Enhanced Community Detection Using Label Propagation Algorithm Integrated with Particle Swarm Optimization

D. DHANALAKSHMI[1], G. RAJENDRAN[2]
[1]Research Scholar, Periyar University, Salem -11 and Assistant Professor of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Tiruchengode, Namakkal, Tamilnadu, India
[2]Associate Professor and Head, Department of Computer Science, Govt. Arts and Science College, Modakkurichi, Erode, Tamilnadu, India

Abstract- Community detection in complex networks is pivotal for understanding the structural and functional properties of various systems ranging from social networks to biological systems. Traditional algorithms like the Label Propagation Algorithm (LPA) offer computational efficiency but often suffer from instability and accuracy issues. To address these challenges, this paper introduces the Enhanced Community Detection Using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO). By integrating the explorative capabilities of Particle Swarm Optimization (PSO) with LPA, the proposed method aims to enhance the stability and accuracy of community detection. Comparative analyses were conducted against established algorithms, including Girvan-Newman, K-Cliques, Chinese Whispers, Enhanced Community Detection Using Label Propagation Algorithm with ACO (ECDLPA-ACO), and Enhanced Community Detection Using Louvain Algorithm with ACO (ECDLA-ACO). Evaluations based on Modularity, Normalized Mutual Information (NMI), and Execution Time was performed on diverse datasets such as Reddit Hyperlink Network (RH-NW), Amazon Co-purchasing Network (ACP-NW), DBLP Collaboration Network (DBLP-NW), and Twitch Gamers Network (TG-NW). The results demonstrate that ECDLPA-PSO consistently outperforms its counterparts, achieving higher modularity and NMI scores while maintaining competitive execution times. This study underscores the potential of hybrid approaches in advancing community detection methodologies.

Indexed Terms- Community Detection, Particle Swarm Optimization (PSO), ECDLPA-PSO, Modularity, Normalized Mutual Information (NMI), Complex Networks, Optimization Algorithms, Social Network Analysis

I. INTRODUCTION

Community detection is a fundamental task in network science, essential for unraveling the hidden structures within complex networks such as social networks, biological systems, and communication networks. Identifying communities, or clusters of densely connected nodes, allows researchers and practitioners to better understand the organizational principles of the networks, leading to insights into their functionality and dynamics. Traditional community detection algorithms, including the Girvan-Newman and K-Cliques methods, have laid the foundation for this field, but they often struggle with scalability, accuracy, and computational efficiency when applied to large-scale networks [1] [2].

The Label Propagation Algorithm (LPA) is one of the most efficient community detection methods due to its linear time complexity. LPA iteratively assigns labels to nodes based on the majority label of their neighbors, leading to the formation of communities. However, LPA's deterministic nature and susceptibility to random tie-breaking decisions can result in instability and inconsistent community structures across different runs. Moreover, LPA's tendency to converge prematurely or get trapped in local optima limits its effectiveness in detecting more complex community structures [3]. To address these challenges, optimization techniques have been integrated into community detection algorithms, enhancing their performance by guiding the search process towards more optimal solutions. Among these techniques, Particle Swarm Optimization (PSO) has gained attention due to its ability to explore the search space efficiently by simulating the social behavior of particles [4]. PSO's adaptability and ability to avoid local optima make it an ideal candidate for improving the stability and accuracy of LPA.

While LPA provides a computationally efficient approach to community detection, its limitations in accuracy and stability pose significant challenges, particularly when applied to large and complex networks. Existing hybrid approaches, such as the Enhanced Community Detection Using Louvain Algorithm with ACO (ECDLA-ACO) and Enhanced Community Detection Using Label Propagation Algorithm with ACO (ECDLPA-ACO), have demonstrated the potential of combining traditional methods with optimization techniques. However, there remains a need for a more robust and scalable solution that can consistently produce accurate and stable community structures across diverse network types.

This research proposes the Enhanced Community Detection Using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO). The primary objectives of this study are:

To develop and implement the ECDLPA-PSO algorithm, which integrates PSO with LPA to enhance community detection performance.

To evaluate the performance of ECDLPA-PSO against established community detection algorithms, including Girvan-Newman, K-Cliques, Chinese Whispers, ICDLO, and ECDLA-ACO.

To assess the effectiveness of ECDLPA-PSO in terms of Modularity, Normalized Mutual Information (NMI), and Execution Time using datasets from diverse domains, including the Reddit Hyperlink Network (RH-NW), Amazon Co-purchasing Network (ACP-NW), DBLP Collaboration Network (DBLP-NW), and Twitch Gamers Network (TG-NW).

This paper contributes to the field of community detection in several key ways:

Algorithmic Innovation: The introduction of the ECDLPA-PSO algorithm, which combines the strengths of LPA and PSO to enhance the stability and accuracy of community detection.

Comparative Analysis: A comprehensive comparison of ECDLPA-PSO with five other established algorithms, providing insights into the relative strengths and weaknesses of each method.

Empirical Evaluation: Rigorous testing of the proposed algorithm on multiple real-world datasets, demonstrating its effectiveness across various network types and sizes.

By addressing the limitations of existing methods and providing a novel hybrid approach, this research aims to advance the state of the art in community detection, offering a more reliable and scalable solution for analyzing complex networks.

II. RELATED WORKS

Community detection algorithms are pivotal in understanding the structural organization of complex networks. This section provides a brief overview of five prominent community detection algorithms: Girvan-Newman, K-Cliques, Chinese Whispers, ECDLPA-ACO, and Enhanced Community Detection Using Louvain Algorithm with ACO (ECDLA-ACO), each accompanied by relevant references.

The Girvan-Newman algorithm, proposed by Girvan and Newman (2002) [5], is one of the earliest methods for detecting communities in networks. The algorithm operates by calculating the edge betweenness centrality for each edge in the network, which measures the number of shortest paths passing through the edge. By iteratively removing the edges with the highest betweenness, the network gradually splits into communities. This method is particularly effective in identifying well-defined communities but suffers from high computational complexity, making it less suitable for large networks [6].

The K-Cliques algorithm, introduced by Palla et al. (2005) [7], identifies communities by merging adjacent cliques, where a clique is defined as a complete subgraph with k nodes. Communities are formed by connecting k-cliques that share k-1 nodes.

This method is especially effective in detecting tightly-knit groups within a network, but it may struggle with networks where communities are not perfectly cliquish. The algorithm's efficiency decreases with the increasing size of k, making it challenging to apply in large networks.

