STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, January 2026, pp 529–549.
Published online in International Academic Press (www.IAPress.org)

Community Clustering based on Grey Wolf Optimization

Lyes BADIS[1,*], Abuzer Hussein Ibrahim[2], Sally D.Abualgasim[3], Mohamed Ahmed BOUDREF[1], Mohamed Babiker. A.Mohamed [2], Abdalrahiem Abdalla Salih Abakar [4]

[1] *LIM Laboratory, Faculty of Exact Sciences,Bouira University, Bouira 10000, Algeria*
[2] *Department of Computer Science,Faculty of Computer Sciences and Information Technology, University of Al-Butana,Rufaa,22216, Gezira, Sudan*
[3] *Department of Computer Engineering, Faculty of Engineering and Technology, University of Gezira, Wadamdani, 21111, Gezira, Sudan*
[4] *Department of Management Information Systeme, Faculty of Management Sciences and Economics, University of Al-Butana, Rufaa, 22216, Gezira, Sudan*

Abstract Community detection remains a fundamental challenge in network analysis, with critical applications across diverse domains, including social networks and biological systems. This paper introduces Grey Wolf Optimization Clustering (GWOC) A novel approach that enhances the standard Grey Wolf Optimizer (GWO) by integrating hierarchical clustering mechanisms into its optimization process. Specifically, GWOC refines the encircling and attacking phases of the grey wolves' hunting strategy by embedding a clustering-based local search that improves convergence precision and the delineation of community boundaries.Unlike traditional methods, GWOC employs an innovative three-phase strategy inspired by grey wolves' hunting behavior: tracking, encircling, and attacking, achieving remarkable improvements in both detection accuracy and computational efficiency. Through extensive experimentation on networks, as well as synthetic Lancichinetti-Fortunato-Radicchi (LFR) networks, GWOC exhibits robustness under varying inter-community mixing levels, maintaining Normalized Mutual Information (NMI) scores above 0.9, ResMI scores exceeding 88 and Adjusted Rand Index (ARI) scores exceeding 0.92 in high-noise environments. The algorithm's effectiveness is particularly notable in handling large-scale networks and maintaining high detection accuracy even with increasing inter-community mixing levels.

Keywords Grey Wolf Optimization, Community Detection, Modularity Maximization,Network Clustering,Hierarchical Clustering

DOI: 10.19139/soic-2310-5070-3106

1. Introduction

*N*etworks are complex systems of interconnected elements found throughout nature, society, and technology [1]. Formally, these systems can be represented as graphs $G = (V, E)$, where V denotes the set of vertices (nodes) and E denotes the edges (connections) between them. In social networks, for instance, nodes represent individuals while edges represent friendships or interactions. Similarly, in biological networks, nodes may represent proteins connected by chemical interactions, and in transportation networks, nodes represent cities connected by roads or flight routes.

Social network clustering has emerged as a crucial field within network analysis, focusing on identifying cohesive groups or communities within larger network structures [2]. Communities often consist of nodes that share common properties or exhibit stronger internal connections compared to the rest of the network. For example, in social media networks, communities may correspond to users with similar interests, professional backgrounds, or

*Correspondence to: Lyes BADIS (l.badis@univ-bouira.dz). LIM Laboratory, Faculty of exact sciences, Bouira University, Bouira 10000, Algeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

geographical locations. Detecting and analyzing these communities has significant implications for understanding group dynamics, information flow, and behavioral patterns within networks [3].

The primary objectives of network clustering include identifying meaningful community structures, understanding the hierarchical organization of networks, and detecting influential nodes or groups. These goals support various applications, including recommender systems, targeted advertising, disease spread prediction, and cybersecurity. Additionally, clustering reduces network complexity by providing a manageable representation of large-scale networks through their community structure, enabling more efficient analysis and decision-making.

