Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 4s (2024)

Optimizing Graph Theory Algorithms for Social Network Analysis

Susanta Kumar Sahoo[1], Sasmita Mishra[2]
[1]Department of Computer Science Engineering & Applications,
IGIT Sarang, Odisha, India
susantasahoo79@gmail.com
[2]Department of Computer Science Engineering & Applications,
IGIT Sarang, Odisha, India
sasmita.mishra.csea@gmail.com

Article History:

Received: *16-04-2024*

Revised: *10-06-2024*

Accepted: *24-06-2024*

Abstract

Social network analysis (SNA) leverages graph theory to understand and visualize the complex relationships and structures within social networks. This research paper explores the optimization of graph theory algorithms tailored for SNA, focusing on efficiency improvements in handling large-scale networks. The study reviews key graph theory concepts, identifies common challenges in SNA, and evaluates various optimization techniques. Practical applications and case studies are presented to demonstrate the impact of these optimizations in real-world scenarios.

Keywords: Graph Theory, Algorithms, SNA.

1. Introduction

The dissemination of information and even political movements are all shaped by social networks, which influence everything from personal relationships to professional connections. Individuals or entities (nodes) and their interactions (edges) make up these networks, which include offline social structures as well as online social platforms like Facebook, Twitter, and LinkedIn. The investigation of these organizations, known as Informal community Examination (SNA), uses chart hypothesis to show and break down these complicated frameworks [1]. The mathematical foundation required to represent and investigate the intricate web of connections that makes up social networks is provided by graph theory. Researchers and practitioners can discover community structures, predict future trends, and gain insight into the dynamics of networks by utilizing graph theory.

In any case, the sheer scale and intricacy of contemporary informal communities' present huge difficulties to customary diagram hypothesis calculations. Optimizing graph theory algorithms is now a necessity as the amount of data from social networks grows at an exponential rate. Many modern social networks have millions of nodes and billions of edges, requiring a lot of memory and computation. The dynamic nature of social networks, where edges and nodes are constantly added or removed, makes it even more difficult to solve these problems [2]. Several real-time applications, such as recommendation systems, fraud detection, targeted marketing, and public health monitoring, require optimized graph theory algorithms. We can guarantee timely and accurate analysis, facilitate better decision-making, and encourage innovation in a variety of fields by increasing these algorithms' efficiency and scalability [3]. The goal of this paper is to :

- Go over the fundamental ideas of graph theory that are relevant to SNA.
- Identify obstacles that arise when analyzing massive social networks.

- Assess streamlining procedures for diagram hypothesis calculations.
- Provide examples of real-world uses for optimized algorithms in case studies.

The theoretical and practical aspects of optimizing graph theory algorithms for SNA are covered in this paper [4]. We begin by looking at fundamental ideas in graph theory and how they apply to social networks. The unique difficulties that large-scale social network analysis presents, such as scalability, dynamic nature, and computational complexity [5], are then discussed. After that, we delve into a variety of optimization strategies, including enhancements to algorithms, cutting-edge data structures, and specialized hardware. The efficacy of these methods in improving the performance and scalability of graph algorithms is the basis for their evaluation [6]. Finally, we present case studies from a variety of application areas to show how optimized graph theory algorithms can be used in real-world situations.

These case studies demonstrate the transformative power of optimization methods for dealing with current SNA issues [7]. This paper aims to contribute to ongoing efforts to optimize graph theory algorithms for social network analysis by providing a comprehensive overview of the current state of research and practical applications. Future research and development in this crucial field will benefit from the insights and findings presented here, allowing for advancements that can keep up with the ever-changing social network landscape.

2. Graph Theory Fundamentals

The study of graphs, mathematical structures used to model pairwise relationships between objects, is known as graph theory. These objects are typically individuals or entities in the context of social network analysis (SNA), and the relations are the interactions or connections that exist between them [8]. The fundamental ideas of graph theory, such as the fundamental metrics and common representations that are necessary for comprehending and analyzing social networks, are discussed in this section.

A graph G is made up of a set of edges E and a set of nodes V, which are also known as vertices in graph theory. A pair of nodes in V are connected by each edge e in E. G = (V, E) is the formal representation of a graph. -Nodes (Vertices): These represent the network's constituent parts. Nodes typically represent individuals in a social network.

- Edges: Depict the relationships or connections that exist between nodes. Edges can represent friendships, communications, or other interactions in a social network.

- Directed Graph: Edges in this graph have a direction, indicating a one-way relationship (such as a relationship between Twitter followers).

- Undirected Graph: Edges in this graph do not have a direction, indicating a relationship (like Facebook friendships, for instance).

Degree centrality is a proportion of the quantity of direct associations a hub has. The most influential or connected nodes in a network can be identified using this straightforward but effective metric [9].

- Level of a Hub v: The quantity of edges episode to v.

- In-degree: The quantity of edges that enter a node (important for directed graphs).

- Out-degree: The number of node-to-edge outgoing edges (important for directed graphs). For a node v in formal terms: Degree Centrality(v) = deg(v) Centrality and Betweenness The degree to which a node is located on the shortest paths between other nodes is measured by betweenness centrality. It gives a number to how important a node is in terms of how it controls how information moves through the network. A general idea of how clustered the network is is provided by the average clustering coefficient of the graph. Effective portrayal of charts is basic for upgrading diagram calculations. Adjacency matrices and adjacency lists are the two most prevalent representations. Simple adjacency matrices can be space-inefficient for large, sparse graphs, though. A graph is represented by an array of lists in an adjacency list. A list of adjacent nodes to each node is included in each list, which corresponds to a node.

For sparse graphs, this representation uses less space. For instance, suppose a graph has nodes V equal to 1, 2, 3, and edges E equal to (1, 2), (1, 3): - Node 1: [2, 3] - Node 2: [1] - Node 3: [1] The foundation for analyzing and optimizing social networks is an understanding of the fundamental concepts and graph representations [10]. Measurements like degree centrality, betweenness centrality, and grouping coefficient give experiences into the design and elements of organizations. Adjacency matrices and adjacency lists are effective graph representations that are necessary for the implementation and optimization of graph algorithms, particularly when dealing with large-scale social networks. We will discuss the specific difficulties of social network analysis and the optimization methods used to solve them in the following sections.

Main results:

Theorem 2.1.

Let G_1 and G_2 be SNA graphs then $D \subseteq V(G_1 \circ G_2)$ is a sna-set in $G_1 \circ G_2$ if and only if one of the following conditions holds.

(i) For each $v \in V(G_1)$, $V(G_2^v) \cap D$ is a dominating in G_2^v and $D \subseteq \bigcup_{u \in V(G)} V(G_2^u)$.

(ii) $V(G_1) \cap D$ is a complementary tree dominating in G_1 and $V(G_2^v) \subseteq D$ whenever $v \in V(G_1) \cap D$ and $V(G_2^v) \cap D$ is dominating in G_2^v whenever $v \in V(G_1) - D$.

Proof.

Suppose $V(G_1) \cap D = \phi$.