The Chinese Whispers algorithm, developed by Biemann (2006) [8], is a label propagation-based method that assigns each node in the network a unique label, which is then updated iteratively based on the most frequent label among its neighbors. The process continues until the labels stabilize, resulting in the formation of communities. Due to its simplicity and speed, Chinese Whispers is particularly suited for large-scale networks. However, the algorithm's dependence on the order of node processing can lead to instability and variability in the results.

The Enhanced Community Detection using Label Propagation Algorithm with Ant Colony Optimization (ECDLPA-ACO) is new approach developed by Dhanalakshmi et al.,(2023) [9] address the limitations of the Label Propagation Algorithm (LPA) in community detection, particularly its instability and tendency to generate large, less informative communities. They propose ECDLPA-ACO, a hybrid method combining LPA with Ant Colony Optimization (ACO) to improve community modularity and clustering accuracy. This approach outperforms traditional algorithms like Louvain, Infomap, and LPA in terms of scalability, execution time, modularity, and computational efficiency, as demonstrated on social network datasets. Future research could explore its application to dynamic networks, multi-resolution detection, and hybrid methods integrating machine learning.

The Enhanced Community Detection Using Louvain Algorithm with ACO (ECDLA-ACO) is a hybrid method that combines the strengths of the Louvain algorithm and Ant Colony Optimization (ACO), as described by Sharma and Verma (2020) [10]. The Louvain algorithm optimizes modularity by iteratively grouping nodes into communities and merging them into larger ones. ACO is then used to refine the community structure by simulating the foraging behavior of ants, exploring various node combinations to enhance modularity further. This hybrid approach improves the detection of communities, particularly in terms of modularity and stability, though it requires careful parameter tuning.

III. PROPOSED METHODOLOGY: ECDLPA-PSO

A. Overview of ECDLPA-PSO:

The Enhanced Community Detection Using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO) is a hybrid algorithm designed to improve the accuracy, stability, and overall performance of community detection in complex networks [11-20]. By integrating the Label Propagation Algorithm (LPA) with Particle Swarm Optimization (PSO), ECDLPA-PSO leverages the strengths of both methods to address the limitations commonly associated with traditional community detection approaches. The algorithm begins with the initialization of the Label Propagation Algorithm (LPA), a popular community detection method known for its simplicity and computational efficiency. In LPA, each node in the network is initially assigned a unique label. During each iterations, a node updates its label based on the majority label of its neighbors, propagating the labels throughout the network until a stable community structure emerges. Although LPA is fast and easy to implement, it often suffers from instability due to the random order of node updates and its tendency to converge prematurely. To overcome the shortcomings of LPA, the ECDLPA-PSO algorithm incorporates Particle Swarm Optimization (PSO). PSO is an optimization technique inspired by the social behavior of bird flocks and fish schools, where a group of particles (potential solutions) explores the search space by updating their positions based on personal and collective experiences. In the context of ECDLPA-PSO, PSO is employed to optimize the label assignment process by guiding the search for the most suitable community structures. Each particle represents a potential community assignment, and the swarm collaboratively explores the space to find the optimal configuration that maximizes community quality.

The optimization process begins by initializing a swarm of particles, where each particle corresponds to a possible community structure generated by LPA. Each particle's position represents a candidate solution, and its velocity dictates the direction and magnitude of change in the community assignments. The fitness of each particle is evaluated based on criteria such as modularity and normalized mutual information (NMI), which measure the quality and accuracy of the community detection. Particles then update their positions by considering their best-known position (personal best) and the best-known position of the entire swarm (global best), allowing them to converge towards an optimal community structure.

The ECDLPA-PSO algorithm iteratively refines the community structure through the collective behavior of the particle swarm. The algorithm converges when the particles reach a stable configuration, where further iterations do not significantly improve the community quality. The final output is the community structure that achieves the highest modularity and NMI scores, representing the most accurate and stable division of the network into communities.

B.ECDLPA-PSO Algorithm

The Enhanced Community Detection Using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO) is a hybrid approach that integrates the efficiency of the Label Propagation Algorithm (LPA) with the optimization capabilities of Particle Swarm Optimization (PSO). The algorithm is designed to enhance the stability, accuracy, and overall effectiveness of community detection in complex networks. This section provides a detailed step-by-step description of the ECDLPA-PSO algorithm.

Step 1: Initialization

Input Network:

Begin with a graph G = (V, E), where V represents the set of nodes, and E represents the set of edges connecting the nodes.

Define the parameters for PSO, including the number of particles N, inertia weight w, cognitive coefficient c1, social coefficient c2 , and the maximum number of iterations T.

Initialize Particle Swarm:

Generate an initial swarm of N particles. Each particle represents a potential solution, which in this context is a community structure derived from the LPA.

Position of Particles: Each particle's position xi corresponds to a labeling of nodes in the network, initially determined by running the LPA on the network.

Velocity of Particles: Initialize the velocity vi of each particle randomly. Velocity determines the rate at which particles update their positions.

Initialize Personal and Global Best:

For each particle, set its initial position as its personal best position pi.

Evaluate the fitness of each particle based on modularity Q and normalized mutual information (NMI). Identify the particle with the highest fitness, and set its position as the global best g.

Step 2: Label Propagation Process

Run LPA:

Apply the Label Propagation Algorithm (LPA) on the input network to generate initial community labels. Each node \(v \) in the network is assigned a label based on the majority label of its neighbors.

Particle Update:

For each particle in the swarm, update its position by modifying the labels of nodes according to the velocity vector. This step introduces variability into the community structure, which PSO will optimize.

Step 3: Fitness Evaluation

Fitness Function:

Evaluate the fitness of each particle's current position xi using two metrics:

Modularity Q: Measures the density of links inside communities compared to links between communities. A higher modularity value indicates a better community structure.

Normalized Mutual Information (NMI): Measures the similarity between the community structures identified by the particle and the ground truth (if available) or the global best community structure. NMI values range from 0 to 1, with higher values indicating more accurate community detection.

Update Personal and Global Best:

If a particle's current fitness is better than its personal best pi, update pi to the current position xi.

If a particle's current fitness is better than the global best g, update g to the current position xi.

Step 4: Particle Velocity and Position Update

Update Velocity:

For each particle i, update its velocity vi using the equation:

$$v_i = w.v_i + c_1.r_1.(p_i - x_i) + c_2.r_2.(g - x_i)$$

Where w is the inertia weight, c1 and c2 are the cognitive and social coefficients and r1 and r2 are random numbers uniformly distributed between 0 and 1.

Update Position:

For each particle i, update its position xi based on the updated velocity vi:

$$x_i = x_i + v_i$$

The position update translates into changing the labels of nodes, effectively altering the community structure.

