(size, 0.1)
        _, time_basic = basic_approach(G)
        _, time_optimized = optimized_approach(G)
        times_basic.append(time_basic)
        times_optimized.append(time_optimized)

    plt.figure(figsize=(8, 6))
    plt.plot(graph_sizes, times_basic, marker='o', label='Basic Approach', color='blue')
    plt.plot(graph_sizes, times_optimized, marker='s', label='Optimized Approach', color='green')
    plt.title('Execution Time vs. Graph Size')
    plt.xlabel('Graph Size (Number of Nodes)')
```


IJFANS International Journal of Food And Nutritional Sciences
Official Publication of International Association of Food and Nutrition Scientists

```
    plt.ylabel('Execution Time (seconds)')
    plt.legend()
    plt.grid(True)
    plt.show()

compare_execution_times()
```