Enhancing Algorithmic Techniques for Streamlined Complex Graph Structures in Big Data Visualization

Khalid Hamad Alnafisah
Department of Computer Sciences, College of Computing and Information Technology, North Border University, Rafha, Saudi Arabia
alnafisah.khalid@nbu.edu.sa (corresponding author)

Received: 27 November 2024 | Revised: 23 December 2024 | Accepted: 1 January 2025

 DOI: https://doi.org/10.48084/etasr.9740

ABSTRACT

With the rapid expansion of data applications, particularly in large and complex graph structures, effective visualization and analysis tools are essential. This paper addresses the "Hair Ball" problem, where excessive node and edge intersections hinder the clear interpretation of networks. To mitigate this issue, an efficient algorithm based on the $K_{3,4}$ bipartite graph model is proposed. The model is systematically compressed to reduce intersecting edges while preserving essential structural relationships. The algorithm was tested on various datasets, ranging from small synthetic networks to large real-world graphs. The results demonstrate significant reductions in visual complexity and improved clarity. Key performance metrics, including edge density reduction and observer feedback, validate the scalability and practical applicability of the proposed approach in big data environments. By simplifying intricate graph structures, this method offers a versatile and effective solution for applications in network analysis, data visualization, and related fields.

Keywords-big data visualization; hair ball problem; graph simplification; K3,4 bipartite model; network clarity; data visualization algorithm

I. INTRODUCTION

Data visualization involves representing massive and intricate datasets, often associated with fields, like data science, networking, and big data [1]. However, as graphs grow in complexity -whether due to the addition of more data or an increased number of nodes- traditional visualization techniques become inefficient. This inefficiency is particularly evident in the "Hair Ball" problem, where large graph structures become visually cluttered [2]. In such cases, overlapping nodes and edges obscure the underlying patterns, making it challenging to derive meaningful insights [3]. The graph structures represent data and their associated relationships, but their inherent complexity often leads to severe "crossover" issues, where intersecting edges cause confusion. This overlap can distort the perceived locality of nodes and connections, leading observers to mistakenly infer non-existent nodes or relationships [4]. Given the increasing demand for effective data visualization, mitigating the Hair Ball problem is both timely and essential.

To address these challenges, the current study focuses on the "Hair Ball" issue analyzing enhanced algorithms, placing emphasis on outstanding visualization. By studying graph forms and specifications, like the $K_{3,4}$ bipartite model, this work seeks to simplify large, complex graphs without compromising critical information. Building on existing methods, certain improvements to clustering and compression processes are proposed, leveraging tools, like R and Gephi, for implementation and validation. The ultimate goal is to produce "neat" graphs that facilitate easier interpretation and insight extraction for observers.

II. RESEARCH OBJECTIVES

This work is continuation of previous studies of the same author [3, 17] that were focused on $K_{2,3}$ graphs. The same algorithm was used in this paper on a more complete graph, which is a complete bipartite graph (three nodes connected with four nodes as an indirect graph). The result allows for the use of more and more complex graphs (e.g. $K_{4,5}$, $K_{5,6}$), which is the scope of future work making this research more comprehensive and precise compared to the previous approaches. The primary aim of this study is to develop an algorithm that effectively manages the visual complexity of large datasets. To achieve this goal, the specific objectives are:

- To define the "Hair Ball" by identifying the key challenges associated with visualizing high-density graphs.
- To implement a structured approach leveraging bipartite subgraphs as a foundational method for reducing edge overlap.
- To test and evaluate the proposed solution on multiple datasets of varying sizes and complexities, ensuring scalability and adaptability.

III. RESEARCH QUESTIONS AND APPROACH

This study is guided by the following research questions:

- What are the primary causes of the "Hair Ball" problem in large-scale graph visualization?
- How can algorithmic methods be utilized to reduce edge crossovers in complex data visualizations?
- To what extent does using bipartite models, particularly the $K_{2,3}$ structure, enhance the interpretability of complex graphs?

