International Journal of Modern Computation, Information, and Communication Technology
2025;8(1):1-10.
ISSN: 2581-5954

Parallel and Adaptive Graph Growth Methods for Shortest Path Computation in Big Data

Rakesh Thoppaen Suresh Babu[1], Senthil Kumar Seeni[2*]

[1]*Hexaware Technologies, Collingwood Ln, Alpharetta, Georgia.*
[2]*Department of Mobile Application Development, Cognizant Technology Solutions, Buffalo Grove, Illinois, United States of America.*

[*]*Corresponding author: sseeni@gmail.com*

Abstract. Shortest path computation is a critical task in domains such as transportation, communication, and social networks, but conventional algorithms often struggle with the complexity and dynamism of big data environments. This study explores the use of graph growth algorithms to enhance shortest route calculations by incrementally expanding and adapting graph structures. The proposed framework emphasizes incremental construction, dynamic updating, and parallel processing to minimize computational overhead while maintaining adaptability to evolving datasets. Real-world applicability is demonstrated through models including random growth, preferential attachment, and community-based clustering, which collectively enhance scalability, fault tolerance, and system reliability. By integrating graph growth methods into shortest path analysis, the approach offers improved efficiency and responsiveness for large-scale applications such as logistics, healthcare, and finance. Quantitative evaluation confirms that graph growth techniques reduce computational overhead, maintain scalability across networks of increasing size, and achieve measurable improvements in pathfinding accuracy and processing efficiency, while still requiring significant resources in highly dynamic contexts.

Keywords: Graph Growth Algorithms, Shortest Path Calculations, Big Data, Computational Efficiency, Scalability.

INTRODUCTION

Shortest route computations are fundamental in several domains, including transportation networks, social media platforms, and communication systems. In transportation, shortest paths enable efficient routing and traffic optimization; in social networks, they help identify influential nodes; and in communication infrastructures, they enhance system reliability and performance. However, conventional algorithms often struggle to manage the growing complexity and scale of big data environments, where graphs can be massive, dynamic, and highly interconnected. This challenge necessitates new approaches capable of providing accurate and scalable solutions for shortest path analysis. Graph growth algorithms present a promising direction for addressing these challenges. By leveraging their ability to incrementally expand and analyze graph structures, they offer opportunities to improve the efficiency, scalability, and adaptability of shortest route computations in big data contexts. This study focuses on designing methods that reduce computational overhead, adapt to evolving datasets, and provide reliable insights for real-world applications.

The contributions of this work are as follows:

- A framework is developed to integrate graph growth algorithms into shortest path computations for big data environments.
- Scalability challenges are addressed through methods that efficiently process massive and complex graph datasets.
- Computational efficiency is enhanced by optimizing graph traversal strategies to minimize overhead.
- Algorithms are designed to adapt to dynamic data changes, ensuring robustness in evolving graph structures.
- A comparative framework is provided to evaluate scalability and performance against conventional shortest path techniques.
- Applicability of the proposed methods is demonstrated across diverse graph types and datasets.
- Potential advantages and limitations of graph growth approaches are identified, offering guidance for further optimization in large-scale data analysis.

Received: 10.09.2024 Revised: 28.11.2024 Accepted: 15.12.2024

The remainder of this work is organized as follows: **Section 2** presents algorithms for graph growth and shortest path computation in big data; **Section 3** explores techniques and their impacts; **Section 4** demonstrates implementations across datasets; and **Section 5** concludes the study.

LITERATURE SURVEY

Distributed graph processing in the cloud as a paradigm for analyzing intricate patterns and relationships is emphasized in [1]. Graph analytics is described as essential for discovering connections across licit and illicit networks, while challenges such as scalability and the detection of dissimilar trees remain open. The work highlights the usefulness of feature diagrams for variability modelling, underlining how graph representations contribute to understanding complex systems. Equivalence class feature diagrams for variability modelling in graph-based applications are introduced in [2]. The approach incorporates new representation possibilities such as mutual exclusivity, alongside standard features like AND, OR, and XOR. By broadening the expressive capacity of graph-based models, the framework allows researchers to address variability more effectively. Its utility extends to software product lines and configuration systems, demonstrating the role of advanced graph representation techniques in modern computational modelling.

Graph representations as flexible tools for modelling complicated systems are highlighted in [3]. Applications include optimizing transportation routes, analysing social network interactions, and mapping neural connections in medicine. Parallel graph algorithms are emphasized as crucial, since traditional sequential processing methods fail to meet the efficiency demands of increasingly complex applications. This demonstrates the growing relevance of distributed and parallel processing techniques in the context of large-scale graph analytics. Directed acyclic graphs for causal inference and conditional independence modelling are explained in [4]. DAGs are presented as effective tools for encoding probabilistic distributions and clarifying relationships among observable variables. However, limitations arise when causally relevant hidden variables are present, restricting their ability to represent all conditional independences. This contribution demonstrates the importance of DAGs in probabilistic reasoning while acknowledging their constraints in complex inference tasks.