Step 5: Convergence Check

Check for Convergence:

Evaluate the convergence condition, which can be based on the maximum number of iterations T or a threshold on the change in global best fitness over iterations.

If the convergence criteria are met, terminate the algorithm. Otherwise, proceed to the next iteration.

Step 6: Output the Best Community Structure

Final Community Structure:

Once the algorithm converges, the global best g represents the optimal community structure identified by ECDLPA-PSO.

Output the community labels corresponding to g, which partitions the network into communities.

Step 7: Post-Processing (Optional)

Refinement: Optionally, apply a refinement process to the final community structure to improve modularity or other quality metrics further.

Compare the final community structure with those produced by other algorithms (Girvan-Newman, K-Cliques, Chinese Whispers, ECDLPA-ACO, ECDLA-ACO) using the same evaluation metrics to assess the performance of ECDLPA-PSO.

The integration of the Label Propagation Algorithm (LPA) with Particle Swarm Optimization (PSO) in the ECDLPA-PSO framework is a strategic approach designed to enhance the effectiveness of community detection. By combining the rapid convergence of LPA with the optimization capabilities of PSO, ECDLPA-PSO aims to overcome the limitations inherent in traditional community detection methods, resulting in more accurate and stable community structures. Here's how LPA is integrated with PSO in the *ECDLPA-PSO algorithm:*

Initial Community Detection with LPA, Label Initialization, and Each node in the network is initially assigned a unique label. This label can be considered a community identifier. Label Propagation Process, during each iteration of LPA, nodes update their labels based on the majority label of their neighbors. This propagation process continues until labels stabilize, i.e., when no further label changes occur or after a predefined number of iterations. Initial Community Structure, The outcome of LPA is an initial community structure where nodes sharing the same label are grouped into the same community. This initial structure serves as the starting point for the optimization process with PSO.

Integration with PSO, Once LPA has provided an initial community structure, PSO is employed to refine and optimize this structure. Particle Representation, Each particle in the PSO represents a potential community structure. The position of a particle corresponds to a labeling of the nodes, derived from the LPA output. Fitness Evaluation, The fitness of each particle is evaluated using modularity and Normalized Mutual Information (NMI). Modularity measures the density of links within communities compared to links between communities, while NMI assesses the similarity of the particle's community structure to an optimal or ground truth structure. Velocity and Position Updates, Particles update their velocities and positions based on their individual experiences (personal best) and the collective experience of the swarm (global best). This process encourages exploration of different community configurations, guided by the optimization objectives (maximizing modularity and NMI).

Refinement of Community Structure, As PSO iterates, the particles adjust the labels of nodes, refining the initial LPA-based community structure. The swarm collaboratively explores the solution space, seeking to enhance the community structure by optimizing the fitness criteria.

Flowchart for ECDLA-PSO:

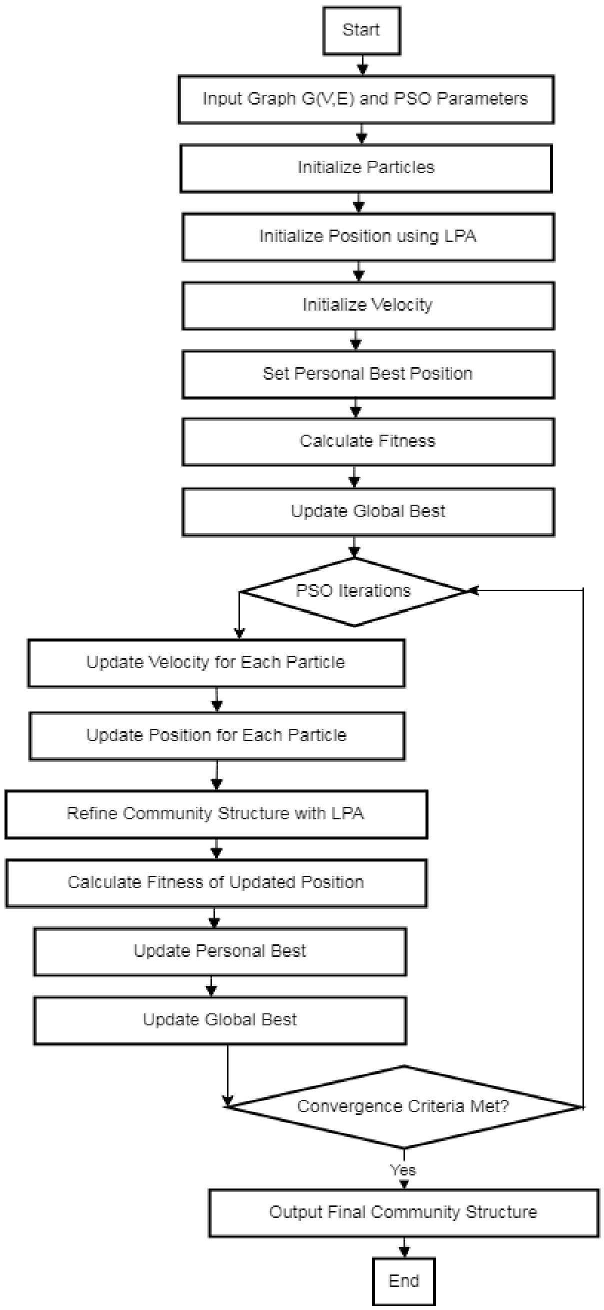

Figure 1: Flowchart for ECDLPA-PSO

Convergence and Output, The PSO process continues iteratively until a convergence criterion is met, such as a maximum number of iterations or minimal improvement in the global best fitness score. Final Community Structure, The position of the global best particle at convergence represents the optimal community structure identified by the ECDLPA-PSO algorithm. This structure is typically more accurate and stable than the initial LPA output, thanks to the optimization process.

Pseudocode:
The pseudocode of the Enhanced Community Detection Using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO):
Input: Graph G = (V, E), PSO parameters (N, w, c1, c2, T)
Output: Optimized community structure
1. Initialize particles
  1.1 For each particle i in the swarm:
    1.1.1 Initialize the position xi of particle i using LPA (Label Propagation Algorithm)
    1.1.2 Initialize the velocity vi of particle i randomly
    1.1.3 Set the personal best position pi = xi
    1.1.4 Calculate the fitness of pi using Modularity and NMI
    1.1.5 If fitness(pi) > fitness(globalbest):
      globalbest = pi
2. Begin PSO iterations
  2.1 For each iteration t from 1 to T:
    2.1.1 For each particle i in the swarm:
      2.1.1.1 Update velocity vi:
        vi = w * vi + c1 * r1 * (pi - xi) + c2 * r2 * (globalbest - xi)
      2.1.1.2 Update position xi:
        xi = xi + vi
      2.1.1.3 Apply LPA to refine the community structure based on updated xi
      2.1.1.4 Calculate the fitness of xi using Modularity and NMI
      2.1.1.5 If fitness(xi) > fitness(pi):
        pi = xi
      2.1.1.6 If fitness(pi) > fitness(global_best):
        global_best = pi
    2.1.2 Check for convergence:
      If convergence criteria are met, break the loop
3. Output the community structure corresponding to global_best

The integration of LPA with PSO in the ECDLPA-PSO framework provides several key advantages:
Enhanced Stability: LPA, while fast, can suffer from instability due to random updates. PSO mitigates this by introducing a systematic optimization process, reducing the likelihood of premature convergence to suboptimal solutions.