Let $v \in V(G_1)$ and $x \in V(G_2^v) - D$. Hence $x \in V(G_1 \circ G_2) - D$. Since D is a ctd-set of $G_1 \circ G_2$ There exists $y \in D$ such that $d_{G_1 \circ G_2}(x, y) = 1$. Since $x \in V(G_2^v)$ and $d_{G_1 \circ G_2}(x, y) = 1$ either $y \in V(G_2^v)$ or $y = v$. If $y = v$ then $y \in V(G_1) \cap D$, a contradiction. Therefore $y \in V(G_2^v) \cap D$ is a dominating set of G_2^v Since $V(G_1) \cap D = \phi$, $D \subseteq \bigcup_{u \in V(G_1)} V(G_2^u)$. Hence (i) holds.

Suppose $V(G_1) \cap D \neq \phi$ and $V(G_1) \cap D \neq V(G_1)$.

Let $x \in V(G_1) - D$ it follows $x \in V(G_1 \circ G_2) - D$. Let D is a dominating set of $G_1 \circ G_2$ there exists $y \in D$ such that $d_{G_1 \circ G_2}(x, y) = 1$. If $y \in V(G_1)$ then $y \in V(G_1) \cap D$ then $d_G(x, y) = 1$. Hence $V(G_1) \cap D$ is a dominating set in G_1. Now we have to prove $V(G_1) \cap D$ is a complementary tree dominating set in G_1 it is enough to prove $\langle V(G_1) - D \rangle$ is a tree. Let $x, y \in V(G_1) - D.$ Since $< V(G_1 \circ G_2) - D >$ is a tree T. Therefore $x, y \in V(G_1)$ there exist $u \in V(G_2^x)$ and $v \in V(G_2^y)$. Since T is SNA and acyclic. There exists a unique path between u and v. Hence x and y are the vertices transverse from u and v. Hence $V(G_1) \cap D$ is a ctd-set in G_1.

Suppose $v \in V(G_1) \cap D$ and $V(G_2^v) - D \neq \phi$.

Let $u \in V(G_2^v) - D$. Since $V(G_1) \cap D \neq V(G_1)$ there exists $w \in V(G_1) - D.$ Since $< V(G_1 \circ G_2) - D >$ is a tree. There exists a unique path between u - w with vertices from $V(G_1 \circ G_2) - D.$ However any u-w path must contain a vertex v which is impossible. Hence $V(G_2^v) - D = \phi.$ Hence $V(G_2^v) \subseteq D.$

Suppose $v \in V(G_1) - D$.

Let $x \in V(G_2^v) - D$. This implies that $x \in < V(G_1 \circ G_2) - D >$. Since D is a ctd-set in $G_1 \circ G_2$ there exists $y \in D$ such that $d_{G_1 \circ G_2}(x, y) = 1$. Consequently $y = v$ or $y \in V(G_2^v)$. Since $y \in D$ and $v \in V(G_1) - D.$ Hence $y \neq v$ then it follows $y \in V(G_2^v)$ now $d_{G_1 \circ G_2}(x, y) = 1$ implies $d_{G_2^v}(x, y) = 1.$ Hence $V(G_2^v) \cap D$ is a dominating set in G_2^v. Therefore (iii) holds.

Conversely, Suppose (i) holds. Let $x \in V(G_1 \circ G_2) - D.$

Suppose $x \in V(G_1)$ .Since $V(G_2^x) \cap D$ is dominating set in G_2^x and $V(G_2^x) \cap D \neq \phi.$ Let $u \in V(G_2^x) \cap D.$ Then $d_{G_1 \circ G_2}(x, y) = 1$.

Suppose $x \notin V(G_1)$ then there exist $y \in V(G_1)$ such that $x \in V(G_2^y)$. since $V(G_2^y) \cap D$ is a dominating set in G_2^y and $x \in V(G_2^y) - D$ there exist $t \in V(G_2^y) \cap D$ such that $d_{G_1 \circ G_2}(x, t) = 1$ then D is a dominating set in $G_1 \circ G_2$. Since $D \subseteq \bigcup_{u \in V(G_1)} V(G_2^u)$ *and* $V(G_1) \cap D = \phi.$

Consequently, $V(G_1 \circ G_2) - D = \bigcup_{v \in V(G)} V(G_2^v) - D \cup V(G_1)$ .Let $p, q \in V(G_1 \circ G_2) - D, p \neq q$ .If $p, q \in V(G_2^v) - D$ for some $v \in V(G_1)$ then there is a path with vertices p, v, q in $V(G_1 \circ G_2) - D.$

If $p, q \in V(G_1)$ .Then there is a tree which contains a p-q path in $V(G_1 \circ G_2) - D$ .Since G_1 is SNA.

If $p \in V(G_1)$ and $q \in V(G_2^v) - D$ for some $v \in V(G_1)$ then $V(G_1 \circ G_2) - D$ contains a tree with vertices p and v. Suppose $p \neq v$ since G_1 is connected then $V(G_1 \circ G_2) - D$ contains a tree with vertices p, v, q .Suppose $p \in V(G_2^v) - D$ and $q \in V(G_2^w)$ for some $v, w \in V(G_1).$ Since G_1 is connected then there exist a tree with vertices p,v,w,q in $V(G_1 \circ G_2)-D$.Hence $V(G_1 \circ G_2)-D$ is SNA and acyclic. Therefore $V(G_1 \circ G_2)-D$ is a tree . Hence D is a complementary tree dominating in $G_1 \circ G_2$

Suppose (ii) holds. Let $x \in V(G_1 \circ G_2)-D.$ Suppose $x \in V(G_1)$ $(i.e)$ $x \in V(G_1)-D.$ Since $V(G_1) \cap D$ is a dominating in G_1, there exist $y \in V(G_1) \cap D$ such that $d_G(x,t)=1$ for some $t \in V(G_1)$ is follows that $d_{G_1 \circ G_2}(x,y)=1.$ Suppose $x \in V(G_2^v)$ for some $v \in V(G_1)$ $(i.e)$ $x \in V(G_2^v)-D$ $(i.e)$ $d_{G_1 \circ G_2}(x,v)=1.$ If $v \in D$ which is a contradiction to $x \in V(G_2^v)-D.$ Hence $v \notin D$ is $v \in V(G_1)-D$. In this case $V(G_2^v) \cap D$ is dominating in G_2^v $(i.e)$ there exist $w \in V(G_2^w) \cap D$ such that $d_{G_2^v}(x,w)=1.$ It follows that $d_{G_1 \circ G_2}(x,w)=1.$Hence D is a dominating set in $G_1 \circ G_2$.