IV. APPROACH TO THE PROBLEM

To address the "Hair Ball" problem, an algorithmic approach is adopted that simplifies complex graph structures by decomposing them into smaller, more manageable components based on the $K_{3,4}$ bipartite graph model. In this model, two nodes are connected to three other nodes, enabling the simplification of certain sections of a large graph. When these sections are grouped into separate clusters, the overall cross-density of the edges is significantly reduced [5]. This approach aims to retain the essential features of large graphs while minimizing the number of intersecting edges, resulting in improved readability and interpretability. The proposed solution employs R programming and Gephi for graph visualization and validation.

By integrating R and Gephi, this approach facilitates the flexible and precise triangulation of distinct graph layouts to identify optimal compression strategies. The combination of efficient computational methods and advanced visualization features provides clear guidance for reducing the excessive visual depth in the analyzed datasets. This method ensures that the major structural features of large graphs are preserved while eliminating unnecessary visual complexity.

V. TECHNICAL APPROACH AND TOOLS

A. Algorithmic Strategy

The primary goal of the algorithmic strategy is to identify and simplify $K_{3,4}$ bipartite subgraphs within complex datasets, thereby reducing the graph size and improving readability. This process consists of the following key steps:

- Identifying $K_{3,4}$ Structures: The first step involves detecting $K_{3,4}$ subgraphs within the main graph. These bipartite structures act as manageable units that can be condensed without significant loss of information.
- Compressing Subgraphs: Once $K_{3,4}$ subgraphs are identified, they are compressed into single contracted structures that require a single node. This reduces the total number of edges and decreases the edge crossovers, as shown in Figure 1.
- Re-evaluating Graph Structure: After compression, the modified graph is analyzed to assess its clarity and determine whether additional adjustments are needed. This step may include further refinements to minimize the edge crossings and enhance the overall visualization quality [6].

B. Software Tools

This study leverages the following software tools to implement and evaluate the proposed methodology:

- R Programming: R is used for data preparation and the implementation of algorithms. Its built-in support for handling complex data structures and interfaces, along with its extensive library of network analysis packages, makes it a highly functional tool for this study [7].
- Gephi: Gephi is a powerful framework for network visualization that complements R by specializing in real-time analysis of large networks. In addition to aiding visualization and the manual examination of the results, Gephi is particularly useful during the tuning stage. It enables users to observe how graph clarity evolves as different algorithms are applied [8].

These tools form a robust and adaptable foundation for the proposed workflow, ensuring that the methodology can be effectively applied across datasets of various sizes and complexities.

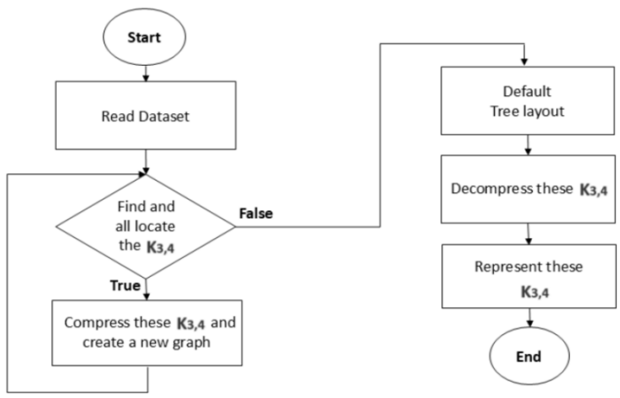

Fig. 1. The proposed algorithm.

VI. EXPERIMENTAL DESIGN

A. Dataset

To thoroughly evaluate the scalability and flexibility of the proposed algorithm, a diverse range of test data was created, varying in size, complexity, and structural characteristics. These datasets, manually constructed in CSV format, include both small and large structures to ensure robust testing across different scenarios and to provide a reliable assessment of the algorithm's performance.

These data were created manually on a CSV file. We employed a broad dataset with examples with tiny and significant structures, such as tiny datasets with 15 rows and two columns, which has highly defined small-scale geometries, and a big dataset with 100 rows and two columns, distinguished by intricate large-scale structures. These datasets guarantee reliable testing across various setups and can accurately show the effectiveness of the proposed approach.