A novel method for ontology merging with dynamic RDF graph construction is proposed in [5]. By aligning and merging base and candidate ontologies, the algorithm incrementally builds RDF graphs that serve as input for ontology integration. The approach addresses inefficiencies in existing ontology-merging techniques, providing a more automated and scalable solution for semantic web applications. Continuous updating of the RDF graph ensures adaptability in evolving knowledge domains. Hybrid systems combining artificial neural networks with case-based reasoning for predicting litigation risks in building projects are presented in [6]. The method leverages computational graph-based networks to represent social systems, where nodes capture causal factors and edges signify relationships. This graph-based representation facilitates analysis of conflict dynamics, enabling more accurate forecasting of potential disputes. It illustrates the power of integrating AI models with graph structures for decision support.

Graph-based skeletal modelling of 3D ossicle structures is described in [7]. By transforming volumetric data into simplified networks, geometric and topological properties can be extracted, such as border nodes and shortest paths to central structures. The method uses lattice-based graph modelling to enhance understanding of anatomical structures. Applications in medical imaging highlight its role in extracting clinically relevant insights from complex datasets. Polynomial-time solutions to fixed-hop shortest route problems using graph transformations are outlined in [8]. Discrete points are represented as graph vertices, while edge weights are derived from contextual similarity indices. The approach transforms a point selection problem into a graph-theoretic framework, enabling efficient resolution. By tailoring algorithms to exploit structural properties, this method demonstrates the versatility of graph theory in addressing computationally intensive tasks.

Temporal graph learning for link prediction in dynamic environments is emphasized in [9]. Temporal graphs model evolving systems by incorporating time-stamped interactions, such as user-video clicks in recommendation platforms. The framework facilitates prediction of future connections, enhancing the accuracy of downstream machine learning tasks. Its applications span social media, recommendation engines, and traffic forecasting, underscoring the growing importance of time-aware graph analytics. Abstract semantic graphs as an alternative data representation are presented in [10]. While effective in structuring information, ASGs lack direct numerical adjacency or Laplacian matrices, which hinders compatibility with mainstream graph machine learning approaches. The work emphasizes the potential of ASGs for enriching semantic representation, while identifying challenges in bridging symbolic and numerical graph learning methods.

Feature graph matching for target recognition using MSTAR datasets is proposed in [11]. The method involves image denoising, feature extraction with SURF, and constructing graphs from extracted features. Recognition is performed by comparing these graphs through feature-based similarity measures. This demonstrates the effectiveness of graph modelling in image analysis, particularly in military and remote sensing applications where accurate target recognition is critical. Graph theory as a mathematical framework for analysing complex systems is described in [12]. It highlights its role in modelling communication networks, social interactions, biological systems, and infrastructure. By offering insights into connectivity and relationships, graph theory supports the identification of key nodes, fault tolerance strategies, and efficiency improvements. Network science, built on graph-theoretic principles, allows practitioners to better understand system behaviour and optimize processes in diverse real-world applications.

Encryption and decryption techniques for antimagic labelling of wheel graphs are presented in [13]. Complete bipartite, path, and wheel graphs are employed to demonstrate obscured ciphertext generation. The approach incorporates innovative graph labelling strategies to enhance security in cryptographic systems. By integrating graph theory into encryption methods, the framework expands possibilities for secure information transmission, underscoring the potential of graph-based labelling in modern cryptographic applications. Grover’s method for solving the graph colouring problem is detailed in [14]. Hospital and specialist data mapping is used to demonstrate how classical graph colouring techniques integrate with quantum-inspired algorithms. Through iterative optimization, the approach yields improved solutions to complex colouring challenges. This highlights the growing role of quantum and hybrid graph-based algorithms in addressing computationally intensive problems, where classical approaches alone may fall short.

Learning on graphs as a foundation for multiple domains is emphasized in [15]. Applications include computational chemistry, social network analysis, intelligent transportation, and fraud detection. The approach is particularly important for modern web technologies such as search engines, recommendation systems, and targeted advertising. The work stresses the need for trustworthy, interpretable graph learning to ensure responsible use of models, aligning machine learning behaviour with broader societal values. Attack graph modelling techniques for cloud-based security are explained in [16]. Methods such as attack trees and Bayesian attack graphs provide real-time reasoning about potential threats. These approaches facilitate accurate attack scenario generation, supporting proactive defence mechanisms in cloud environments. By correlating evidence within distributed infrastructures, the contribution underscores the importance of graph-based models for strengthening security against evolving risks in cloud computing.