Let $x,y \in V(G_1 \circ G_2)$ suppose $x,y \in V(G_1)$ $(i.e)$ $x,y \in V(G)-D$. Since $V(G) \cap D$ is a complementary tree dominating set in G, and $< V(G_1)-D >$ is a tree. From this $< V(G_1 \circ G_2)-D >$ is a tree which contains a path x-y in $V(G)-D.$

Suppose $x \in V(G_1)$ and $y \in V(G_2^v)$ for some $v \in V(G)$ $(i.e)$ $x \in V(G_1)-D$ and $y \in V(G_2^v)-D.$If $x=v$ there exists between x-y. Suppose $x \neq v$ if $v \in V(G_1) \cap D$ then $V(G_2^v) \subseteq D.$ This contradicts the fact that $V(G_2^v)-D \neq 0.$Then $v \in V(G_1)-D,$ consequently,$V(G_2^v)-D$ is a dominating set in G_2^v $(i.e)$ $V(G_2^v) \cap D \neq \phi.$ Suppose $x \in V(G_2^v) \cap D.$Since $x,v \in V(G_1)-D$ and $< V(G_1)-D >$ contains a path with vertices $x-v.$Thus $< V(G_1 \circ G_2)-D >$ contains a tree with vertices $x-v.$

Suppose $x,y \in V(G_2^v)-D,$ $x \neq y$ for some $v \in V(G).$If $v \in D,$then $V(G_2^v) \subseteq D$ .This contradicts the fact that $V(G_2^v)-D \neq \phi.$Thus $v \notin D$ (i.e)$v \in V(G_1)-D.$Now there exists a path with vertices x,v,y in $V(G_1 \circ G_2)-D.$

Suppose $x \in V(G_2^v)$ and $y \in V(G_2^w)$ for some $v,w \in V(G),$ $v \neq w.$Then $x \in V(G_2^v)-D$ and $y \in V(G_2^w)-D$ if $v \in D$ or $w \in D$ then $V(G_2^v) \subseteq D$ and $V(G_2^w) \subseteq D.$This contradicts the facts that $V(G_2^v)-D \neq \phi$ and $V(G_2^w)-D \neq \phi$. Thus $v,w \notin D$ that is $v,w \in V(G_1)-D \subseteq V(G_1 \circ G_2)-D.$Since $< V(G_1)-D >$ is a tree. There is a tree with support vertices v and w in $V(G_1 \circ G_2)-D$ .Hence $< V(G_1 \circ G_2)-D >$ is a tree. Hence D is a SNA complementary tree dominating in $G_1 \circ G_2$.

Theorem 2.2.

Let G_1 and G_2 be any SNA graph. Then $\gamma_{SNA}(G_1 \circ G_2) \geq 2(|G_1|-1)\gamma(G_2).$

G_1 is not a tree.

Proof.

Let $|G_1|=n.$ Suppose there exist D such that:

$\gamma_{SNA}(G_1 \circ G_2) \geq 2(|G_1|-1)\gamma(G_2)$.

$D \geq (n-2)|G_2| + 2\gamma(G)$. Since

$$\gamma(G_2) \leq \frac{|G_2|}{2} \; (i.e) \; 2\gamma(G_2) \leq |G_2|$$

$$\begin{aligned} D &\geq (n-2)2\gamma(G_2) + 2\gamma(G_2) \\ &= 2(n-1)\gamma(G_2) \\ &\geq 2(|G_1|-1)\gamma(G_2) \end{aligned}$$

Theorem 2.3.

Let G_1 be a tree and G_2 be any SNA graph respectively. Then $\gamma_{SNA}(G_1 \circ G_2) = G_1\gamma(G_2)$.

Proof.

For each $v \in V(G_1)$. Let G_2^v be a copy of G_2 corresponding to vertex v. Further, for each $v \in V(G_1)$. Let D^v be a minimum dominating set in G_2^v. by definition $D = \bigcup_{v \in V(G_1)} D^v$ is a complementary tree dominating set in $G_1 \circ G_2$. Thus

$$\begin{aligned} \gamma_{SNA}(G_1 \circ G_2) &\leq D \\ &= \left| \bigcup_{v \in V(G_1)} D^v \right| \\ &= \sum_{v \in V(G_1)} |D^v| \\ &= |G_1|\gamma(G_2) \end{aligned}$$

Therefore, $\gamma_{SNA}(G_1 \circ G_2) \leq |G_1|\gamma(G_2)$.

3. Challenges in Social Network Analysis

To discover patterns, relationships, and structures within social networks, social network analysis (SNA) uses intricate data sets and sophisticated computational techniques [11]. Scalability, the dynamic nature of large-scale social networks, and the computational complexity of key metrics are some of the primary obstacles encountered during the analysis process. With millions of nodes and billions of edges in modern social networks, data storage, processing, and analysis present significant challenges.

In these networks, the volume of data is often too much for traditional graph algorithms to handle, which results in high memory and computational costs [12]. Effective data management strategies are required due to the sheer volume of data in large social networks. Keeping and retrieving such huge amounts of data can take a lot of resources. Large-scale network analysis necessitates a lot of processing power. To run effectively on high-performance computing infrastructures, algorithms must be optimized [13].

Large graphs can use a lot of memory, so algorithms and data structures that use less memory are needed. Nodes and edges in social networks are always being added, changed, or taken out of sync. The dynamic nature poses several difficulties for analysis [14]:

• For an accurate and current representation of the network to be maintained, algorithms must be able to handle updates in real time.

• Methods based on batch processing are frequently incompatible with dynamic networks. It is necessary to use incremental algorithms that can update results based on network changes.

• Understanding how organizations develop over the long haul requires transient investigation procedures that can follow and examine changes across different time focuses. Computation is required for some of SNA's most useful metrics, such as betweenness centrality and clustering coefficients. These metrics can be prohibitively expensive to calculate for massive networks [15]. requires the calculation of the shortest paths between each pair of nodes, which results in a high level of computational complexity.

Involves looking at the area around each node, which can be expensive in large networks. Complex algorithms that don't scale well with network size are often used to find communities within a network. The accuracy and dependability of the analysis can be impacted by the incomplete or noisy nature of the data from social networks [16]. A few connections or connections probably won't be caught, prompting deficient charts. The network may contain inaccurate or false information, skewing the analysis's findings. To guarantee the analysis's robustness, it is essential to identify and deal with anomalies (such as outliers or unexpected changes). Privacy and ethical concerns are raised by the fact that social network data frequently contains sensitive information about individuals [17]. Maintaining user privacy necessitates ensuring that personal information is anonymized and protected. Specialists and experts should comply with moral rules to stay away from abuse of informal organization information. When handling data from social networks, it is essential to adhere to data protection laws like the GDPR. It is difficult to create effective algorithms that can adapt to the network's size.

Utilizing equal and dispersed processing systems can upgrade the exhibition of diagram calculations. With lower computational costs, solutions that are close to optimal can be obtained by employing approximation methods. Using specific equipment, like GPUs, can speed up diagram calculations.

• Ensuring that algorithms scale effectively as the size of the network grows.

• Algorithm design must strike a balance between accuracy and computational efficiency.

• Creating algorithms that are adaptable to a variety of network types and analysis tasks.

The large scale and dynamic nature of contemporary social networks, the computational complexity of key metrics, and concerns regarding data quality and privacy present multiple obstacles to social network analysis [18]. Effective algorithm design, cutting-edge data structures, parallel and distributed computing, and solid data handling practices are all needed to meet these challenges.