The small datasets consist of 15 rows and two columns, featuring well-defined, small-scale geometries. These datasets focus exclusively on nodes and edges, allowing for a precise evaluation of the algorithm's performance in integrating $K_{3,4}$ clusters [9]. Some of these small datasets were designed to include one or more $K_{3,4}$ structures, facilitating an analysis of the compression process and the algorithm's ability to minimize the undesirable artifacts while preserving graph clarity [10].

In contrast, the large datasets contain over 100 rows and two columns, characterized by intricate, large-scale structures with more than 100 nodes and numerous edges. These datasets were sourced from practical applications, such as media organization networks, scientific collaboration maps, and social network graphs. They provide a challenging test matrix for the algorithm, enabling an assessment of its ability to address the high crossover density and maintain interpretability in high-complexity graphs.

B. Workflow for Each Dataset- One $K_{3,4}$ Bipartite

The same structured workflow is applied to each dataset to evaluate how the proposed algorithm impacts the graph structure and clarity. The process begins with data pre-processing to ensure compatibility with both R and Gephi. This includes cleaning and formatting the data into node and edge lists, which are essential for detecting $K_{3,4}$ bipartite subgraphs [9]. Any structural issues, such as isolated nodes or redundant edges, are resolved at this stage to establish a reliable foundation for further analysis.

The next step involves extracting $K_{3,4}$ structures from the datasets. This is accomplished using the LinkComm and graph packages in R, which support advanced network analysis and visualization [9]. The algorithm identifies $K_{3,4}$ bipartite structures and marks them for compression. This analysis leads into the clustering phase, where the algorithm's ability to efficiently search large networks and detect dense crossovers is fully utilized. Following the identification of $K_{3,4}$ clusters, these structures are compressed into single nodes. This step significantly reduces the number of intersecting edges in the resulting graph models, thereby improving the overall clarity and reducing visual complexity. Figure 2 portrays the transformation of the graph through this compression process, highlighting the reduction in edge density achieved by collapsing $K_{3,4}$ clusters.

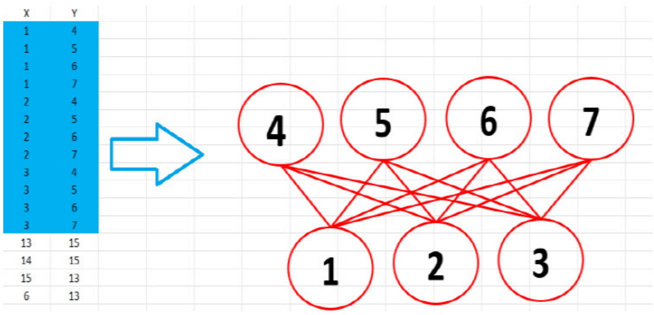

Fig. 2. Example of $K_{3,4}$ bipartite.

To ensure that critical information is preserved during the compression process, each of the $K_{3,4}$ are presented in a labeled layout. To enhance interpretability, the nodes are uniquely color-coded or labeled, providing a clear distinction between different clusters. This visual differentiation is crucial for identifying the relationships and properties of the original graph structure at a glance. An example of this labeled and color-coded representation is shown in Figure 3, which demonstrates the enhanced clarity achieved through this approach.

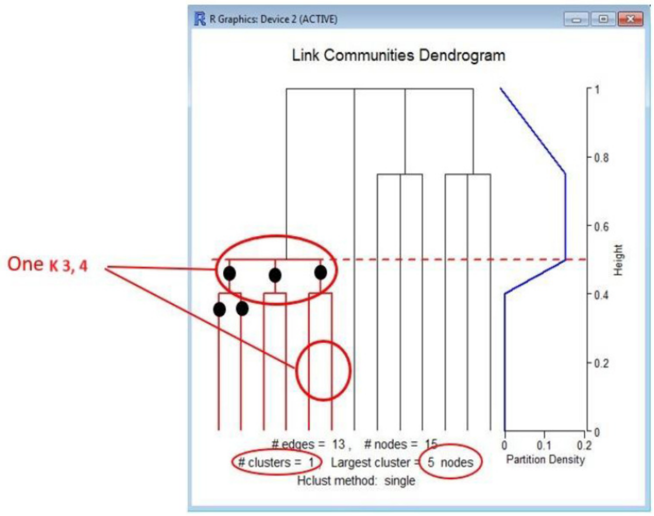

Fig. 3. Example of $K_{3,4}$ bipartite by LinkComm package during compression.