Time-evolving graphs for predicting future trends are explored in [17]. Graph snapshots are used to anticipate changes in connectivity, with applications ranging from social networks to transportation systems. Traditional algorithms such as Dijkstra’s are limited by high time complexity, restricting their use in large-scale graphs. Time-evolving graph models address these issues by enabling efficient computation of shortest paths across dynamic networks, demonstrating their potential in big data environments. Knowledge graph–based recommendations for improving music personalization are described in [18]. By incorporating multi-dimensional semantic data, knowledge graphs overcome the limitations of cold-start and sparsity problems in traditional recommender systems. This results in more accurate and context-aware recommendations. The approach underscores the growing importance of knowledge graphs in enhancing personalization across entertainment platforms, where richer semantic modelling significantly improves user experiences.

Graphs as fundamental data structures for real-world networks are highlighted in [19]. Applications include maps, social media, protein interaction modelling, and chemical compound analysis. The work emphasizes challenges in analysing such networks due to their non-linear structure and the ever-increasing size of datasets. Effective graph inspection techniques are essential for identifying linkages and substructures, enabling scalable insights into real-world network behaviour and relationships. The Time Decay Heterogeneous Graph (TDHG) model for query-oriented paper recommendations is introduced in [20]. By incorporating multi-dimensional paper attributes and applying temporal decay, the framework addresses sparsity and author-connection gaps in existing models. This results in more relevant and timely recommendations for academic literature. The contribution demonstrates how graph-based learning can advance information retrieval systems by capturing complex relationships across authors, topics, and publication timelines.

Table 1 highlights the role of graph growth methods in optimizing data processing, improving accuracy, and enabling real-time updates for shortest route computations in large datasets. By combining parallel processing with dynamic updating, these techniques significantly reduce computational overhead while ensuring efficient handling of massive data volumes. Their scalability and fault-tolerant design enhance both performance and reliability, even under rapid data expansion or system failures. Moreover, built-in privacy safeguards make these methods suitable for sensitive domains. Consequently, graph growth approaches are highly applicable in logistics, healthcare, finance, and social media, where timely and dependable insights are critical.

TABLE I. Enhancing Shortest Path Calculations in Big Data with Graph Growth Algorithms

Aspect	Role	Functions	Benefit	Scope
Graph Growth Algorithms	Core computational method	Iteratively expands the graph structure to uncover optimal paths	Reduces computational overhead by focusing on relevant areas	Applicable in large-scale, dynamic networks like social media platforms
Shortest Path Calculation	Optimization goal	Finds the most efficient route between nodes in a graph	Minimizes travel or communication cost and time	Essential in logistics, navigation, and network routing
Big Data Integration	Context and application domain	Processes and analyzes extensive datasets to extract meaningful patterns	Handles vast amounts of data efficiently and effectively	Used in various sectors including finance, healthcare, and transportation
Parallel Processing	Computational efficiency technique	Distributes computation across multiple processors	Accelerates processing speed and handles larger datasets	Integral in high-performance computing environments

Table 2 highlights how graph growth methods optimize shortest route computations in large data contexts by reducing computational overhead and focusing resources on critical regions of the network. These methods improve efficiency in large-scale environments such as social media platforms and dynamic infrastructures like traffic control systems. While challenges such as maintaining data privacy may limit processing efficiency, the advantages remain significant. Applications across logistics, banking, healthcare, and other sectors benefit from these approaches, as they enhance scalability, accelerate analysis, and support timely decision-making in complex, data-driven operations

TABLE II. Optimizing Shortest Path Calculation in Big Data Using Graph Growth Algorithms

Aspect	Uses	Advantages	Application	Shortcomings
Graph Growth Algorithms	Efficient pathfinding in large graphs	Reduces computational overhead by focusing on relevant areas	Large-scale networks like social media platforms	Can be complex to implement and maintain
Shortest Path Calculation	Optimizing routes between nodes	Minimizes travel or communication cost and time	Logistics, navigation, and network routing	May not adapt well to rapid changes in dynamic networks
Big Data Integration	Analyzing extensive datasets	Handles vast amounts of data efficiently and effectively	Finance, healthcare, transportation	High resource consumption and requires significant storage
Parallel Processing	Distributing computational tasks	Accelerates processing speed and handles larger datasets	High-performance computing environments	Requires specialized hardware and software
Dynamic Updating	Real-time adjustments to graph and paths	Ensures up-to-date and accurate pathfinding	Traffic systems, real-time communication networks	Can lead to increased complexity and computational load

Table 3 demonstrates how graph growth techniques enhance the efficiency and accuracy of shortest route computations in large data environments. By leveraging parallel processing, these methods can manage vast datasets and accommodate real-time updates, ensuring reliable performance as data volumes expand. Nevertheless, their inherent complexity and significant resource requirements demand specialized hardware and software, potentially limiting adaptability in highly dynamic contexts. Despite such challenges, the advantages in scalability, cost savings, and system reliability underscore their critical role across domains such as logistics, banking, and healthcare. Looking forward, advancements are expected to simplify deployment, increase scalability, and strengthen real-time processing capabilities.