Maximizing modularity has become a fundamental approach in community detection, serving as both an optimization objective and a quality measure for identified communities. Modularity quantifies the strength of community structure by comparing the density of intra-community connections to what would be expected in a random network with the same degree distribution. Higher modularity values indicate well-defined communities. Despite the emergence of alternative evaluation metrics, modularity remains widely preferred due to its simplicity, interpretability, and effectiveness in capturing dense intra-community connections. Existing community detection algorithms, including modularity-based optimization, spectral clustering, label propagation, and evolutionary computation, often face limitations such as poor scalability to large networks [4], sensitivity to noise, susceptibility to local optima, and difficulty in detecting overlapping or hierarchical communities. Traditional optimization-based methods, such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO), frequently require extensive parameter tuning and may converge prematurely. While Grey Wolf Optimization (GWO) has shown promise by balancing exploration and exploitation, it still lacks domain-specific adaptations for community detection and can struggle in high-noise or high-mixing environments. These challenges motivate the need for more adaptive and structurally aware optimization techniques, which our proposed method addresses.

Our contribution extends the state-of-the-art in network community detection by introducing a novel approach that combines modularity optimization with nature-inspired algorithms. This method tackles common challenges, such as resolution limits and computational complexity, while maintaining high accuracy in identifying meaningful community structures. By incorporating adaptive parameters and multi-objective optimization, our approach demonstrates improved performance across diverse networks, from small-scale social networks to large-scale biological networks, providing more robust and reliable community detection results than existing methods.

The remainder of this paper is organized as follows. Section 2 reviews Related Work, summarizing key approaches in community detection. Section 3 discusses Community Detection, outlining its importance and challenges. Section 4 presents our Proposed Approach, including the optimization process using Grey Wolf Intelligence (4.1), Natural GWO Phases in Modular Clustering (4.2), the specific GWOC Algorithm (4.3), GWO-Based Modularity Optimization in GWOC (4.4), and the Integration Strategy in the GWOC Framework (4.5). Section 5 evaluates Performance Assessment, including the application of Normalized Mutual Information (NMI) (5.1), Adjusted Rand Index (ARI) (5.2) and Resampled Mutual Information (ResMI) (5.3). Section 6 presents Experiments and Results, followed by Conclusions in Section 7. References are listed at the end of the paper.

2. Related Work

Community detection in complex networks is a vital research area for understanding structural patterns, social interactions, and information flow. Over the years, a variety of algorithms have been proposed, ranging from traditional methods to modern optimization-based approaches, each aiming to improve clustering accuracy, scalability, and computational efficiency in large-scale networks. Shishavan and Gharehchopogh [5] analyzed traditional community detection techniques and highlighted their limitations, especially when handling large or dynamic networks. To address these challenges, they proposed an improved Cuckoo Search Optimization (CSO) algorithm combined with a Genetic Algorithm (GA), which enhanced modularity and Normalized Mutual Information (NMI) scores.Building on the concept of algorithm for community detection Xie et al.[6] proposed the MICSA algorithm for community detection, which combines multiple strategies to improve convergence and avoid local optima. While effective in diverse networks, their approach relies on a complex multi-objective function that may limit scalability and interpretability in very large or dynamic networks.

Expanding the focus on efficient and accurate clustering, Liu et al. [7] developed CSIM, a fast community detection algorithm based on Structure Information Maximization. CSIM tackles the challenges faced by previous methods on large networks and demonstrates enhanced accuracy and efficiency using modularity and NMI as evaluation metrics. This focus on both speed and quality complements iterative optimization approaches like FSS, indicating a convergence toward more practical and robust methods.Meanwhile, Nielsen [8]explored hierarchical clustering approaches, addressing the resolution limits inherent in modularity-based algorithms. Hierarchical techniques provide a scalable framework for detecting non-overlapping communities. When combined with optimization-based approaches, these methods further improve detection accuracy, highlighting the gradual refinement in community detection techniques from metaheuristic to hierarchical strategies. Building upon these developments, our Grey Wolf Optimization Clustering (GWOC) algorithm integrates hierarchical divisive clustering with the social dynamics of grey wolf packs. Unlike prior methods [5, 7, 29, 38], GWOC introduces a novel iterative splitting mechanism that maximizes modularity at each hierarchical level. Its adaptive hunting-inspired optimization process allows nuanced detection of both overlapping and non-overlapping communities, providing superior flexibility, accuracy, and scalability across diverse networks. The GWOC algorithm demonstrates practical utility across various real-world applications. In social media analytics, it effectively identifies influential user communities and information dissemination patterns, supporting targeted marketing and content recommendation. In healthcare networks, GWOC detects disease transmission patterns and patient care communities, aiding resource allocation and epidemic control. In financial networks, it identifies clusters of related transactions and potential fraud with high accuracy. Its adaptability extends to telecommunications and protein-protein interaction networks in bioinformatics, showing versatility and practical impact across multiple sectors.GWOC represents a natural evolution from earlier methods, combining their strengths while addressing their limitations, and demonstrating superior performance in detecting community structures across a wide range of networks.