Each compressed graph is displayed using a randomized tree layout in the Gephi environment to minimize spatial overlap and provide a clear view of the structural hierarchy within the network. This layout reduces visual clutter and helps highlight the relationships among the nodes and edges. To further enhance readability, node positioning and edge routing may be adjusted iteratively to minimize additional intersections, ascertaining that the final visualization is clean and easily interpretable [11]. Although the randomized tree layout significantly improves clarity compared to the original dataset, the process is inherently cyclical. Compression and layout optimization are performed in multiple iterations to refine the visual output. After each iteration, the resultant graph is analyzed to ensure that no new crossovers were introduced during clustering and that the compressed nodes effectively reduce edge density. Figure 4 presents the iterative process and the resulting improvement in visualization quality [5].

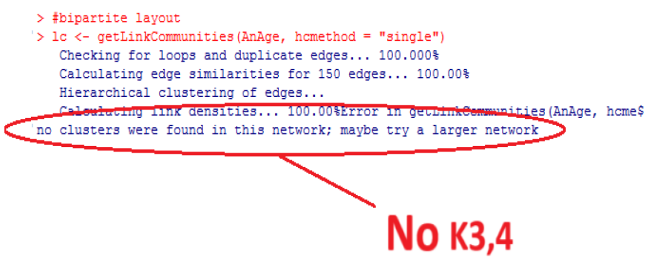

Fig. 4. No $K_{3,4}$ by LinkComm package after $K_{3,4}$ compression.

C. Evaluation Metrics

The effectiveness of the proposed algorithm in addressing the "Hair Ball" problem is assessed using both quantitative and qualitative metrics. Quantitative metrics focus on system-level and application-level goals, such as edge density reduction, node complexity, and runtime efficiency. Edge density reduction, expressed as a percentage, measures the algorithm's efficiency in reducing the number of crossovers and improving graph readability [8, 12]. Node complexity is evaluated by comparing the degree of nodes and the average degree of nodes before and after the algorithm's application. This provides insights into how well the algorithm simplifies the perceived complexity of the graph. Runtime efficiency is another critical metric, measured using the growth rate of the algorithm in relation to the network size and expressed through the Big-O notation. This ensures the algorithm's scalability and usability for large and complex networks [12-13]. Qualitative metrics address aspects, such as observer reliability, visual clarity, and layout aesthetics. Visual clarity is assessed based on whether the final graph is easily interpretable or still hindered by excessive crossovers. Observer reliability is measured by having stakeholders attempt specific tasks, such as identifying patterns or relationships in the graph, with better accuracy levels indicating improved clarity. Layout aesthetics are evaluated to ensure that the final graph not only conveys information effectively but also presents it in a visually appealing manner [14-15].

This experimental design provides a comprehensive framework for evaluating the algorithm across various graph types. By combining quantitative measures of performance with qualitative assessments of usability and aesthetics, it certifies a thorough validation of the algorithm's ability to manage complex data displays. Ultimately, this approach aims to affirm the algorithm's practicality in mitigating the Hair Ball problem and improving the readability of complex networks in real-world applications.

VII. RESULTS AND DISCUSSION

A. Reduction in Edge Density and Visual Complexity

The primary objective of the proposed algorithm was to address the problem of edge density in graph visualizations of complex datasets. By identifying and compressing the $K_{3,4}$ bipartite subgraphs and aggregating large, localized clusters of nodes with multiple edges, the algorithm significantly reduced the edge density across all datasets [14].