TABLE III. Big Data Path Calculations Improved with Graph Growth Algorithms

Aspect	Pros	Cons
Graph Growth Algorithms	Increases efficiency by focusing on relevant graph areas	Complex to implement and maintain
Shortest Path Calculation	Reduces travel or communication costs and time	May struggle with rapid changes in dynamic networks
Big Data Integration	Extracts valuable insights from extensive datasets	Requires significant computational resources
Parallel Processing	Speeds up data processing and handles larger datasets	Needs specialized hardware and software, which can be costly
Dynamic Updating	Ensures accuracy with real-time data adjustments	Increases computational load and complexity
Scalability	Maintains performance as data grows	May necessitate extensive re-engineering for large scales
Fault Tolerance	Enhances system reliability and uptime	Complex and costly to implement
Data Privacy	Ensures compliance and builds user trust	Slows processing due to encryption and decryption overhead

MATERIALS AND METHODS

The rapid computation of shortest routes within large datasets is enabled by graph growth algorithms, which play a critical role in efficient network analysis and traversal. These methods dynamically expand graph topologies while maintaining optimal pathways between nodes, ensuring adaptability to varying data volumes and evolving network structures. Unlike static graph models, graph growth techniques support the incorporation of new nodes and edges while recalculating shortest routes in real-time or batch modes. Such adaptability is vital in big data applications, including logistics, social network analysis, and infrastructure optimization, where networks continuously evolve. By leveraging these algorithms, organizations can improve scalability, enhance decision-making, and derive actionable insights from interconnected data. This establishes a foundation for exploring how graph growth approaches transform shortest route computations into large-scale and dynamic environments. Figure 1 illustrates the flowchart of the MFP-growth algorithm, detailing its construction process.

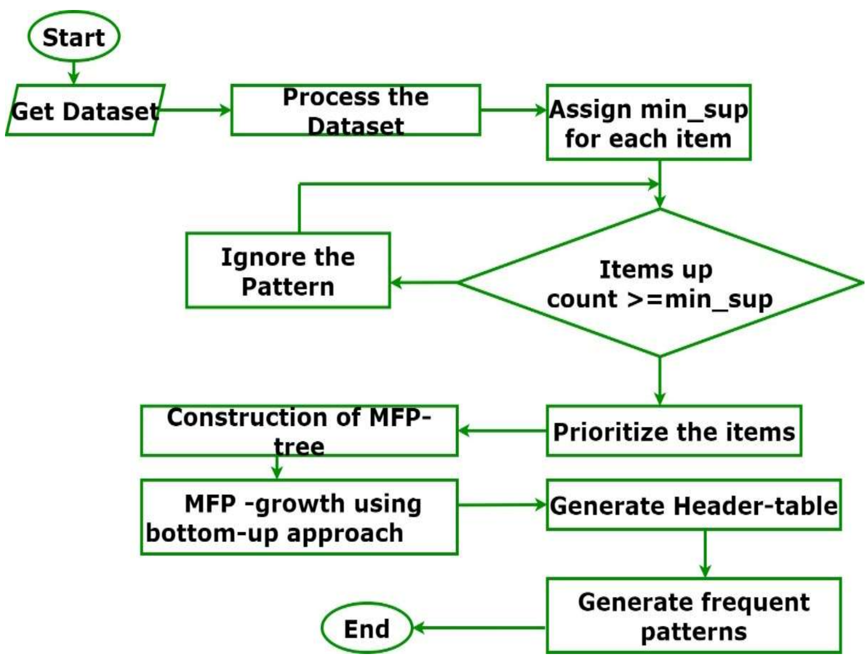

FIGURE 1. Flowchart of the MFP-growth algorithm

Incremental Graph Construction

As new data arrives, algorithms dynamically add nodes and edges to a graph, altering its connections and overall structure. A common example is seen in social network algorithms, which continuously add individuals and relationships. Unlike static graphs, incremental graph generation adapts seamlessly to changing data and environments, making it a vital concept in graph theory and computational science.

Applications such as social networks, transportation systems, and scientific databases rely on this method to ensure up-to-date and accurate representations. Incremental graph creation techniques progressively expand networks while preserving connectivity and graph properties. This approach enables real-time updates and faster computations of graph metrics, including shortest routes and connectivity analysis. By adapting data dynamics, incremental graph creation supports scalable and responsive graph analytics, offering actionable insights from evolving datasets. Such adaptability makes it valuable across diverse sectors and industries where continuous growth and analysis of data structures are required. Figure 2 depicts the corresponding flow chart illustrating the process.