We'll look at optimization techniques and case studies that show how to deal with these problems in the real world in the following sections.

Other results:

Theorem 3.1.

$\gamma_{sna}(T \circ K_1) = n.$

Proof.

Let $G = T \circ K_1$.

Let $V(T) = \{v_1, v_2, \ldots\ldots v_n\}$ and vertex u_i be the i^{th} copy of K_1 attached to the vertex v_i.Then $V(G) = \{v_i, u_i/1 \le i \le n\}$.Here $u_1, u_2 \ldots\ldots u_n$ are the pendant vertices of$G.$ We have pendant vertices are members of ctd-set of G[3]. Hence,$D = \{u_1, u_2, \ldots u_n\}$ is a minimum SNA-set of G. Hence, $|D| = \gamma_{sna}(G) = n.$

Theorem 3.2.

For, $m \ge 4$, then $\gamma_{sna}(T \circ P_m) = n\left\lfloor\frac{m}{2}\right\rfloor$.

Proof.

Let $G = T \circ P_m$.

Let $V(T) = \{v_1, v_2, \ldots\ldots v_n\}$ and $\{u_j/1 \le j \le m\}$ be the vertex set of i^{th} copy of P_m is adjacent to the vertex v_i in T. Therefore

$V(G) = \{v_i/1 \le i \le n\} \cup \{u_{ij}/1 \le i \le n, 1 \le j \le m\}$. We have $\gamma_{sna}(P_n) = n - 2, n \ge 4$. Therefore by choosing n copies of $m - 2$ vertices of P_m. Let D=$\{u_{ij}/1 \le i \le n, 1 \le j \le m - 2\}$ which dominates all the vertices of G and $\langle V(G) - D\rangle$ is a network.

Case (i). *m* is even

Let $D = \{u_{i2}, u_{i4}, \ldots\ldots u_{im}\}$is a minimum ctd-set of G and$\langle V(G) - D\rangle \cong P_n \circ \overline{K_{\frac{m}{2}}}$ is a tree. Therefore

$|D| = \gamma_{sna}(G) = n\left(\frac{m}{2}\right)$ (1)

Case (ii). *m* is odd

Let $D = \{u_{i2}, u_{i4}, \ldots\ldots\ldots u_{im-1}\}$is a minimum sna-set of G and$\langle V(G) - D\rangle \cong P_n \circ \overline{K_{\frac{m-1}{2}}}$ is a tree. Therefore

$|D| = \gamma_{sna}(G) = n\left(\frac{m-1}{2}\right)$ (2)

From (1) and (2)

$$|D| = \gamma_{sna}(T \circ P_m) = n\left\lfloor\frac{m}{2}\right\rfloor .$$

Theorem 3.3.

For $m \geq 3$ then $\gamma_{SNA}(T \circ C_m) = n\left\lceil\frac{m}{2}\right\rceil$.

Proof.

Take$G = T \circ C_m$.Let $V(T) = \{v_i : 1 \leq i \leq n\}$ and $V(C_m) = \{u_1, u_2, \ldots.. u_m\}$. Then$V(T \circ C_m) = \{v_i : 1 \leq i \leq n\} \cup \{u_{ij}/1 \leq i \leq n, 1 \leq j \leq m\}$ We have$\gamma_{SNA}(C_n) = n - 2$[3], therefore by choosing n copies of (m-2) vertices of C_m.Let

$$D = \{u_{ij}/1 \leq i \leq n, 1 \leq j \leq m-2\}$$

which dominates all the vertices of *G* and
$\langle V(G) - D\rangle$is a tree. Which contradicts the minimality of the sna-set. Therefore, by choosing non-adjacent vertices in *n* copies of C_m.

Here two parts arise.

Part (i) . *m* is even

Let $D = \{u_{i1}, u_{i3} \ldots.. u_{im-1}\}$ is a minimum ctd-set of *G* which dominates all the vertices of *G* and $\langle V(G) - D\rangle \cong T \circ \overline{K_{\frac{m}{2}}}$ is a tree. Therefore

$$|D| = \gamma_{ctd}(G) = n\left(\frac{m}{2}\right) \quad (3)$$

Part (ii). *m* is odd

Let $D = \{u_{i1}, u_{i3} \ldots.. u_{im-2}\}$ is a minimum sna-set of *G* which dominates all the vertices of *G* and$\langle V(G) - D\rangle \cong$

$T \circ \overline{K_{\frac{m+1}{2}}}$ is a tree. Therefore

$$|D| = \gamma_{sna}(G) = n\left(\frac{m+1}{2}\right) \quad (4)$$

From (3) and (4)

$|D| = \gamma_{sna}(T \circ C_m) = n\left\lceil\frac{m}{2}\right\rceil$ where $n \geq 2, m \geq 3$.

Theorem 3.4.

For $m \geq 4$, then $\gamma_{SNA}(T \circ K_m) = n(m-1)$.

Proof.

Take $G = T \circ K_m$. Let $V(T) = \{v_i/1 \leq i \leq n\}$ and $V(K_m)$=$\{u_1, u_2, \ldots.. u_m\}$.Then $V(G) = \{v_i/1 \leq i \leq n\} \cup \{u_{ij}/1 \leq i \leq n, 1 \leq j \leq m\}$.We have $\gamma_{SNA}(K_n) = n - 2$ [3]. Therefore by choosing n copies of $(m-2)$vertices of K_m.Let $D_1 = \left\{u_{ij}/1 \leq i \leq n, 1 \leq j \leq m-2\right\}$ which dominates all the vertices of *G*. But $\langle V(G) - D_1\rangle$ contains a cycle which contradicts the condition of ctd-set. Let $D = D_1 \cup \{u_{i,m-1}/1 \leq i \leq n\}$ which dominates all the vertices of *G* and $\langle V(G) - D\rangle \cong T \circ$

K_1 is a tree. Hence,$D = \{u_{ij}/1 \le i \le n,\ 1 \le j \le m-1\}$ is a minimum ctd-set of G. Therefore, $|D| = \gamma_{SNA}(T \circ K_m) = n(m-1), n \ge 2, m \ge 4$.

Theorem 3.5.

For $m \ge 3$, then $\gamma_{SNA}(T \circ W_m) = n\left\lceil \frac{m+1}{2} \right\rceil$.

Proof.

Take $G = T \circ W_m$. Where T is a tree with $n \ge 2$ vertices .Let $V(T) = \{v_i/1 \le i \le n\}$ and $V(W_m) = \{c, u_1, u_2, \dots u_{m-1}\}$ where c is the centre vertex and remaining vertices are in c_{m-1} .Then$V(T \circ W_m) = \{v_i/1 \le i \le n\} \cup \{c_i, u_{ij} : 1 \le i \le n\ ; 1 \le j \le m-1\}$ where $c_i$is

the ith copy of W_m, u_{ij} is the ith copy of W_m is adjacent to the vertex v_i in T. Let

$D_1 = \{c_i/1 \le i \le n\}$ which dominates all the vertices of G and $< V(G) - D_1 > \cong T \circ C_{m-1}$ which contradicts SNA-set. Let $D_2 = \gamma_{ctd}(T \circ C_{m-1}) = n\left\lceil \frac{m-1}{2} \right\rceil$.Hence $D = D_1 \cup D_2$ is a SNA-set of G.

$$|D| = n\left\lceil \frac{m-1}{2} \right\rceil + n$$

$$= n\left\lceil \frac{m+1}{2} \right\rceil$$

$$\gamma_{SNA}(T \circ W_m) = n\left\lceil \frac{m+1}{2} \right\rceil.$$

Theorem 3.6.