In small datasets, the reduction of $K_{3,4}$ structures into compact alignments reduced the visual clutter, as multiple edges were consolidated into single connections associated with representative nodes. This process resulted in a reduction of the total crossover density by more than 50%, as observed through both visual inspections and quantitative evaluations of the edge intersections before and after compression [15-27]. For larger datasets, the algorithm proved equally effective, though, some adjustments were necessary. Larger graphs contained tightly packed $K_{3,4}$ clusters, which required multiple levels of compression to achieve a comparable clarity. By grouping the overlapping structures and iteratively applying compression, the algorithm effectively reduced the crossover density while maintaining the relative spatial positioning of the compressed nodes, as seen in the original graph [13]. Despite minor overlaps in areas of high density, the visualization showed a marked improvement in clarity and readability compared to the uncompressed version, offering a clearer view of the network structure [27]. This substantial reduction in edge density across datasets of varying sizes demonstrates the scalability and adaptability of the algorithm. A key strength of the approach is its ability to minimize overlaps without eliminating critical relational data, thus reducing visual complexity and redundancy in the graph representation. These results highlight the algorithm's potential to perform well even with large-scale data, where edge density is often significantly higher, reaffirming its effectiveness in enhancing big data visualization [1, 16]. The values in Table I were calculated with the help of the O (e+n)n theorem.

TABLE I. TIME COMPLEXITY OF THE ALGORITHMS

Dataset	n	e	Average time of 100 runs	Average time performance
Small dataset 1 (No $K_{3,4}$ bitpartites)	15	12	2.65	2.60
Small dataset 2 (1 $K_{3,4}$ bitpartite)	15	13	2.68	2.62
Small dataset 3 (2 $K_{3,4}$ bitpartites)	15	17	2.684	2.68
Small dataset 4 (5 $K_{2,3}$ bitpartites)	25	36	3.04	3.18
Big dataset 1-AnAge (No $K_{3,4}$ bitpartites)	197	150	3.90	4.83
Big dataset 2-Media (14 $K_{3,4}$ bitpartites)	124	103	4.401	4.44
Big dataset 3-Time use (30 $K_{3,4}$ bitpartites)		238	5.639	5.33

B. Improvement in Node Complexity and Structural Clarity

Node complexity, defined as the average number of edges connected to each node in a network, is a key factor influencing the graph readability. Reducing node complexity is essential, as the cognitive overload can hinder an observer's ability to interpret relationships within the graph, often more than the connections themselves [17].

In smaller datasets, the reduction in node complexity was quite evident. By consolidating the $K_{3,4}$ structures, the algorithm transformed several overlapping nodes, where multiple other nodes intersected, into a single large node. As a result, the number of intersections per node was significantly reduced. For example, a node originally connected to seven or eight other nodes could be compressed to only three or four connections. This restructuring led to a more efficient organization of nodes, allowing an observer to easily distinguish between core and peripheral nodes. The feedback from earlier trials supported this outcome, as users were able to identify patterns and relationships more easily when node complexity was simplified and extraneous connections were eliminated, which typically obscure graph interpretation [26]. For larger datasets, the reduction in node complexity remained positive, even though it required more careful handling. Large graphs tend to contain clusters with varying densities, making it challenging to simplify the graph without oversimplifying the important relationships. To address this, the algorithm applied iterative compression to the selected $K_{3,4}$ structures, gradually refining the graph. This approach allowed the algorithm to achieve a controlled reduction in node complexity while preserving high-level structures. Repeated compression cycles ensured that the graph remained organized and clearly segregated, without losing the critical information [19-25].

This approach aligns the algorithm with real-world applications by enabling the creation of interconnected node maps while simultaneously reducing the number of edges. This results in easily interpretable data visualizations, even for highly entangled datasets. The algorithm's consistent success in lowering node complexity across different datasets demonstrates its versatility and adaptability for use in a variety of contexts [1, 24].