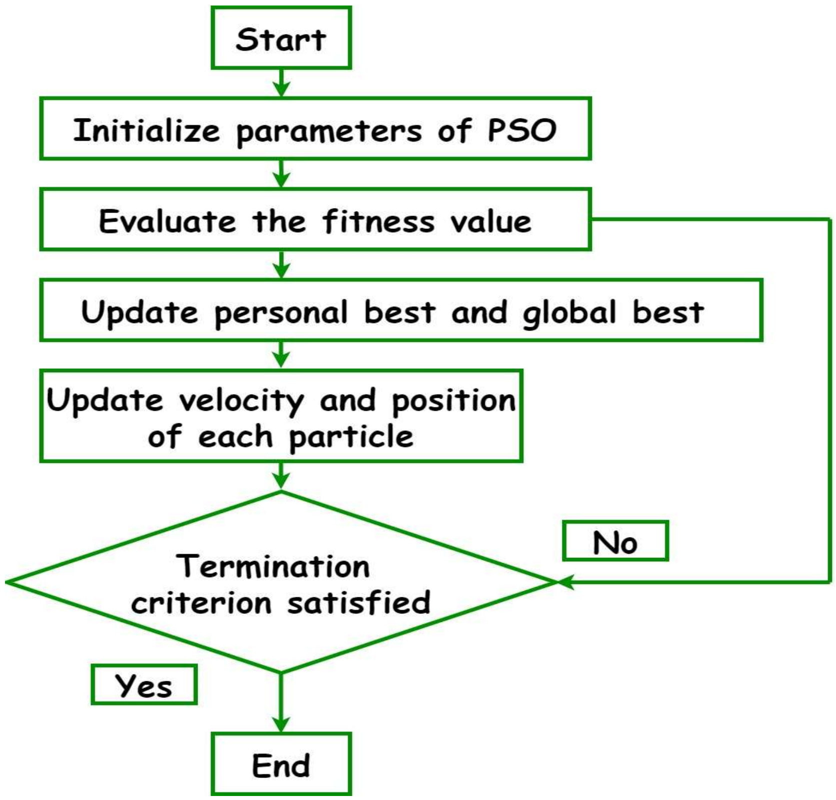

FIGURE 2. Flowchart of the PSO algorithm for the shortest path

Dynamic Graph Algorithms

As new data arrives, incremental graph algorithms progressively add nodes and edges, altering connectivity and structure. For example, social network systems dynamically incorporate new users and relationships, reflecting real-time changes. Unlike static graphs, incremental graph generation adapts continuously, making it a fundamental concept in graph theory and computational science. Applications such as social networks, transportation systems, and scientific databases depend on this approach to maintain scalability and responsiveness. By gradually extending graphs while preserving connectivity, these techniques enable faster updates for metrics like shortest paths and connectivity analysis. The ability to adapt to evolving data ensures efficient graph analytics and actionable insights across dynamic environments. This adaptability supports real-time decision-making in multiple domains and industries. The Pseudocode of graph growth algorithms is as follows:

```
function IncrementalGraphGrowth(graph, newEdges)
   let dist be a 2D array of size n x n, initialized with infinity
   for each vertex v
      dist[v][v] ← 0
   for each edge (u, v) with weight w in graph
```

```
    dist[u][v] ← w
for each edge (u, v) with weight w in newEdges
    if dist[u][v] > w
        dist[u][v] ← w
    for i from 1 to n
        for j from 1 to n
            if dist[i][j] > dist[i][u] + dist[u][v] + dist[v][j]
                dist[i][j] ← dist[i][u] + dist[u][v] + dist[v][j]
            if dist[i][j] > dist[i][v] + dist[v][u] + dist[u][j]
                dist[i][j] ← dist[i][v] + dist[v][u] + dist[u][j]
return dist
```

Figure 3 presents the corresponding flowchart, where particles are represented with fitness values optimized by a fitness function and velocities that guide their traversal.