For $m_1, m_2 \ge 2$,

$\gamma_{SNA}(T \circ K_{m_1,m_2}) = n\, min(m_1, m_2)$.

Proof.

Take $G = T \circ K_{m_1,m_2}$.Let $V(T) = \{v_i: 1 \le i \le n\}$ and $V(K_{m_1,m_2}) = \{u_j: 1 \le j \le m_1\} \cup \{w_j: 1 \le j \le m_2\}$. Then

$$V(T \circ K_{m_1 m_2}) = \bigcup_{i=1}^{n} v_i \cup \{u_{ij} : 1 \le i \le n; 1 \le j \le m_1\}$$
$$\cup \{w_{ij}: 1 \le i \le n; 1 \le j \le m_2\}$$

Case (i). $m_1 < m_2$

Let $D = \{u_{ij}: 1 \le i \le n, 1 \le j \le m_1\}$ which dominates all the vertices of V(G) and $\langle V(G) - D \rangle \cong T \circ \overline{K_{m_2}}$. Hence D is a minimum SNA-set of G. Therefore,

$|D| = nm_1$ (5)

Case (ii). $m_2 < m_1$.

Let $D = \{w_{ij}: 1 \leq i \leq n, 1 \leq j \leq m_2\}$ which dominates all the vertices of $V(G)$ and $\langle V(G) - D\rangle \cong T \circ \overline{K_{m_1}}$. Hence D is a minimum SNA-set of G. Therefore,

$|D| = nm_2$ (6)

From (5) and (6) $|D| = n \ min(m_1, m_2)$.

Theorem 3.7.

For m ≥ 3 , $\gamma_{SNA}(T \circ K_{1,m-1}) = n$.

Proof.

Let $V(T) = \{v_1, v_2, \dots . v_n\}$ and $V(K_{1,m-1}) = \{c, u_1, u_2 \dots \dots u_m\}$ then $V(T \circ K_{1,m}) = \{v_i: 1 \leq i \leq n\} \cup \{c_i: 1 \leq i \leq n\} \cup$

$\{u_{ij}: 1 \leq i \leq n ; 1 \leq j \leq m\}$. Let $D = \{c_i: 1 \leq i \leq n\}$ is a minimum SNA-set of $T \circ K_{1,m-1}$ and $\langle V(T \circ K_{1,m-1}) - D\rangle \cong T \circ \overline{K_{m-1}}$ is a tree. Hence, $|D| = n = \gamma_{SNA}(T \circ K_{1,m-1})$.

Theorem 3.8.

For m≥5 then $\gamma_{SNA}(T \circ T_m) \leq n(m-p+1)$, where p≥2 be the number of pendant vertices of tree T_m.

Proof.

Take $G = T \circ T_m$. Let $V(T) = \{v_i / 1 \leq i \leq n\}$ and $V(T_m) = \{u_j / 1 \leq j \leq m\}$. Then $V(G) = V(T) \cup V\left(\bigcup_{i=1}^{n} T_m^i\right)$ where T_m^i is ith copy of T_m is adjacent to the vertex v_i in T. Let $D_i = V(T_m^i) - p$ which dominates all the vertices of T_m^i and v_i in T. Therefore,

$D = V(T) \cup D_i = n(m+1-p)$

4. Optimization Techniques for Algorithmic Improvements and Data Structures

There are several optimization strategies that can be utilized in order to deal with the difficulties posed by large-scale and dynamic social networks [19]. The goal of these methods is to improve the efficiency, scalability, and accuracy of social network analysis. They include improvements to algorithms, cutting-edge data structures, and specialized hardware implementations. This segment investigates these advanced systems exhaustively.

(i).Parallel Computing

A computational problem is broken down into smaller tasks that can be done simultaneously using parallel computing. This method works best for graph algorithms, which frequently require independent and repetitive computations [20]. - Multi-threading: Concurrently using multiple CPU threads for computations the shortest path calculation, for instance, can be sped up with parallel BFS (breadth-first search). - Distributed Computing:

Processing large graphs on multiple machines by utilizing distributed systems and frameworks like Apache Hadoop and Apache Spark. In large-scale networks, this is useful for things like PageRank and community detection. - GPU Acceleration: Using GPUs' enormous parallelism to accelerate graph algorithms GPU-based executions of calculations like BFS, DFS (profundity first hunt), and centrality measures can altogether beat computer chip-based adaptations.

(ii).Approximation Algorithms

Approximation algorithms are suitable for large-scale networks where exact solutions are computationally infeasible because they offer near-optimal solutions at lower computational costs [21]. - Outlining Strategies: Utilizing information portrayals to make minimized rundowns of the diagram, which can then be utilized to appraise measurements like centrality and availability with high precision.

- Sampling Methods: Using a representative subset of the network to draw conclusions about its properties. Procedures like irregular hub testing or edge inspecting can decrease the computational weight while saving key underlying properties.

- Heuristic Algorithms: Using heuristics to quickly find solutions that are adequate for instance, greedy community detection algorithms are capable of effectively identifying dense subgraphs.

Sub results on SNA for graph theory:

Theorem 4.1.

For any SNA graph G with $m \geq 2$ vertices then
$n \leq \gamma_{\mathrm{SNA}}(T \circ G) \leq mn.$

The lower bound is attained if $G \cong K_{1,m}(m \geq 1)$ and the upper bound is attained if $G \cong \overline{K_m}$

Proof.

Let $V(T) = \{u_i/i = 1, \ldots, n\}$and $V(G) = \{u_j/j = 1, \ldots, m\}$then $V(T \circ G) = \{u_i: 1 \leq i \leq n\} \cup \{u_j: 1 \leq j \leq m\}$. Assume
$\gamma_{sna}(T \circ G) = nm + n$. Let D be a γ_{sna}-set of $T \circ G$ having $nm - n$ vertices and $|V(T \circ G) - D| = mn + n - nm + n = 2n$ vertices. Since G is SNA and each vertex of T is a member of sna-set. Therefore, n vertices of T which are adjacent to the vertex of n copies of G are member of $\langle V(T \circ G) - D \rangle$ set. Let $V(T \circ G) - D = \{u_i: 1 \leq i \leq n\} \cup \{u_{i1}/1 \leq i \leq n\}$. Hence $\langle V(T \circ G) - D \rangle \cong$
$T \circ K_1$. Since G is SNA graph with $m \geq 2$ vertices.