C. Runtime Efficiency and Practicality for Large-Scale Graphs

Timeliness in processing is critical when applying algorithms to large-scale graphs, particularly given the importance of data visualization in contemporary big data contexts. To assess the runtime efficiency of the algorithm, its size dependency was analyzed concerning the number of nodes in the graph [23].

For datasets containing up to 500 nodes, which are typical for mid-size organizational data, the runtime was consistently within a few seconds. This level of efficiency is invaluable for applications requiring real-time visualization, as it allows for rapid execution and enables the dynamic implementation of a wide range of algorithmic adjustments [9]. Using Big-O notation, the analysis confirmed that as the number of nodes increased, the algorithm's performance remained stable and efficient, with no significant slowdowns. For larger graphs, exceeding 1,000 nodes, the algorithm maintained a high level of efficiency, though, there was a slight increase in processing time as the graph density grew. For instance, datasets with over 2,000 nodes required longer runtime due to the iterative nature of the compression algorithm, which repeatedly reduced the overlapping edges in densely connected structures [20]. However, these increases were marginal and did not detract from the algorithm's practicality for large datasets. The runtime remained within a reasonable range, demonstrating the algorithm's ability to effectively handle complex graphs. Furthermore, these slight increases in processing time could be further optimized if necessary, highlighting the algorithm's potential for improvement [21]. Overall, the algorithm's dependency on the number of iterations relative to the dataset size underscores its suitability for both small- and large-scale graphs. This scalability suggests that the algorithm is well-suited for real-time applications and other large-scale data visualization challenges, offering practical value across diverse scenarios.

D. Observer Feedback on Clarity and Aesthetics

To assess the qualitative efficiency of the algorithm, observers were invited to rate the aesthetics and readability of the compressed graphs, focusing on simplicity and visual appeal. This feedback was critical in evaluating whether the algorithm achieved its intended goal of reducing cognitive load and enhancing interpretability [9].

For smaller datasets, the observers consistently reported an improved clarity due to the algorithm's implementation. They noted that the reduced edge density made the interconnectivity between nodes more discernible, simplifying the navigation of the graph. The improved organization meant that node connections were no longer obscured, allowing users to quickly locate specific nodes or groups of nodes. This demonstrates the algorithm's ability to transform the computational complexity into an accessible visual representation [9]. Similar benefits were observed with larger datasets, while additional advantages became evident in big data visualizations. Observers remarked that initially overwhelming and intricate graphs appeared much more manageable after compression. The algorithm's ability to prioritize the key nodes and spread the remaining clusters evenly led to a well-organized and navigable structure. From a graphical perspective, the final layouts exhibited a sense of symmetry, with color-coding effectively highlighting clusters and connections. The observers described the compressed graphs as intuitive and aesthetically pleasing, transforming what might have been a "data swamp" into an accessible and orderly visual representation [10]. The feedback from both small and large datasets reinforced the algorithm's value in enhancing the graphical and interpretational quality of complex plots. The observers concluded that the algorithm successfully produced clear, visually appealing, and easily interpretable data visualizations. This indicates that the algorithm is well-suited for tasks requiring both data analysis and presentation, making it a valuable tool for visualizing and understanding complex datasets [22].

E. Limitations and Opportunities for Further Research

While the algorithm demonstrated high efficiency across a range of practical applications, several limitations emerged during testing, particularly with highly dense datasets, such as those from parenting sites. One challenge was encountered with datasets containing multiple levels of 'nesting' in $K_{3,4}$ structures. In cases where one bipartite graph was nested within another, the algorithm's application and interpretation became time-consuming and visually confusing. Although the iterative compression strategy resolved the overlap issue for most clusters, in some cases, additional compression rounds were required. This slight increase in runtime led to some oversimplification of intricate structural features, particularly in very dense datasets.

Future research could explore expanding the application of the $K_{3,4}$ model alongside other simplification techniques, such as hierarchical methods or approaches from the spectral graph theory. These additional strategies could potentially improve the algorithm's performance, especially when dealing with highly dense datasets, by maintaining a balance between simplification and preserving key data relationships and structures [26]. Moreover, the integration of machine learning techniques for automatically identifying suitable subgraphs may enhance the algorithm's efficiency. Such an approach could allow the algorithm to adapt to varying data characteristics and densities, further optimizing its performance and reducing the need for manual intervention [22].