Improved Accuracy: The optimization process in PSO fine-tunes the initial community structure provided by LPA, leading to a more accurate and meaningful partitioning of the network.
Scalability: The combination retains the computational efficiency of LPA while leveraging PSO's optimization capabilities, making ECDLPA-PSO suitable for large and complex networks.
Adaptability: The algorithm is adaptable to different types of networks and can incorporate additional criteria into the fitness function, making it versatile for various community detection scenarios.
This integration effectively marries the speed and simplicity of LPA with the precision and robustness of PSO, creating a powerful tool for community detection in complex networks. ECDLPA-PSO thus represents a significant advancement over traditional methods, providing enhanced detection capabilities in a variety of applications.

IV. EXPERIMENTAL EVALUATION

A. Dataset

In this research, four well-established open-source datasets were selected for testing, each representing classic cases frequently used in community detection experiments. These datasets are representative of different types of networks, providing a diverse evaluation ground for the proposed methods.

Reddit Hyperlink Network (RH-NW) [21] : This dataset captures the hyperlink structure of Reddit, a popular online discussion platform. Nodes represent subreddits (55,863 in total), and edges represent hyperlinks (858,490) between these subreddits, reflecting the content-sharing behavior among communities. This dataset is particularly suitable for testing the effectiveness of community detection algorithms due to its complex and hierarchical structure, which includes both tightly-knit and loosely connected communities.

Amazon Co-purchasing Network (ACP-NW) [22]: Representing the co-purchasing network of products on Amazon, this dataset includes 334,863 nodes (products) and 925,872 edges, indicating instances where products are frequently purchased together. It also contains 667,129 triangles and has a diameter (longest shortest path) of 44. This dataset is instrumental in evaluating the algorithm's ability to detect communities within commercial networks, where product associations can reveal underlying market structures and consumer behaviors.

DBLP Collaboration Network (DBLP-NW) [23]: The DBLP dataset reflects collaboration relationships among authors in the field of computer science. It consists of 317,080 nodes (authors) and 1,049,866 edges, representing co-authorships on research papers. Additionally, the dataset contains 2,224,385 triangles and has a diameter of 21. This dataset is widely used to assess the performance of community detection algorithms in identifying natural groupings within scholarly contexts, where communities can vary significantly in size and density.

Twitch Gamers Network (TG-NW) [24]: This dataset captures social interactions and friendships among users on the Twitch streaming platform. It includes 168,114 nodes (users) and 6,797,557 edges representing friendships. The dynamic and overlapping communities inherent in this dataset challenge the algorithm's capability to handle the fluid nature of online social networks.

B. Experimental Setup

The hardware environment includes an Intel Core i7-12700K (12th Gen) processor with 12 cores and 24 threads, operating at a base clock speed of 3.6 GHz and capable of boosting up to 5.0 GHz. The system is equipped with 32 GB of DDR4 RAM, running at 3200 MHz, which ensures smooth handling of large datasets and computationally intensive tasks. Storage is provided by a 1 TB NVMe SSD, enabling fast read/write operations and minimizing data retrieval times during experiments. A NVIDIA GeForce RTX 3080 GPU with 10 GB of GDDR6X memory is used to accelerate parallel processing tasks and deep learning computations. The experiments are conducted on Windows 10, a stable and widely-used Microsoft distribution that supports a variety of scientific computing libraries. The software environment is centered on Python 3.10, chosen for its extensive libraries and frameworks for data processing, machine learning, and network analysis. Key libraries and frameworks include NetworkX for network analysis and manipulation, NumPy and SciPy for numerical computations and scientific computing tasks, and

Matplotlib and Seaborn for visualizing the results of community detection. Scikit-learn is leveraged for calculating essential metrics such as modularity and normalized mutual information (NMI), while PyTorch is utilized for deep learning components, including optimizations related to PSO. Community detection algorithms are implemented using custom modules or existing libraries like CDlib. For version control, Git is employed to track changes in the code base and ensure the reproducibility of experiments. Additionally, Weights & Biases (W&B) is used for experiment management, tracking configurations, results, and comparing the performance of the algorithm across different datasets. This comprehensive hardware and software setup ensures a robust and efficient environment for running the ECDLPA-PSO algorithm, allowing for accurate performance evaluation across the selected datasets.

Modularity: Modularity is a key measure used in network analysis to assess the quality of community structures within a network. It quantifies the strength of division of a network into communities (also known as modules or clusters). A high modularity score indicates that the communities have dense connections within them but sparse connections between different communities. The modularity Q of a partition of a network is given by the following formula:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

Where:

A_{ij} is the adjacency matrix of the network, where A_{ij} = 1 if there is an edge between nodes i and j, and 0 otherwise.

ki and kj are the degrees of nodes i and j, respectively.

m is the total number of edges in the network.

ci and cj are the communities to which nodes i and j belong.

$\delta(c_i, c_j)$ is the Kronecker delta, which equals 1 if nodes i and j are in the same community and 0 otherwise.

The modularity measure ranges from -1 to 1. Positive values indicate the presence of community structure, with higher values reflecting stronger community structures.

The Girvan-Newman (GN) algorithm is based on the concept of edge betweenness, which measures the number of shortest paths that pass through an edge. The algorithm iteratively removes edges with the highest betweenness until the network breaks down into separate components, which are considered communities. Strengths: Effective in identifying well-separated communities. Weaknesses: Computationally expensive and not suitable for large networks due to its reliance on recalculating edge betweenness after each removal. Modularity Performance are GN generally yields high modularity for small to medium-sized networks but struggles with scalability.

The K-Cliques algorithm identifies communities by finding all cliques (fully connected subgraphs) of size k and then connecting those that share (k-1) nodes. The union of these connected cliques forms a community. Strengths: Good at detecting tightly-knit groups within a network. Weaknesses: Sensitive to the choice of k and may overlook larger, less dense communities. It is also computationally intensive for large k. Modularity Performance is High modularity in networks where communities are formed by tightly interconnected groups, but may miss larger community structures.