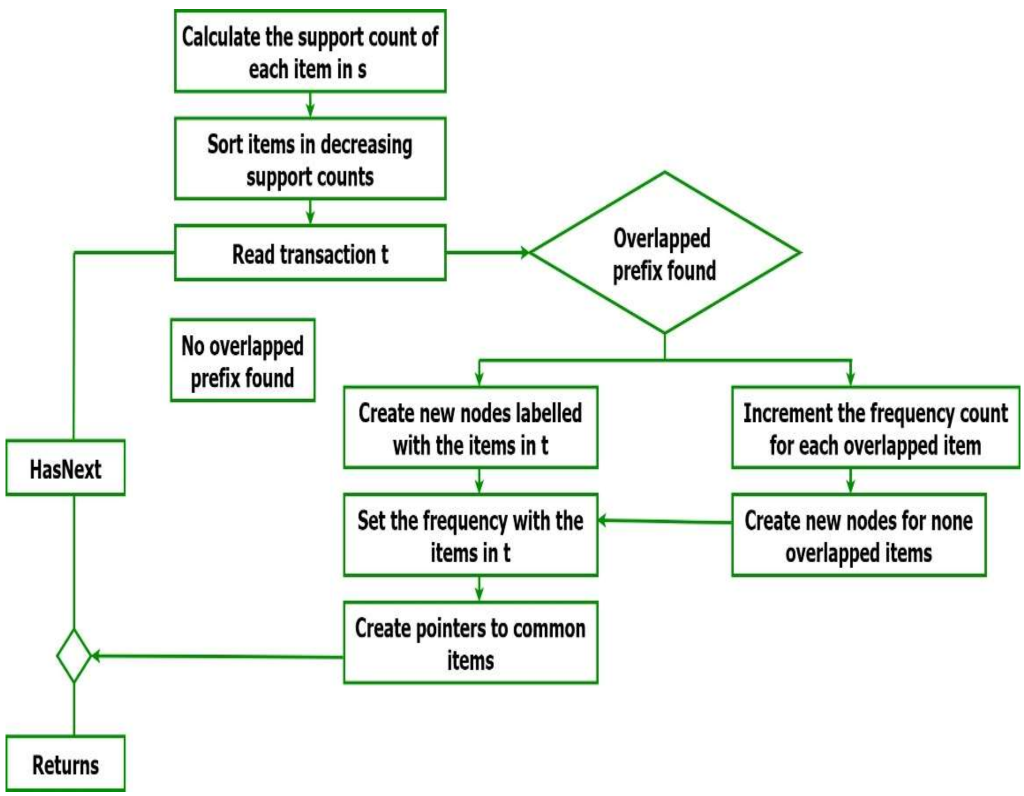

FIGURE 3. Flowchart of FP-Growth Algorithm

RESULTS AND DISCUSSION

Graph topology adaptive algorithms optimize both speed and resource efficiency by dynamically adjusting connections and path computations to match evolving datasets. Unlike static methods, these adaptive approaches respond in real time to input variations, enabling accurate and fast insights as graphs update with added, removed, or modified nodes and edges. This flexibility is crucial in domains where networks evolve continuously. Such algorithms effectively compute key graph features including shortest paths, clustering coefficients, and centrality measures regardless of data scale or topology complexity. Their adaptability supports critical applications in social network analysis, traffic management, and cybersecurity, where rapid insights and proactive decisions are essential. By aligning computation with dynamic data characteristics, adaptive graph algorithms deliver scalable and resilient solutions that significantly enhance graph-based analytics in real-world scenarios. Figure 4 illustrates a random graph growth model for AlphaGraph_1. Each cell contains two-digit values representing edged weights or distances between nodes. The network evolves iteratively, adding new nodes and modifying existing links based on proximity or similarity. This adaptive structure reflects changing data patterns, enabling rapid shortest path estimation and uncovering complex relationships. Such dynamic modeling improves both scalability and precision in big data graph analytics.

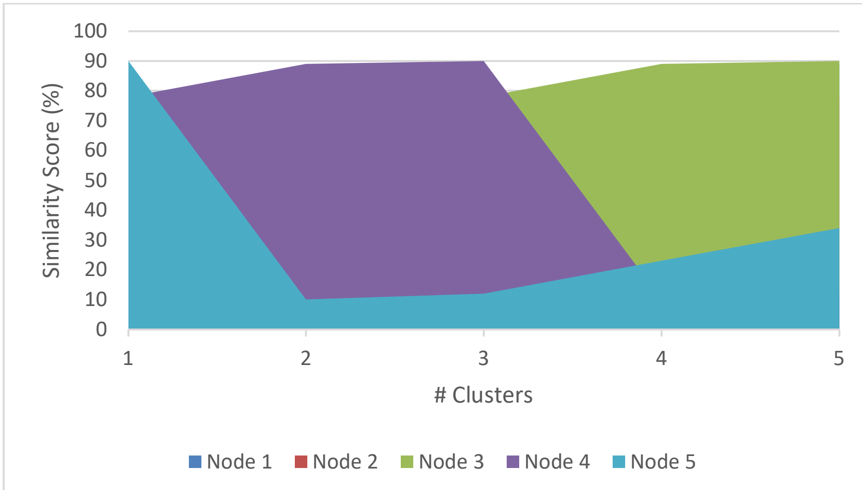

FIGURE 4. Random Graph Growth Model

Figure 5 presents the *AlphaGraph_2* dataset modeled using a preferential attachment approach. Each cell contains two-digit numeric values that represent node properties or edge weights. The preferential attachment mechanism favors connections to already well-connected nodes, resulting in a scale-free network with a power-law distribution of node degrees.