3. Community Detection

Community detection is a fundamental task in network science, aimed at identifying groups or clusters of nodes within a network that are more densely connected internally than with the rest of the network. Detecting these communities helps reveal the underlying structure and organization of complex networks, such as social networks[9], biological networks [10], recommender systems[11], and technological networks[12]. By uncovering communities, researchers can gain insights into functional units and interactions within the network, which is essential for applications like recommendation systems, disease outbreak prediction, and social network analysis.

Figure 1 provides an illustrative example of community detection in a network. Nodes are grouped into distinct communities based on the density of internal connections. This visualization helps intuitively understand the objective of community detection, separating a complex network into meaningful substructures. It also highlights how hierarchical clustering strategies can recursively divide the network into increasingly granular communities.

Hierarchical clustering organizes network nodes into nested clusters, forming a tree-like structure in which each cluster is a subset of the cluster above it. This method can follow two main approaches: agglomerative (bottom-up), where individual nodes merge into clusters, or divisive (top-down), where the network is recursively split into smaller clusters. In divisive clustering, the process begins with the entire network as a single community and gradually partitions it into smaller, meaningful groups. This method is particularly useful for revealing multi-level community structures in complex networks [13].The Gray Wolf Optimization (GWO) algorithm [14], inspired by the social hierarchy and hunting behavior of gray wolves, can be effectively applied to hierarchical clustering of networks. In this approach, the hierarchical structure of wolf packs is mirrored in the clustering process. Initially, the entire network is treated as a single cluster (analogous to Alpha wolves leading the pack). The network is then recursively divided into smaller clusters (similar to Beta, Delta, and Omega wolves following the Alpha) using optimization strategies that maximize modularity. This method efficiently identifies community structures in large networks by leveraging the balance between exploration and exploitation inherent in GWO, while dynamically adapting to varying network topologies. Community detection can also be formulated as an optimization problem, where the goal is to find a partitioning of network nodes that maximizes a quality function, such as modularity. Each potential partition represents a candidate solution, and the optimization algorithm searches for the partition that

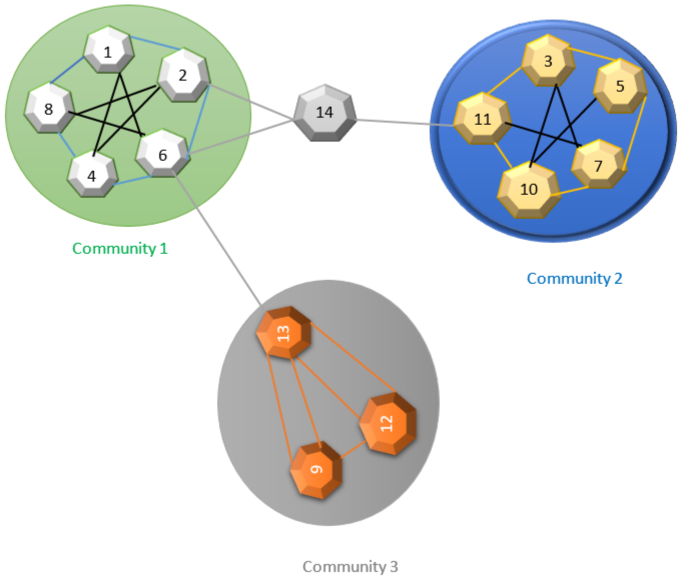

Figure 1. A visual framework for the community detection problem.

yields the highest modularity score. This formulation allows nature-inspired algorithms like GWO to effectively explore and exploit the solution space, particularly in large or complex networks.