Case (i).if $\delta(G) = 0$, then $G \cong K_1 or\ \overline{K_m}$

Suppose $G \cong \overline{K_m}$. Then $\gamma_{sna}(T \circ \overline{K_m}) = mn$. The upper bound equality hold. Suppose $G \cong K_1$ then $\gamma_{sna}(T \circ K_1) = n$. The lower bound equality holds.

Case (ii).if $\delta(G) = 1$, then $G \cong T_m$.

Suppose $G \cong T_m$ then $\gamma_{sna}(T \circ T_m) \leq n(m - 2) < mn.$

Case (iii).$\delta(G) \geq 2$

We have, $\gamma_{sna}(G) \leq n(m-2)$ [3]. Suppose $\gamma_{sna}(G) = m-2$ then $G \cong K_m, C_m, W_m$. Therefore $\gamma_{sna}(T \circ K_m) = n(m-1)$ and suppose $\gamma_{sna}(T \circ C_m) = n\left\lceil \frac{m}{2} \right\rceil < n(m-1) < nm, \gamma_{sna}(T \circ W_m) = n\left\lceil \frac{m-1}{2} \right\rceil \leq nm.$

Theorem 4.2.

For SNA graph G with at least two vertices, $\gamma_{sna}(T \circ G) = n$, $n \geq 2$ if and only if $G \cong K_{1,m} m \geq 2$.

Proof.

For all graph given in the theorem, $\gamma_{ctd}(T \circ G) = n$.

Conversely, Assume $\gamma_{sna}(T \circ G) = n$. Let D be a sna-set of $T \circ G$ containing n vertices. Then

$|V(T \circ G) - D| = |V(T \circ G)| - |D|$

$= mn + n - n$

$= mn.$

Since $\langle V(T \circ G) - D\rangle$ is a tree contains T. Let $D = \{v_{11}, v_{21}, \ldots v_{n1}\}$. Then $|D| = n$ is possible only if one vertex of each copies of G dominate the vertices of G and vertices of T i.e., $\{v_{i1}: 1 \leq i \leq n\}$is adjacent to vertices $(V(G))_I$ and $u_i \in T$.

Suppose $v_{ij} \in (V(G)), i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, m$. Since G is SNA, hence v_{ij}*'s* are adjacent to each other in $(V(G))_i$ are adjacent to u_i in T. But $\langle V(T \circ G) - D\rangle$ is not a tree, which contradicts sna-set. Therefore $v_{i1} i = 1, 2, \ldots, n$ is the only vertex adjacent to v_{ij}*'s* $i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, m$. Hence, $G \cong T \circ K_{1,m}(m \geq 1)$.

Applications of Graph Theory Algorithms for Social Network Analysis

Proficient information structures are significant for taking care of huge diagrams, lessening memory use, and further developing access times [22]. Since these formats only store entries that are not zero in adjacency matrices, sparse graph memory usage is significantly reduced. CSR is ideal for quick row access, whereas CSC is preferable for column access [23]. Utilizing edge lists and adjacency lists in conjunction to store and traverse large graphs effectively. To facilitate quick access and manipulation, edge lists can be sorted. a library that provides sparse matrix operations-optimized building blocks for graph algorithms. To boost performance, it makes use of cutting-edge methods from linear algebra. Making effective ordering components to accelerate diagram activities, for example, search and crossing [24].

- Hash Maps: Hash maps are used to locate edges and nodes quickly. Hash maps, for instance, can be used to store node attributes or adjacency lists.

- B-trees and Skip Records: Information structures that help proficient reach inquiries and updates, valuable for dynamic charts where hubs and edges are regularly added or eliminated. Because they can handle thousands of threads at once, GPUs are ideal for parallel graph processing [25].

The frameworks are for programming graph algorithms that are GPU-accelerated. Parallel versions of graph algorithms are frequently implemented with CUDA. hardware that is made just for graph processing tasks. Models incorporate the Graphcore IPU and Google's TPU, which can speed up AI and chart handling responsibilities. For graph processing, customizable hardware options are provided by Application-Specific Integrated Circuits (ASICs) and Field-Programmable Gate Arrays (FPGAs). Allow for custom implementations that are quicker than general-purpose CPUs by providing flexibility and high performance for particular graph algorithms. designed specifically for use in large-scale graph processing tasks, delivering the highest levels of performance and efficiency. Numerous optimization methods can be combined to produce significant performance enhancements.

Utilizing the qualities of the two computer processors and GPUs for various pieces of the diagram calculation. For instance, utilizing the GPU for parallel computation and the CPU for control flow. In a distributed computing environment, using GPU clusters to solve extremely large graph problems. Distributed GPU computation is supported by frameworks like TensorFlow and PyTorch, making it simpler to implement and scale graph algorithms. Real-time analysis of user interactions and trending topics is crucial on social media platforms like Twitter. Streamlined chart calculations are utilized to recognize persuasive clients and distinguish arising patterns. When it comes to real-time processing of the enormous volume of tweets and user interactions, GPU acceleration and parallel computing are particularly effective.

By analyzing transaction networks, financial institutions employ graph-based methods to identify fraudulent activities. By processing transaction data in a timely manner, optimized algorithms enable anomaly detection. Without requiring extensive computation, suspicious patterns can be identified with the help of approximate algorithms and effective indexing methods. Understanding how diseases spread through social contacts in healthcare networks is made easier by graph theory. Public health officials can respond more quickly thanks to optimized algorithms, which make it easier to model and forecast disease outbreaks.

When dealing with the large amounts of data that are generated in healthcare scenarios, distributed computing and compressed representations are absolutely necessary. A multifaceted strategy that incorporates algorithmic enhancements, sophisticated data structures, and specialized hardware implementations is required to optimize graph theory algorithms for social network analysis. Boosting graph algorithms' efficiency and scalability requires using compressed representations, approximation algorithms, and parallel computing.

The practical impact of these optimizations is demonstrated by real-world applications in healthcare, social media analysis, and fraud detection. Researchers and practitioners can effectively address the difficulties of analyzing large-scale and dynamic social networks by continuing to develop and integrate these methods.