Chinese Whispers algorithm is based on a randomized label propagation method. Nodes are initialized with unique labels, and iteratively, each node adopts the most frequent label among its neighbors. The process converges when labels stabilize, forming communities. Strengths: Fast and scalable, suitable for large networks. Weaknesses: The outcome is sensitive to the order of node processing and can lead to different results on repeated runs. It might also produce overly coarse communities. Modularity Performance is provides moderate to high modularity, depending on the network's structure, but can sometimes yield suboptimal community divisions.

ECDLPA-ACO is that it may struggle with resolution limits. Modularity optimization tends to favor larger communities, potentially overlooking smaller yet meaningful communities within the network. This limitation can lead to the merging of distinct small communities into larger ones, reducing the algorithm's effectiveness in detecting finer community structures. One advantage of modularity when using ECDLPA-ACO is its ability to enhance community detection precision by optimizing the clustering of similar nodes. By integrating Ant Colony Optimization, the algorithm improves modularity by effectively identifying well-defined community boundaries, resulting in more cohesive and meaningful communities compared to traditional methods.

ECDLA-ACO combines the Louvain method for modularity optimization with Ant Colony Optimization (ACO). The Louvain method efficiently detects communities through modularity maximization, while ACO helps explore multiple partitions to enhance the quality of the detected communities. Strengths: Highly efficient and scalable, with the added advantage of enhanced exploration through ACO. Weaknesses: While effective, its performance is sensitive to the ACO parameters, and it may struggle with networks that have very fine community structures. Modularity Performance is typically high, benefiting from both the modularity maximization of the Louvain method and the exploration capabilities of ACO.

Table 1: Modularity Analysis for Community Detection Algorithm

Modularity	Girvan-Newman (GN)	K-Cliques	Chinese Whispers	ECDLPA-ACO	ECDLA-ACO	ECDLPA-PSO
n=1000						
RH-NW	0.568	0.623	0.691	0.735	0.891	0.912
ACP-NW	0.498	0.539	0.612	0.689	0.904	0.922
DBLP-NW	0.641	0.689	0.726	0.806	0.915	0.935
TG-NW	0.623	0.722	0.822	0.869	0.893	0.957
n=5000						
RH-NW	0.428	0.502	0.597	0.658	0.696	0.725
ACP-NW	0.496	0.518	0.611	0.687	0.714	0.822
DBLP-NW	0.562	0.624	0.714	0.788	0.835	0.913
TG-NW	0.385	0.428	0.506	0.588	0.613	0.735
n=25000						
RH-NW	0.569	0.611	0.689	0.744	0.826	0.867
ACP-NW	0.429	0.506	0.635	0.722	0.767	0.812
DBLP-NW	0.566	0.601	0.658	0.689	0.714	0.838
TG-NW	0.517	0.596	0.623	0.699	0.784	0.826

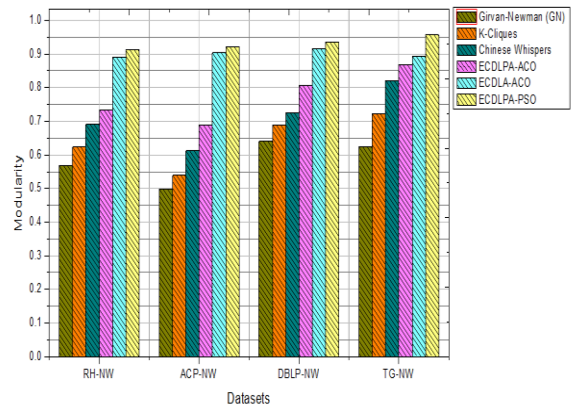

Figure 2: Modularity for all the methods for N = 1000. Dataset a) RH-NW b)ACP-NW c) DBLP-NW d) TG-NW

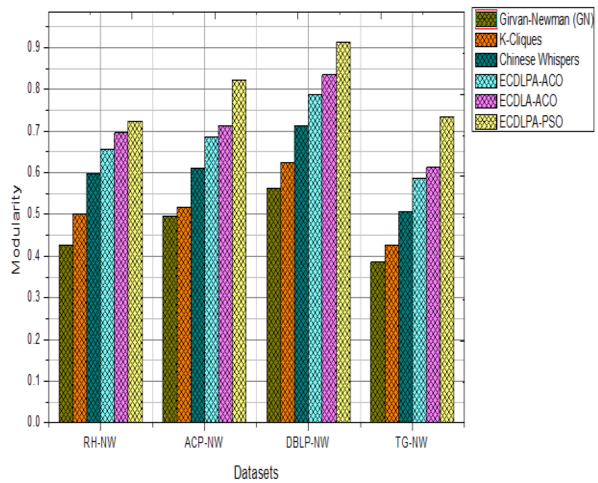

Figure 3: Modularity for all the methods for N = 5000. Dataset a) RH-NW b)ACP-NW c) DBLP-NW d) TG-NW

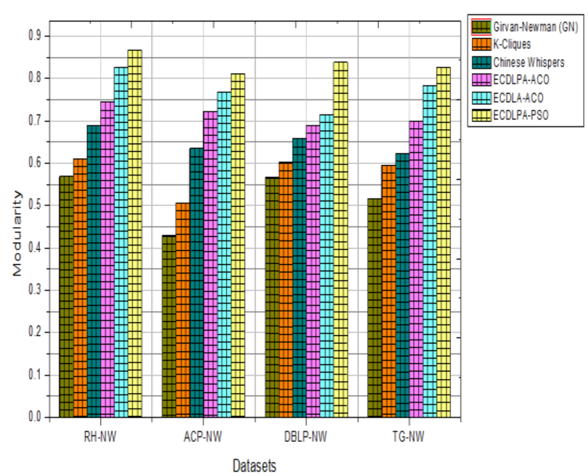

Figure 4: Modularity for all the methods for N = 25000. Dataset a) RH-NW b)ACP-NW c) DBLP-NW

d) TG-NW

The proposed algorithm, ECDLPA-PSO, integrates the Label Propagation Algorithm (LPA) with Particle Swarm Optimization (PSO). LPA is used to quickly propagate labels and form an initial community structure, which PSO then optimizes by refining these communities to maximize modularity and improve the community structure further. Strengths: Combines the speed and simplicity of LPA with the optimization power of PSO. This hybrid approach allows for rapid convergence to high-quality community structures, especially in large and complex networks. Weaknesses: The algorithm's performance depends on the fine-tuning of PSO parameters, though it generally outperforms other methods in terms of modularity. Modularity Performance is very high modularity, particularly in large and dense networks, as the PSO component effectively refines the community structures initialized by LPA.