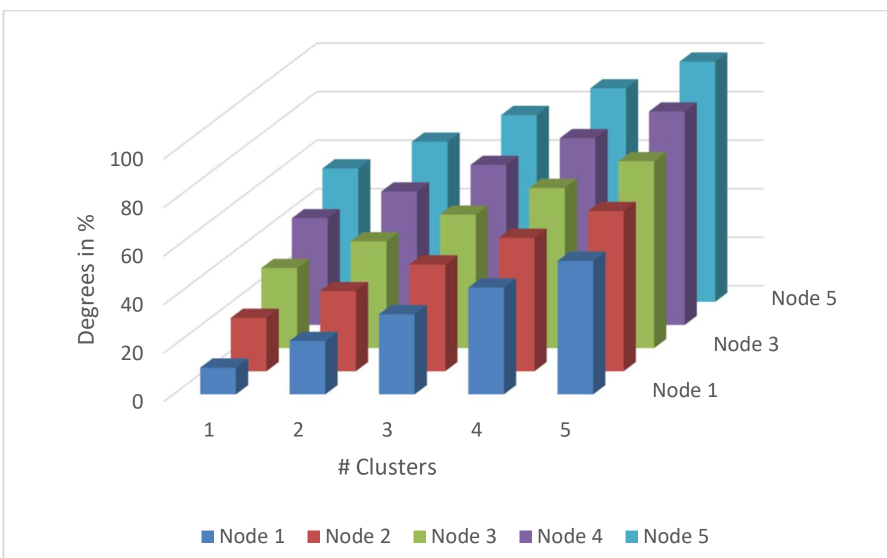

FIGURE 5. Preferential Attachment Model

As the graph evolves, new nodes preferentially link to hubs, creating a structure where a few nodes dominate connectivity while others maintain fewer links. This dynamic growth pattern supports efficient shortest path computations, as highly connected nodes act as traversal shortcuts. Such properties make the model particularly effective for big data contexts, where rapid analysis of large and complex networks is required.

Figure 6 illustrates the AlphaGraph_3 dataset generated through a community-based growth model. Each cell contains two-digit numeric values representing node characteristics or edge weights. The algorithm fosters both intra-community cohesion and inter-community connectivity, resulting in the formation of tightly clustered sub-networks within the graph. This structural organization enhances shortest path computations by leveraging localized clusters while maintaining global connectivity across the network. Such clustering provides valuable insights into data organization and linkages, enabling more efficient analysis of complex systems. The community-based approach supports scalable big data applications, offering improved decision-making in domains where dense and overlapping network relationships play a critical role.

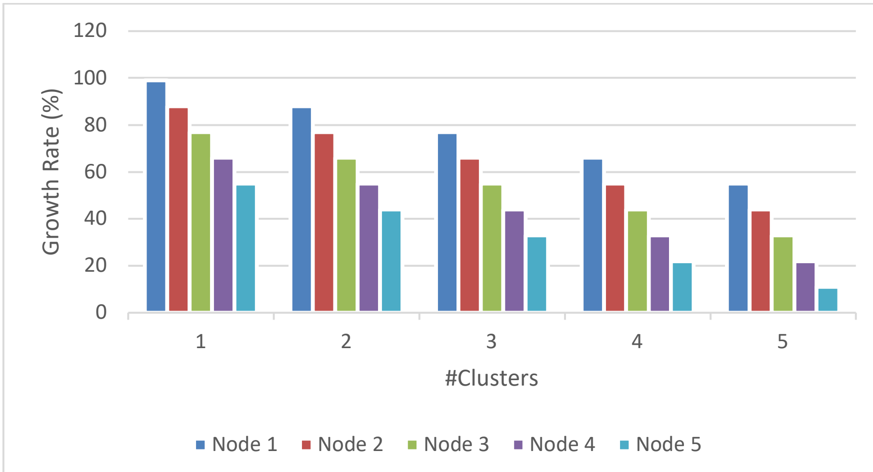

FIGURE 6. Community-Based Growth Model

CONCLUSION

The findings establish graph growth algorithms as a promising solution for scalable and efficient shortest path computation in big data environments. Their ability to dynamically expand networks, adapt to real-time changes, and distribute workloads across parallel systems enhances both accuracy and speed in large-scale analysis. Experimental demonstrations across synthetic graph models confirm their effectiveness in reducing computational overhead, improving pathfinding precision, and enabling fault-tolerant processing. However, implementation complexity, high resource consumption, and challenges in maintaining data privacy remain limiting factors. These shortcomings underline the need for specialized infrastructure and advanced optimization strategies to fully leverage their potential. Future research should focus on integrating machine learning–driven heuristics for adaptive path recalculation, refining parallelization techniques for distributed environments, and simplifying deployment through containerized and cloud-based platforms. By addressing these gaps, graph growth algorithms can evolve into a robust framework for shortest route analysis across diverse domains, ensuring reliable decision-making in the face of rapidly expanding and evolving data.