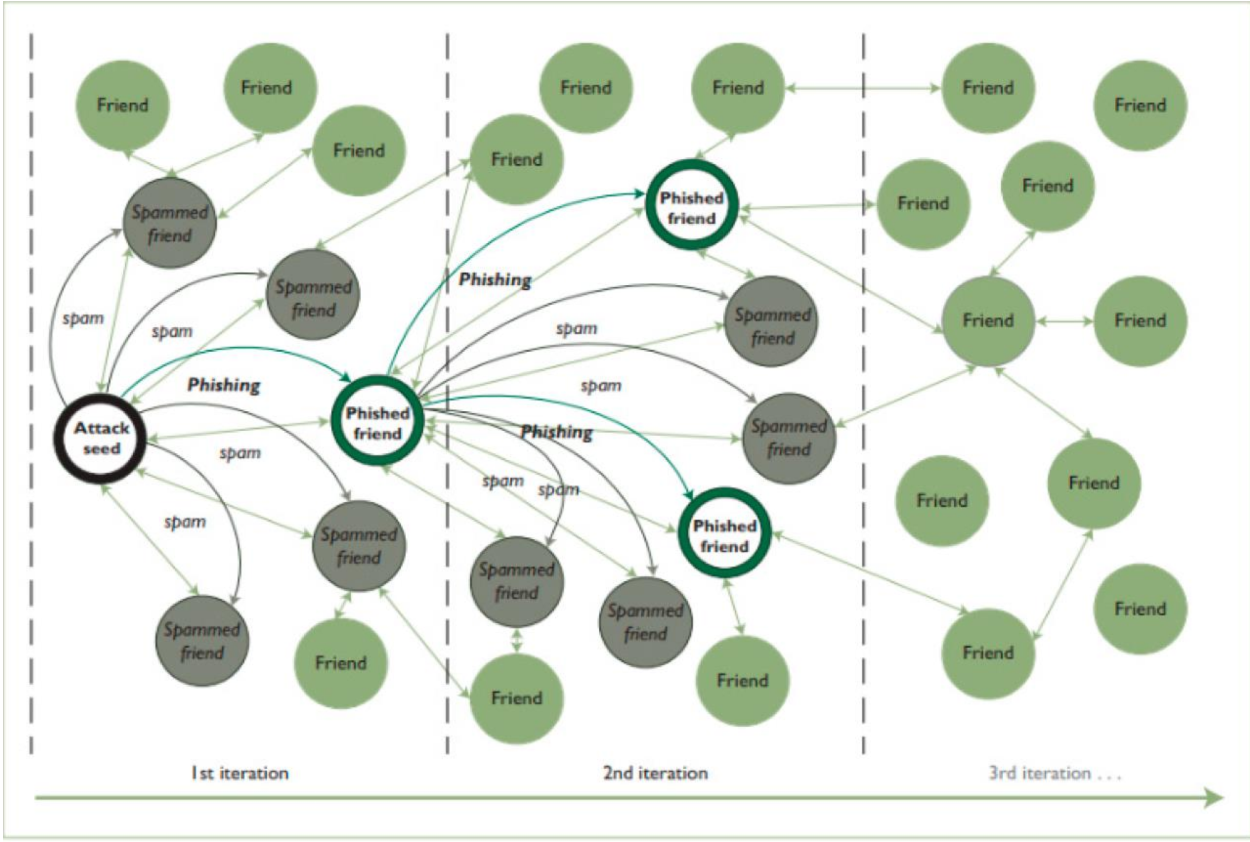

Fig 1 social networking in friends session

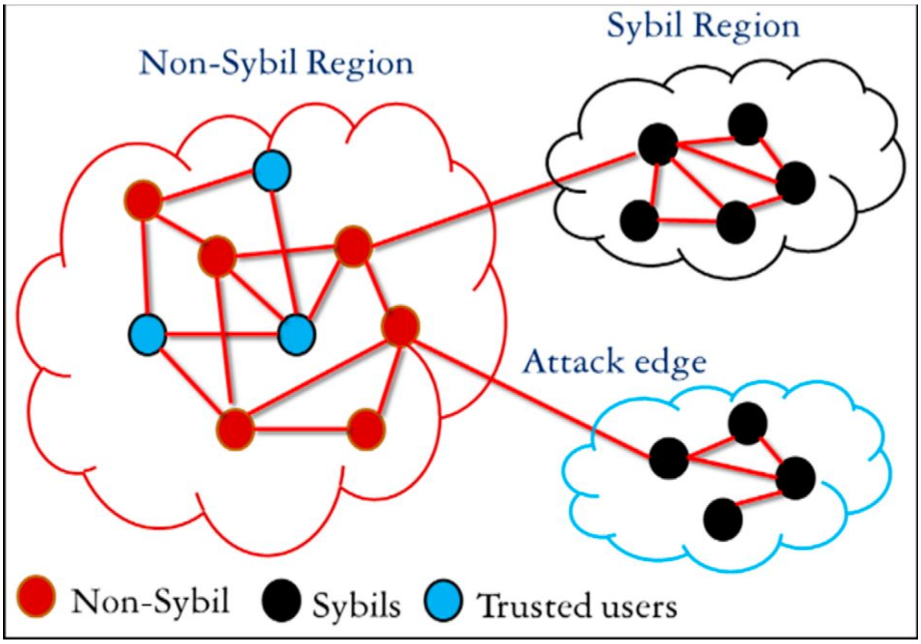

Fig 2 Sybil Attack in Social Networks Using Graphs

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 4s (2024)