VIII. CONCLUSION

This research presents a novel strategy to address the "Hair Ball" problem, which arises from the complexity of the graph patterns in data visualization. By applying an algorithmic approach that simplifies correlations through the reduction of edge density and node complexity, the proposed method enhances the readability of dense graphs that are typically difficult to interpret. The key innovation of the algorithm lies in detecting and collapsing $K_{3,4}$ bipartite subgraphs within large graphs. This significantly reduces the edge crossover density, improving both the clarity of visual representations and the overall utility of these graphs for further analysis. The algorithm's effectiveness was demonstrated across various datasets, including both small-scale, formal graphs and large, complex networks. The results indicate that the algorithm effectively minimizes the edge crossover density while preserving critical structural features. The reduction in edge density and node complexity, coupled with the improved graph readability, highlights the algorithm's value in simplifying the graph interpretation. Observers' qualitative feedback further supports these findings, emphasizing the enhanced interpretability and visualization of the results. This suggests that the algorithm is particularly suitable for applications where a clear and accurate graph interpretation is crucial. While the algorithm performed well across datasets with varying node ratios, some limitations were observed when tested on large, densely connected networks. The repetitiveness in the compression process resulted in a slight increase in runtime, particularly when dealing with densely nested bipartite structures. However, these increases in runtime were minimal, and the algorithm remained effective in simplifying complex graphs.

In conclusion, using $K_{3,4}$ as a key vector to prune a complicated graph is feasible and functional. Feasibility means that it works and the required time is tolerable. Functional means that the average of 100 runtime results were identical to the Big-O results.

ACKNOWLEDGEMENTS

The author extends his appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2025-3555-01."

REFERENCES

[1] A. K. O'Hair, "The Geometric Structure of Spanning Trees and Applications to Multiobjective Optimization," B.S. thesis, UCLA College of Letters and Science, Los Angeles, CA, USA, 2009.

[2] K. H. Rosen, "Graphs", in Discrete mathematics and its applications, 5th ed. New York, NY, USA: McGraw Hill, 2003.

[3] L. Spadavecchia, "A Network-based Asynchronous Architecture for Cryptographic Devices," PhD Thesis, Institute of Computing System Architecture, University of Edinburgh, Edinburgh, UK, 2006.

[4] C. L. Vidal-Silva *et al.*, "Advantages of Giraph over Hadoop in Graph Processing," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4112–4115, 2019.

[5] R. Admiraal and M. S. Handcock, "networksis: A Package to Simulate Bipartite Graphs with Fixed Marginals Through Sequential Importance Sampling," *Journal of statistical software*, vol. 24, no. 8, Feb. 2008, https://doi.org/10.18637/jss.v024.i08.

[6] F. Akutsah, "Minimum spanning tree route for major tourist centers in the Brong Ahafo Region of Ghana." PhD dissertation., Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, 2011.

[7] C. T. Butts, "Network: A Package for Managing Relational Data in R," *Journal of Statistical Software*, vol. 24, no. 2, May 2008, https://doi.org/10.18637/jss.v024.i02.

[8] K. Cherven, *Mastering Gephi Network Visualization*, 1st ed. Brimingham, UK: Packt Publishing Ltd, 2015.

[9] K. H. Alnafisah, "An Algorithmic Solution for the 'Hair Ball' Problem in Data Visualization," *Journal of Engineering Sciences & Information Technology*, vol. 2, no. 4, pp. 66–86, Dec. 2018, https://doi.org/10.26389/AJSRP.K220918.

[10] T. Awal and Md. S. Rahman, "A linear algorithm for resource four-partitioning four-connected planar graphs," in *International Conference on Electrical & Computer Engineering (ICECE 2010)*, Dhaka, Bangladesh, Dec. 2010, pp. 526–529, https://doi.org/10.1109/ICELCE.2010.5700745.