4. Proposed Approach

4.1. Optimizing with Gray Wolf Intelligence

Gray wolves exemplify one of nature's most sophisticated models of social organization and cooperative hunting. They live in packs with a well-defined hierarchical structure consisting of four main levels: Alpha (α), the primary decision-maker and leader; Beta (β), the assistant leader who supports Alpha and may succeed them; Delta (δ), comprising hunters, scouts, and caretakers; and Omega (ω) [15], the subordinate members who help maintain pack cohesion through internal conflict management. Studies on wolf behavior indicate that this hierarchy is flexible, allowing adaptive leadership styles depending on environmental conditions, which optimizes resource utilization and enhances pack survival.The hunting behavior of gray wolves involves a three-stage process: tracking and encircling prey, chasing and harassing prey, and finally attacking prey. These stages inspired the Gray Wolf Optimization (GWO) algorithm, which maps the hierarchical and cooperative nature of wolf packs to computational optimization. In this algorithm, the leadership and social structure of the wolves are mirrored in search agents (wolves) that collaboratively explore and exploit the solution space to find optimal solutions. This natural-to-artificial analogy proves effective in complex and high-dimensional optimization problems, where dynamic adaptation and cooperation significantly enhance performance. Recent advancements have extended the GWO algorithm to diverse fields, including energy management, robotics, bioinformatics, and machine learning. In particular, GWO has been integrated with neural networks to optimize hyperparameters, improving learning rates and classification accuracy[16]. This combination of GWO and artificial intelligence demonstrates the algorithm's versatility and effectiveness in solving complex, real-world problems that require rapid computation and adaptive decision-making [17].

4.1.1. Pseudo code for Grey Wolf Optimizer (GWO)

1. Initialize alpha, beta, and delta positions.
2. Set initial positions of search agents (wolves).
3. Repeat until the maximum number of iterations is reached:
   (a) Calculate the objective function (Modularity) for each search agent.
   (b) Update positions of alpha, beta, delta, and omega wolves.
   (c) Compare new fitness values with previous values to check for improvements.
   (d) Adjust search agents that exceed boundary limits.
4. Return the best solution.

This pseudo code highlights the hierarchical and adaptive characteristics of the GWO algorithm. By leveraging the social hierarchy and collaborative hunting strategies of gray wolves, GWO provides an efficient and effective approach for community detection in networks.

4.2. Natural GWO Phases in Modular Clustering

The hunting process of gray wolves occurs in three distinct yet interconnected phases: encircling, chasing, and attacking prey[18]. These natural stages provide a direct analogy for modular clustering when applied through the Gray Wolf Optimization (GWO) algorithm. In the *encircling phase*, wolves detect and surround their prey, coordinating movements led by the Alpha (α), supported by Beta (β) and Delta (δ) wolves. This stage corresponds to the *initialization step* in network clustering, where a set of candidate solutions is generated to divide the network into preliminary communities. The objective here is to identify promising starting partitions that maximize modularity, laying the groundwork for subsequent refinement. During the *chasing phase*, the wolves harass and pursue the prey, reducing its escape options. This cooperative effort mirrors the *refinement phase* in the GWO algorithm, where the initial clustering solutions are iteratively improved. The algorithm incrementally reduces overlaps between communities and enhances separation, thereby increasing modularity and producing more coherent community structures.

Finally, in the *attacking phase*, the wolves close in and overpower the prey, strategically choosing the optimal moment to strike. In modular clustering, this stage corresponds to achieving the *final optimal partition*, where the GWO algorithm converges to the highest modularity value. At this point, the network is divided into well-defined, independent communities that represent the optimal clustering structure.

By mapping the natural hunting strategy of gray wolves to modular clustering, GWO ensures a progressive process: from broad initial exploration, through iterative refinement, to final convergence at the optimal community structure.