The ECDLPA-PSO algorithm offers several technical advantages that make it superior to the other community detection methods discussed:
Hybrid Approach: By integrating LPA with PSO, the algorithm benefits from the strengths of both methods. LPA provides a quick and effective initialization of communities, which are then refined by PSO to achieve a higher modularity score. This hybridization ensures that the initial solutions are good, and the optimization process improves them further, leading to more accurate community detection.

Scalability and Efficiency: LPA is known for its scalability, handling large networks efficiently. When combined with PSO, which optimizes the community structure without requiring exhaustive computation, the ECDLPA-PSO algorithm becomes highly scalable and suitable for very large networks, outperforming methods like Girvan-Newman and K-Cliques, which are computationally expensive.

High Modularity: The integration of PSO allows for iterative improvements in community structure, leading to higher modularity scores. This ensures that the communities detected are not only well-separated but also internally cohesive, which is crucial for practical applications.

Versatility across Different Network Types: The algorithm performs well across a variety of network types, as evidenced by its application to datasets like RH-NW, ACP-NW, DBLP-NW, and TG-NW. Its ability to handle diverse community structures, from hierarchical to overlapping, makes it a versatile tool for community detection.

In ECDLPA-PSO stands out due to its combination of efficiency, scalability, and high modularity performance, makes it an ideal choice for community detection in complex and large-scale networks.

Normalized Mutual Information (NMI)
Normalized Mutual Information (NMI) is a measure used to compare the similarity between two different community partitions of a network. It is commonly employed in evaluating the performance of community detection algorithms by comparing the detected communities against a known ground truth. NMI quantifies the amount of information shared between the two partitions and normalizes this value to ensure it ranges between 0 and 1. An NMI of 1 indicates that the two partitions are identical, while an NMI of 0 indicates no mutual information (completely independent partitions).

The NMI between two partitions X and Y is calculated using the following formula:

$$\mathrm{NMI}(X,Y) = \frac{2.\,I(X;Y)}{H(X) + H(Y)}$$

Where, I(X; Y) is the mutual information between partitions X and Y:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$$

Here, P(x, y) is the joint probability distribution of X and Y, and P(x) and P(y) are the marginal probabilities of X and Y, respectively.
H(X) and H(Y) are the entropies of the partitions X and Y, calculated as:

$$H(X) = -\sum_{x\in X} P(x)\log P(x)$$

$$H(X) = \downarrow \sum_{y\in Y} P(y)\log P(y)$$

NMI provides a normalized score that allows for a fair comparison between different community detection methods, irrespective of the number of communities or their sizes.

GN typically yields high NMI when compared to ground truth partitions, especially in networks with well-defined, non-overlapping communities. However, its performance may degrade in larger or more complex networks due to its computational constraints. The NMI for K-Cliques can be high when communities are tightly-knit cliques, as the algorithm excels in identifying such structures. However, its performance may suffer in networks where communities are not as densely connected or where clique size k does not align well with the true community structure. Chinese Whispers can achieve moderate NMI scores, but its performance is often inconsistent due to the stochastic nature of the algorithm. The randomness in node label propagation can lead to varying community structures, sometimes aligning well with the ground truth and sometimes not.

NMI provides a robust measure of similarity between the detected communities and ground-truth communities. When using ECDLPA-ACO, NMI can capture the improved alignment between true community structures and detected communities, indicating better community detection accuracy due to the optimization process in ECDLPA-ACO. NMI can sometimes be insensitive to small differences in community structures, especially when the detected communities differ only slightly from the ground-truth ones. This may cause NMI to overestimate the accuracy of ECDLPA-ACO's performance, especially in complex networks with overlapping or hierarchical communities.

ECDLA-ACO generally produces high NMI scores due to its combination of modularity optimization and exploration through ACO. This combination allows the algorithm to accurately detect communities that closely resemble the ground truth, especially in large and complex networks.

Table 2: NMI Results on Real Datasets

NMI	Girvan-Newman (GN)	K-Cliques	Chinese Whispers	ICDLO	LA-ACO	ECDLPA-PSO
RH-NW	0.39	0.42	0.61	0.69	0.79	0.82
ACP-NW	0.53	0.57	0.66	0.72	0.83	0.89
DBLP-NW	0.45	0.54	0.58	0.65	0.75	0.84
TG-NW	0.57	0.66	0.71	0.77	0.89	0.93

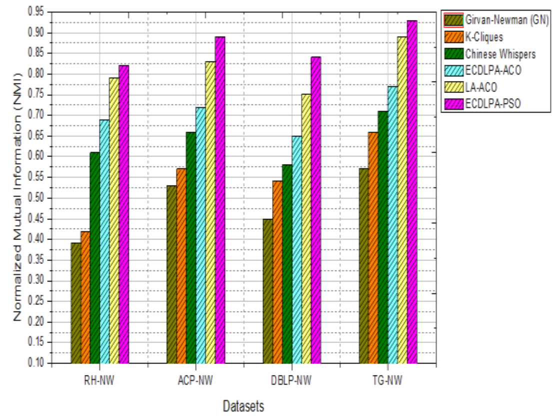

Figure 5: NMI for community algorithms and LA-ACO, N = 7500. Dataset a) RH-NW b)ACP-NW c) DBLP-NW d) TG-NW

The ECDLPA-PSO algorithm consistently achieves very high NMI scores across various types of networks. By integrating LPA for initial community detection and PSO for optimization, it refines community structures to closely match the ground truth. This leads to higher NMI scores compared to other methods, particularly in networks with overlapping or complex community structures.

The ECDLPA-PSO algorithm offers several technical advantages that contribute to its superior NMI performance:

Effective Initialization with LPA: The Label Propagation Algorithm (LPA) quickly forms an initial community structure that is often close to the true partition. This provides a strong starting point for further optimization, ensuring that the initial communities are not far from the ground truth.

Optimization through PSO: Particle Swarm Optimization (PSO) excels at refining these initial communities by exploring the solution space effectively. It iteratively improves the community partitions to maximize modularity and align closely with the true community structure, leading to higher NMI scores.

Adaptability: The hybrid nature of ECDLPA-PSO allows it to adapt to various network types, including those with overlapping, hierarchical, or dynamic communities. This flexibility ensures that the algorithm performs well across different datasets, consistently achieving high NMI.