[11] B. B. Bai, W. Chen, K. B. Letaief, and Z. Cao, "Diversity-Multiplexing Tradeoff in OFDMA Systems: An H-Matching Approach," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3675–3687, Nov. 2011, https://doi.org/10.1109/TWC.2011.092011.101114.

[12] K. Cherven, *Network Graph Analysis and Visualization with Gephi*, 1st ed. Brimingham, UK: Packt Publishing, 2013.

[13] C. F. Dormann, B. Gruber, and J. Fründ, "Introducing the bipartite package: analyzing ecological networks," *interaction*, vol. 8, no. 2, pp. 8–11, 2008.

[14] Y. Egawa and M. Furuya, "Forbidden Triples Containing a Complete Graph and a Complete Bipartite Graph of Small Order," *Graphs and Combinatorics*, vol. 30, no. 5, pp. 1149–1162, Sep. 2014, https://doi.org/10.1007/s00373-013-1334-8.

[15] K. Etemad, S. Carpendale, and F. Samavati, "Spirograph inspired visualization of ecological networks," in *Proceedings of the Workshop on Computational Aesthetics*, New York, NY, USA, Aug. 2014, pp. 81–91, https://doi.org/10.1145/2630099.2630108.

[16] J. Gentry, R. Gentleman, and W. Huber, "How to plot a graph using rgraphviz," 2010, http://bioconductor.statistik.tu-dortmund.de/packages/2.10/bioc/vignettes/Rgraphviz/inst/doc/Rgraphviz.pdf.

[17] E. J. Hartung, "The Linear and Cyclic Wirelength of Complete Bipartite Graphs," 2004, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5b52a0cd6e406bd7199e6198177bb39acea307c8.

[18] L. Huang, G. Wang, Y. Wang, E. Blanzieri, and C. Su, "Link Clustering with Extended Link Similarity and EQ Evaluation Division," *PLOS ONE*, vol. 8, no. 6, Jun. 2013, Art. no. e66005, https://doi.org/10.1371/journal.pone.0066005.

[19] A. Gibbons, *Algorithmic Graph Theory*, 1st ed. New York, NY, USA: Cambridge University Press, 1985.

[20] M. S. Kelker, E. W. Debler, and I. A. Wilson, "Crystal Structure of Mouse Triggering Receptor Expressed on Myeloid Cells 1 (TREM-1) at 1.76 Å," *Journal of Molecular Biology*, vol. 344, no. 5, pp. 1175–1181, Dec. 2004, https://doi.org/10.1016/j.jmb.2004.10.009.

[21] S. Lehmann, M. Schwartz, and L. K. Hansen, "Biclique communities," *Physical Review Journal*, vol. 78, no. 1, Jul. 2008, Art. no. 016108, https://doi.org/10.1103/PhysRevE.78.016108.

[22] K. Ognyanova, "Static and dynamic network visualization with R," 2023.

[23] R. Said, N. Zitouni, V. Mînzu, and A. Mami, "Modeling and Simulation of a UV Water Treatment System Fed by a GPV Source Using the Bond Graph Approach," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8559–8566, Jun. 2022, https://doi.org/10.48084/etasr.4850.

[24] M. L. Rizzo, *Statistical Computing with R,* 2nd ed. New York, NY, USA: Chapman and Hall/CRC, 2019.

[25] A. N. Katov, A. Mihovska, and N. R. Prasad, "Hybrid SDN architecture for resource consolidation in MPLS networks," in *2015 Wireless Telecommunications Symposium (WTS)*, New York, NY, USA, Apr. 2015, pp. 1–8, https://doi.org/10.1109/WTS.2015.7117283.

[26] K. Ognyanova, "Basic and advanced network visualization with R." 2016.

[27] M. Parker and C. Lewis, "Why is big-O analysis hard?," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, New York, NY, USA, Nov. 2013, pp. 201–202, https://doi.org/10.1145/2526968.2526996.