REFERENCES

[1]. D. Atrushi, and S. R. Zeebaree, 2024, "Distributed graph processing in cloud computing: a review of large-scale graph analytics," *Indonesian Journal of Computer Science*, 13(2), pp. 1542-1562.

[2]. R. Anzoom, R. Nagi, and C. Vogiatzis, 2024, "Uncovering illicit supply networks and their interfaces to licit counterparts through graph-theoretic algorithms," *IISE Transactions*, 56(3), pp. 224-240.

[3]. P. Gepner, B. Kocot, M. Paprzycki, M. Ganzha, L. Moroz, and T. Olas, 2024, "Performance evaluation of parallel graphs algorithms utilizing graphcore IPU," *Electronics*, 13(11), pp. 1-15.

[4]. Z. Hu, and R. Evans, 2024, "A fast score-based search algorithm for maximal ancestral graphs using entropy," arXiv preprint arXiv:2402.04777, pp. 1-55.

[5]. M. S. Rudwan, and J. V. Fonou-Dombeu, 2024, "A novel algorithm for multi-criteria ontology merging through iterative update of rdf graph," *Big Data and Cognitive Computing*, 8(3), pp. 1-20.

[6]. R. Khalef, and I. H. El-adaway, 2024, "Investigating causes of disputes resulting in litigation in airport

development projects in the United States using graph-based techniques," *Journal of Management in Engineering*, 40(3), pp. 1-21.
[7]. P. Vyas, C. M. Brannon, L. Formery, C. Lowe, and M. Prakash, 2024, "Algorithmic construction of topologically complex biomineral lattices via cellular syncytia," *bioRxiv*, pp. 1-46.
[8]. W. Mei, X. Wei, B. Ning, Z. Chen, and R. Zhang, 2024, "Movable-Antenna position optimization: A graph-based approach," *IEEE Wireless Communications Letters*, 13(7), pp. 1853-1857.
[9]. W. Cong, J. Kang, H. Tong, and M. Mahdavi, 2024, "On the generalization capability of temporal graph learning algorithms: Theoretical insights and a simpler method," *arXiv preprint arXiv:2402.16387*, pp. 1-57.
[10]. G. Wu, and A. Sullivan, 2024, "AlloyASG: Alloy predicate code representation as a compact structurally balanced graph," arXiv preprint arXiv:2403.00170, pp. 1-12.
[11]. B. Mitra, M. Deshmukh, and A. Kumar, 2024, "A novel approach for single-shot target recognition by matching graph of feature pixels," *Procedia Computer Science*, 235, pp. 530-538.
[12]. G. Jayabalasamy, C. Pujol, and K. Latha Bhaskaran, 2024, "Application of graph theory for blockchain technologies," *Mathematics*, 12(8), Article. 1133.
[13]. A. Krishnaa, 2024, "Some applications of antimagic graphs in cryptology," *The Mathematics Education*, 20(1), pp. 1-10.
[14]. L. Pandey, S. Chaulagain, M. Khanal, P. Bhattrai, S. Wagle, and S. Sahani, 2024, "Solving the general planar graph coloring problem using Grover's algorithm," *International Journal of Innovative Science and Research Technology*, 8(12), pp. 1902-1910.
[15]. J. He, J. Kang, F. Nargesian, H. Wang, A. Zhang, and D. Zhou, 2024, "TrustLOG: The second workshop on trustworthy learning on graphs," *Companion Proceedings of the ACM on Web Conference*, vol. 2024, pp. 1785 – 1788.
[16]. C. Dong, Y. Feng, and W. Shang, 2024, "A new method of dynamic network security analysis based on dynamic uncertain causality graph," *Journal of Cloud Computing*, 13(1), Article. 24.
[17]. Y. Cheng, C. Huang, H. Jiang, X. Xu, and F. Wang, 2024, "An efficient SSSP algorithm on time-evolving graphs with prediction of computation results," *Journal of Parallel and Distributed Computing*, 186, Article. 104830.
[18]. X. Liu, Z. Yang, and J. Cheng, 2024, "Music recommendation algorithms based on knowledge graph and multi-task feature learning," *Scientific Reports*, 14(1), Article. 2055.
[19]. M. B. Koca, and F. E. Sevilgen, 2024, "An efficient index-based algorithm for exact subgraph isomorphism on bipartite graphs," *Scientific Research Communications*, 4(1), pp. 1-17.
[20]. Z. Huang, D. Tang, R. Zhao, and W. Rao, 2024, "A scientific paper recommendation method using the time decay heterogeneous graph," *Scientometrics*, pp. 1-25.