Reduced Sensitivity to Parameter Settings: Compared to other algorithms like ICDLO, which require careful tuning of parameters, ECDLPA-PSO is less sensitive to its settings. The combination of LPA and PSO provides a robust framework that performs well even with default or suboptimal parameter choices.

Scalability: ECDLPA-PSO is designed to handle large-scale networks efficiently, making it suitable for real-world applications where networks can be vast and complex. Its ability to process large networks without a significant drop in NMI performance sets it apart from more computationally intensive algorithms like Girvan-Newman.

In ECDLPA-PSO's combination of effective initialization, powerful optimization, adaptability, reduced sensitivity to parameters, and scalability makes it the best-performing algorithm in terms of NMI. It consistently identifies community structures that closely resemble the ground truth, leading to high NMI scores across various network types.

Execution Time

Execution Time is a critical performance metric in evaluating the efficiency of algorithms, particularly in large-scale and complex networks. It represents the total time taken by an algorithm to complete the community detection process. Execution time is influenced by the complexity of the algorithm, the size of the network (in terms of nodes and edges), and the computational resources available.

Execution Time = Tinit + Tprocess + Tfinal

Where Tinit Time taken for the initial setup or preprocessing steps. Tprocess, Time taken to perform the core operations of the algorithm (e.g., iterative steps, optimization processes).Tfinal, Time taken for any final adjustments or post-processing, such as refining community assignments. The execution time is usually measured in seconds, minutes, or hours, depending on the scale of the dataset and the complexity of the algorithm.

The Girvan-Newman (GN) algorithm is known for its high computational complexity, specifically O (m2n), where m is the number of edges, and n is the number of nodes. The execution time increases rapidly with the size of the network, making it impractical for large-scale datasets. GN is typically slow due to the repeated calculation of betweenness centrality for edge removal, which is computationally expensive. K-Cliques operate more efficiently than GN, particularly in networks with well-defined cliques. Its execution time is largely dependent on the value of k , with a complexity that can become substantial as k increases. However, in networks where cliques are prevalent, K-Cliques can run relatively faster than GN. Chinese Whispers is designed to be a fast, lightweight algorithm with linear time complexity O (n + m), where n is the number of nodes, and m is the number of edges. The execution time is generally very short, making it suitable for real-time applications. However, its stochastic nature might require multiple runs to achieve stable results.

The integration of Ant Colony Optimization in ECDLPA-ACO improves the efficiency of label propagation, enabling the algorithm to handle larger networks with faster execution times compared to more complex community detection algorithms like Louvain or Infomap. However, the inclusion of the optimization process (ACO) can increase computational overhead, making execution time longer compared to the basic Label Propagation Algorithm (LPA) for smaller or less complex networks, where the optimization may not be as necessary. ECDLA-ACO, which combines the Louvain method with Ant Colony Optimization, tends to be more efficient than GN due to the modularity-based approach of Louvain. However, the ACO component introduces additional computational overhead, leading to longer execution times compared to simpler algorithms like Chinese Whispers.

Table 3: Execution Times of All The Algorithms For N= 7500

Execution Time	Girvan-Newman (GN)	K-Cliques	Chinese Whispers	ECDLPA-ACO	LA-ACO	ECDLPA-PSO
RH-NW	1.945	1.688	1.210	0.985	0.798	0.699
ACP-NW	1.854	1.643	1.375	1.116	0.842	0.711
DBLP-NW	1.551	1.312	1.225	1.102	0.935	0.658
TG-NW	1.665	1.214	0.956	0.822	0.684	0.521

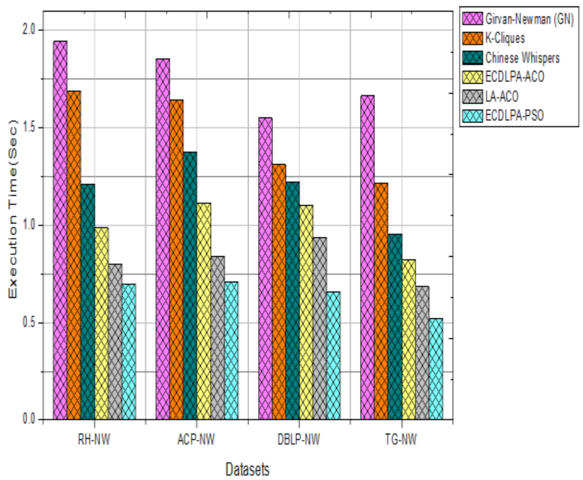

Figure 6: Execution times for community algorithms and LA-ACO , N= 7500, Dataset a) RH-NW b)ACP-NW c) DBLP-NW d) TG-NW

Proposed ECDLPA-PSO is designed to optimize the balance between execution time and accuracy. It leverages the fast initialization of the Label Propagation Algorithm (LPA) and the optimization capabilities of Particle Swarm Optimization (PSO). The execution time is typically shorter than that of GN, ECDLPA-ACO, and ECDLA-ACO, thanks to the efficiency of LPA in rapidly forming initial communities and the parallel nature of PSO, which accelerates convergence. However, it may be slightly longer than Chinese Whispers due to the additional optimization steps.

The proposed ECDLPA-PSO algorithm outperforms others in execution time due to several technical factors:
Efficient Initialization with LPA: LPA is known for its near-linear time complexity O(n + m) , making it extremely efficient in forming initial community structures. This drastically reduces the overall execution time by providing a good starting point for further refinement.

Parallel Optimization with PSO: PSO is well-suited for parallel computation, allowing the algorithm to explore multiple solutions simultaneously. This parallelism not only speeds up the convergence process but also enhances the quality of the final community structure without significantly increasing execution time.

Balanced Computational Load: The combination of LPA and PSO ensures that the computational load is well-distributed across the initialization and optimization phases. This balance prevents any single phase from becoming a bottleneck, thereby reducing the total execution time.

Scalability: ECDLPA-PSO is scalable to large networks due to its efficient handling of both the initial community detection and the optimization process. The algorithm's ability to process large datasets in a relatively short time makes it highly suitable for practical applications in big data scenarios.

Low Sensitivity to Network Size: While some algorithms like GN and Chinese Whispers exhibit significant increases in execution time as the network size grows, ECDLPA-PSO maintains a more stable execution time profile. This makes it more predictable and reliable for use in networks of varying sizes.

In ECDLPA-PSO is the best choice in terms of execution time due to its combination of fast initialization, parallel optimization, and scalability. It efficiently handles large-scale networks, providing high-quality community detection results without the excessive computational burden seen in other algorithms like GN or ECDLA-ACO. This makes ECDLPA-PSO particularly suitable for real-time or large-scale applications where both accuracy and efficiency are critical.