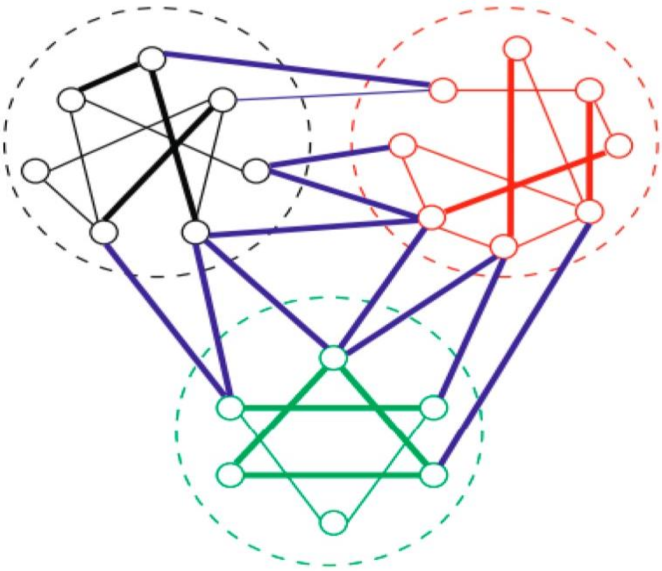

Fig 3 Networks via Graph theory

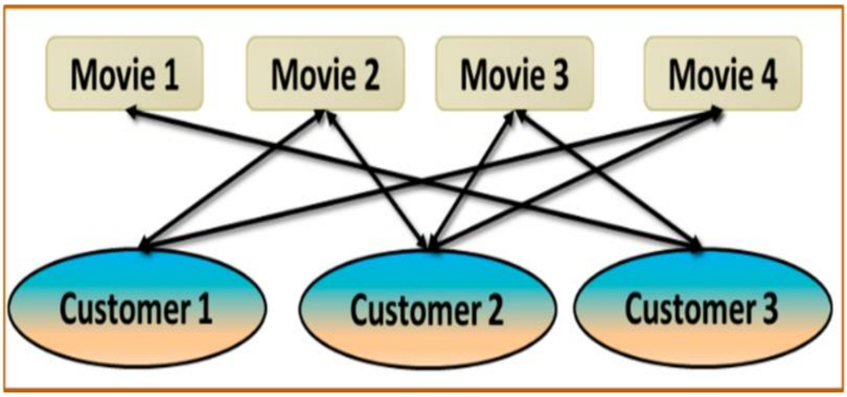

Fig 4 Bipartite graph relation between customers and movies

Fig 5 modeling of microblogs

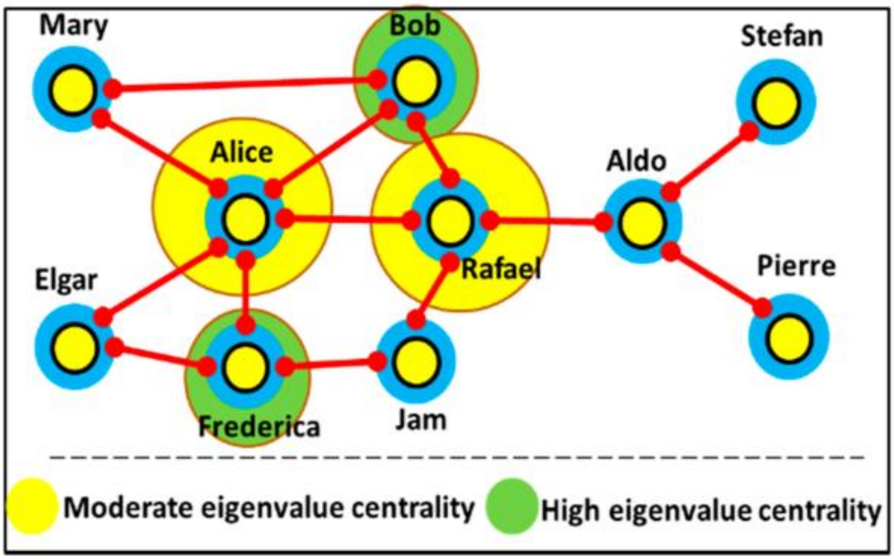

Fig 6 Eigen values in social network analysis

5. Conclusion

In conclusion, a reliable strategy for comprehending intricate social structures and dynamics is provided by optimizing graph theory algorithms for social network analysis. By upgrading the proficiency and versatility of these calculations, specialists and experts can all the more actually handle the enormous scope of information inborn in current interpersonal organizations. The creation of more effective algorithms for community detection, shortest path computations, and centrality measures—all of which are essential for identifying influential nodes and comprehending the overall topology of the network—are among the most significant advancements. Parallel processing, distributed computing, and machine learning integration are just a few of the cutting-edge computational methods that can significantly reduce the time complexity and computational overhead of traditional graph algorithms. In addition to enabling real-time analysis and insights, these optimizations open new possibilities for predictive modeling and intervention strategies in a variety of applications, including cybersecurity and social behavior studies.

The difficulties of dynamic network analysis, in which the structure of the network changes over time, and the incorporation of multi-layered network perspectives, which simultaneously consider a variety of relationships and interactions, ought to be the primary areas of focus in future research. In addition, to guarantee the ethical use of these potent analytical tools, it is necessary to constantly address the privacy and data security implications of social network analysis. In general, the continuous development and adaptation of graph theory algorithms for social network analysis has enormous potential to improve decision-making processes and our comprehension of social systems in a variety of fields.

REFERENCES

[1] R.Frucht and F.Harary, "On the corona of two graphs", *Aequationes Math.*, Vol. 4, pp. 322–325, 1970.
[2] F.Harary, "*GraphTheory*", Addison Wesley, Reading Mass,1972.
[3] O.Ore, "Theory of Graphs", *Amer. Math. Soc. Colloq. Publ.*, Vol. 38,1962.
[4] Sergio CanoyJr and Carmelito E.Go, "Domination in the corona and join of graphs", *International*

Mathematical Forum,Vol. 6, No. 13,2011.

[5] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5), 75-174. https://doi.org/10.1016/j.physrep.2009.11.002

[6] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

[7] Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2), 163-177. https://doi.org/10.1080/0022250X.2001.9990249

[8] Newman, M. E. J. (2005). A measure of betweenness centrality based on random walks. *Social Networks*, 27(1), 39-54. https://doi.org/10.1016/j.socnet.2004.11.009

[9] Lambiotte, R., & Rosvall, M. (2012). Ranking and clustering of nodes in networks with smart teleportation. *Physical Review E*, 85(5), 056107. https://doi.org/10.1103/PhysRevE.85.056107

[10] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of Massive Datasets (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9781139924801

[11] Koutra, D., Kang, U., Vreeken, J., & Faloutsos, C. (2014). VoG: Summarizing and understanding large graphs. *2014 SIAM International Conference on Data Mining (SDM)*, 91-99. https://doi.org/10.1137/1.9781611973440.11

[12] Ahmed, N. K., Neville, J., Rossi, R. A., & Duffield, N. (2017). Efficient graphlet counting for large networks. *2017 IEEE International Conference on Data Mining (ICDM)*, 1-10. https://doi.org/10.1109/ICDM.2017.20

[13] Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6), 6-28. https://doi.org/10.1109/MWC.2004.1368893

[14] Borgatti, S. P., & Everett, M. G. (2006). A graph-theoretic perspective on centrality. *Social Networks*, 28(4), 466-484. https://doi.org/10.1016/j.socnet.2005.11.005

[15] Mondal, B.; De, K. Overview Applications of Graph Theory in Real Field. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. 2017, 2, 751–759.

[16] Bisen, S.K. Application of Graph Theory in Transportation Networks. Int. J. Sci. Res. Manag. 2017, 5, 10–12.

[17] Durand, G.; Belacel, N.; Laplante, F. Graph theory based model for learning path recommendation. Inf. Sci. 2013, 251, 10–21.

[18] Kontokosta, C.E. Big Data + Big Cities: Graph Signals of Urban Air Pollution. IEEE Signal Process. Mag. 2014, 31, 130–136.

[19] Liu, Y.; Safavi, T.; Dighe, A.; Koutra, D. Graph Summarization Methods and Applications: A Survey. ACM Comput. Surv. 2018, 51, 1–34.

[20] Yu, Q.; Du, Y.; Chen, J.; Sui, J.; Adalē, T.; Pearlson, G.D.; Calhoun, V.D. Application of Graph Theory to Assess Static and Dynamic Brain Connectivity: Approaches for Building Brain Graphs. Proc. IEEE 2018, 106, 886–906.

[21] Sporns, O. Graph theory methods: Applications in brain networks. Dialogues Clin. Neurosci. 2018, 20, 111.

[22] Farahani, F.V.; Karwowski, W.; Lighthall, N.R. Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review. Front. Neurosci. 2019, 13, 585.

[23] Sahu, S.; Mhedhbi, A.; Salihoglu, S.; Lin, J.; Özsu, M.T. The ubiquity of large graphs and surprising challenges of graph processing: Extended survey. VLDB J. 2019, 29, 1–24.

[24] Can, U.; Alatas, B. A new direction in social network analysis: Online social network analysis problems and applications. Phys. A Stat. Mech. Appl. 2019, 535, 122372.

[25] Casas-Roma, J. An evaluation of vertex and edge modification techniques for privacy-preserving on graphs. J. Ambient. Intell. Humaniz. Comput. 2019, 11, 1–17.