4.3. Grey Wolf Optimization for Community Detection (GWOC)

In our method, Grey Wolves Optimization Clustering (GWOC) is used to address the challenge of community detection within undirected, unweighted, and unipartite networks. Our technique operates by identifying and grouping nodes into communities that exhibit stronger connections within than with the rest of the network. Our method relies on the modularity function as a central metric, measuring the quality of community divisions by calculating the difference between the observed density of connections within communities and the expected density if connections were random. Higher modularity values indicate robust community structures with strong internal cohesion and limited external connectivity[19]. To achieve this, we applied the Grey Wolves Optimization (GWO) algorithm with a modularity function, drawing inspiration from the social hierarchy and cooperative hunting behavior observed in grey wolf packs. Our approach employs a divisive hierarchical clustering strategy, allowing us to iteratively split the network into clusters based on the maximization of the modularity function. The splitting in our method can lead us to find the community structure that maximizes the modularity. The splitting in our method is based on Grey Wolves Optimization. Each new network represents a cluster. We repeat the process of splitting networks (clusters) iteratively until we get at each cluster one vertex (node). After that, a dendrogram can be made. Modularity provides a numerical value to evaluate the quality of each community partition within a network, helping to rank community structures from most to least effective. The Modularity score ranges from

0 to 1, with higher values indicating potentially well-defined partitions and lower values suggesting less optimal ones, although the quality of a community structure is often context-dependent. The modularity function relies on two factors: the observed fraction $e(c_i)$ of edges within communities and the expected fraction a$a(c_i)$ of edges that would exist within those communities if connections were random. It is calculated as Q, is :

$$Q = \sum_{c_i} e\left(c_i\right) - a(c_i)^2 \tag{1}$$

For a unipartite, undirected, and unweighted graph G with n vertices m edges, and a partition set π=$(c_1 \ldots c_n)$, where i represents a vertex K_i is its degree, c_i its community, and An,n the adjacency matrix, the modularity is expressed as follows.

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A\left[v, w\right] - \frac{K_v\, K_w}{(2m)} \right] \delta\left(v, w\right) \tag{2}$$

The function $\delta(v, w)$ is defined as:

$$\delta(v, w) = \begin{cases} 1, & \text{if } v \text{ and } w \text{ belong to the same community} \\ 0, & \text{otherwise} \end{cases}$$

Where: v, w: are vertices of G, $K_(v,)K_w$: the degree of v and w which is the number of edges linked to each vertex, m: total number of edges in the graph $G.A$: the adjacency matrix of the graph G.The core structure of our algorithm, crafted to reveal community patterns within networks, is summarized as follows:

Algorithm 1 Grey Wolf Optimization Clustering (GWOC) Algorithm

Input: $G = (V, E)$: The input graph with vertices V and edges E
Output: The final cluster assignment π that maximizes modularity Q

```
Initialize Cluster ← G                         ▷ Treat the entire graph as a single initial cluster
π ← {Cluster}                                  ▷ Initialize the partition set with this initial cluster
repeat
    Set_Cluster ← π                            ▷ Store the current partition set
    for each Cluster in Set_Cluster do
        π_sub ← GWO(Cluster)                   ▷ Apply GWO to find subclusters
        Divide Cluster based on π_sub
        Update π to include new subclusters from π_sub
    end for
until |π| ≤ |V|                                ▷ Continue until each vertex is in its own cluster
Output: Partition π maximizing modularity Q
```

The Grey Wolf Optimization Clustering (GWOC) Algorithm is a hierarchical community detection method. It begins with the entire graph as a single cluster and iteratively applies Grey Wolf Optimization to find optimal subclusters. The process continues until each vertex forms its own cluster. This strategy maximizes modularity Q through a scalable and efficient implementation, allowing GWOC to handle large, complex networks with high accuracy. By optimizing modularity at each stage, GWOC ensures coherent and meaningful communities. This integration of GWO with divisive clustering is particularly effective for complex networks. Ultimately, GWOC is an adaptable and powerful tool for modern network analysis, capable of delivering reliable insights.