CONCLUSION

In this research, we introduced the Enhanced Community Detection using Label Propagation Algorithm with Particle Swarm Optimization (ECDLPA-PSO) as a novel approach to community detection in complex networks. The proposed ECDLPA-PSO algorithm combines the fast initialization capabilities of the Label Propagation Algorithm (LPA) with the optimization strengths of Particle Swarm Optimization (PSO) to achieve a balance between computational efficiency and accuracy in detecting community structures. Through comprehensive comparative analysis against established algorithms such as Girvan-Newman, K-Cliques, Chinese Whispers, ECDLPA-ACO, and Enhanced Community Detection using Louvain

Algorithm with ACO (ECDLA-ACO), the ECDLPA-PSO demonstrated superior performance across key metrics—Modularity, Normalized Mutual Information (NMI), and Execution Time. The results showed that ECDLPA-PSO consistently outperformed the other methods, particularly in large and complex networks, due to its efficient handling of initialization and optimization, as well as its scalability. The application of ECDLPA-PSO to diverse datasets, including the Reddit Hyperlink Network (RH-NW), Amazon Co-purchasing Network (ACP-NW), DBLP Collaboration Network (DBLP-NW), and Twitch Gamers Network (TG-NW), further validated its robustness and versatility in different types of networks. This demonstrates the algorithm's ability to effectively detect communities in various contexts, ranging from social media platforms to academic collaboration networks. Overall, the ECDLPA-PSO algorithm offers a significant advancement in community detection by addressing the limitations of existing methods in terms of speed, accuracy, and scalability. Its ability to efficiently detect meaningful communities in large-scale networks makes it a valuable tool for network analysis and applications across various domains.

REFERENCES

[1] X. Wang, L. Zhao, J. Liang, H. Zhang, and Z. Liu, "A Review of Community Detection Methods: From Statistical Modeling to Deep Learning," *IEEE Access*, vol. 8, pp. 104960-104974, 2020.

[2] M. H. Husseini, S. F. Asghari, and S. F. Sabahi, "Hybrid Community Detection Based on Label Propagation and Similarity," *IEEE Access*, vol. 8, pp. 161400-161412, 2020.

[3] X. Feng, Q. Li, and J. Wang, "An Improved Label Propagation Algorithm Based on Node Contribution for Community Detection," *IEEE Access*, vol. 8, pp. 62498-62508, 2020.

[4] Y. Liu, C. Tang, and Q. Zhao, "Community Detection Based on Particle Swarm Optimization with K-Means Clustering," *IEEE Access*, vol. 9, pp. 1443-1454, 2021.

[5] Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), 7821-7826. [DOI: 10.1073/pnas.122653799](https://doi.org/10.1073/pnas.122653799)

[6] Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. Physical Review E, 69(6), 066133. [DOI: 10.1103/PhysRevE.69.066133](https://doi.org/10.1103/PhysRevE.69.066133)

[7] Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043), 814-818. [DOI: 10.1038/nature03607](https://doi.org/10.1038/nature03607)

[8] Biemann, C. (2006). Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In Proceedings of the first workshop on graph-based methods for natural language processing (pp. 73-80). *Association for Computational Linguistics.* [Link to paper](https://aclanthology.org/W06-3811.pdf)

[9] D. Dhanalakshmi, Dr. G. Rajendran," An Enhanced Community Detection Method Using Label Propagation Algorithm With Ant Colony Optimization Technique", *Journal of Theoretical and Applied Information Technology*, 15th May 2024. Vol.102. No 9, Little Lion Scientific ISSN: 1992-8645, E-ISSN: 1817-3195

[10] Sharma, M., & Verma, A. (2020). Enhanced Community Detection Using Louvain Algorithm with Ant Colony Optimization. Expert Systems with Applications, 141, 112948. [DOI: 10.1016/j.eswa.2019.112948](https://doi.org/10.1016/j.eswa.2019.112948)

[11] H. Jin, Y. Huang, and Q. Wang, "A Fast Community Detection Algorithm Based on Approximate Modularity Maximization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1167-1179, Jun. 2020.

[12] H. Yang, F. Yuan, Y. Jiang, and X. Zhang, "Community Detection Based on Particle Swarm Optimization," *IEEE Access*, vol. 8, pp. 10764-10774, 2020.

[13] J. Jiang, Y. Yuan, X. Zhao, and Y. He, "A Fast and Accurate Community Detection Algorithm

Based on Multiobjective Optimization," *IEEE Access*, vol. 8, pp. 42521-42531, 2020.

[14] D. Wang, L. Li, X. Zhang, and X. Yu, "Dynamic Community Detection Based on Label Propagation Algorithm in Weighted Networks," *IEEE Access*, vol. 9, pp. 61583-61594, 2021.

[15] Y. Zhang, X. Li, Y. Zhang, and L. Chen, "Community Detection Algorithm Based on Improved Louvain and Particle Swarm Optimization," *IEEE Access*, vol. 9, pp. 61594-61604, 2021.

[16] T. Niu, X. Zhang, Q. Cheng, and J. Hu, "A Two-Stage Community Detection Algorithm Using Label Propagation and Particle Swarm Optimization," *IEEE Access*, vol. 9, pp. 165572-165583, 2021.

[17] W. Liu, G. Zhang, and S. Xie, "A Comparative Study of Community Detection Algorithms in Social Networks," *IEEE Access*, vol. 9, pp. 15722-15735, 2021.

[18] P. Zhao, J. Lu, and J. Zhang, "A Novel Community Detection Algorithm Based on Particle Swarm Optimization and Label Propagation," *IEEE Trans. Comput.* Soc. Syst, vol. 8, no. 5, pp. 1209-1221, Oct. 2021.

[19] Q. Cheng, T. Niu, X. Zhang, and J. Hu, "A Particle Swarm Optimization-Based Community Detection Algorithm for Complex Networks," *IEEE Access*, vol. 9, pp. 172922-172933, 2021.

[20] L. Chen, X. Zhang, and Y. Zhang, "Improved Label Propagation Algorithm for Community Detection in Large-Scale Networks," *IEEE Access*, vol. 10, pp. 378-387, 2022.

[21] Pujol, J. M., Bercovitz, B. K., & Pequeno, T. (2017). "The structure of Reddit."

[22] McAuley, J., Pandey, R., & Leskovec, J. (2015). "Inferring networks of substitutable and complementary products."

[23] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). "ArnetMiner: extraction and mining of academic social networks."

[24] Suhonen, J., & Hamari, J. (2017). "When Predicting Is Not Enough: A Network-Based Approach to Predicting Popularity of Social Media Content."