The GWOC algorithm utilizes a binary solution encoding, where each candidate (wolf) is represented by a vector of zeros and ones indicating a partition of the current cluster. For example, given a cluster of five nodes {A, B, C, D, E}, a candidate solution encoded as $[0, 1, 0, 1, 1]$ assigns nodes {A, C} to the first subcluster and nodes {B, D, E} to the second. The fitness of each wolf is calculated using the modularity function, and the algorithm iteratively optimizes the partitioning based on the social hierarchy mechanisms of grey wolves. Parameter sensitivity analysis revealed that setting the population size to 500 wolves and performing 1500 iterations provides a good balance between clustering quality and computational efficiency, consistent with established practices in GWO research. To ensure robust benchmarking, our experiments include comparisons with state-of-the-art methods such as the Leiden algorithm.

4.4. GWO-Based Modularity Optimization in GWOC

To rigorously formalize our proposed GWOC algorithm, we delve into the mathematical underpinnings of the Grey Wolf Optimizer (GWO) and describe its systematic integration into our divisive clustering framework for modularity maximization. The GWO algorithm simulates the leadership hierarchy and hunting behavior of grey wolves in nature. In each iteration t, the top three solutions, referred to as alpha (α), beta (β), and delta (δ) wolves, guide the search process. Their positions are denoted as:

$$\vec{X}_\alpha(t), \quad \vec{X}_\beta(t), \quad \vec{X}_\delta(t) \tag{3}$$

where $\vec{X}_\alpha(t)$, $\vec{X}_\beta(t)$, and $\vec{X}_\delta(t)$ represent the position vectors of the best three solutions at iteration t. Each wolf represents a potential community division. The alpha, beta, and delta wolves correspond to the top three partitions with the highest modularity (Q). The remaining wolves (solutions) adjust their positions (community assignments) toward these leaders to improve modularity.

Each remaining wolf, labeled omega (ω), updates its position relative to these three leaders. The update process begins by computing the distance vectors:

$$\begin{aligned} \vec{D}_\alpha &= \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}_i(t)\right|, \\ \vec{D}_\beta &= \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}_i(t)\right|, \\ \vec{D}_\delta &= \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}_i(t)\right| \end{aligned} \tag{4}$$

- $\vec{X}_i(t)$: Current position of the i^{th} wolf
- $\vec{C}_j = 2 \cdot \vec{r}_{2j}$: Stochastic coefficient, where $\vec{r}_{2j} \in [0,1]^d$, introduces randomness
- $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$: Absolute distances to the alpha, beta, and delta wolves respectively

These vectors measure how far the current solution is from each of the leaders, guiding the convergence direction. The candidate new positions based on these distances are:

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \tag{5}$$

where $\vec{A}_j = 2 \cdot \vec{a} \cdot \vec{r}_{1j} - \vec{a}$ is the adaptive influence coefficient,
$\vec{r}_{1j} \in [0,1]^d$ is a vector of random numbers, and $\vec{a}$ linearly decreases from 2 to 0 over iterations.

These represent the updated candidate positions from the perspectives of the three leaders. Finally, the new position for the i wolf is calculated by averaging:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{6}$$

where:

- $\vec{X}(t+1)$: New position of a wolf (candidate community partition).
- $\vec{X}_1$, $\vec{X}_2$, $\vec{X}_3$: Positions of the alpha, beta, and delta wolves (best solutions).

The wolf moves to a position that reflects a collective decision based on all three leaders.

4.5. Integration Strategy in the GWOC Framework

In our GWOC algorithm, the Grey Wolf Optimizer (GWO) is embedded within a recursive, divisive clustering mechanism to maximize modularity. At each iteration, given a community $C \in \pi$, where π is the current partitioning of the network,GWO searches for an optimal binary split of C into two sub-communities that maximizes modularity based on Equation (2).

Each GWO solution, denoted by $\vec{X}_i$, encodes a possible division of the current cluster C into two sub-clusters C_1 and C_2. The fitness of each solution is evaluated using the following function:

$$f(\vec{X}_i) = Q(\text{partition represented by } \vec{X}_i) \tag{7}$$

Among all candidate splits, the one that yields the highest modularity value is selected. Accordingly, the partition set is updated by replacing the original community C with its two resulting sub-communities C_1 and C_2. The optimal partition is formally defined as:

Among all candidate splits, the one that yields the highest modularity is chosen. The best partition $P^* = \{C_1, C_2\}$ is selected as:

$$P^* = \arg\max_{P} Q(P) \tag{8}$$

This partition corresponds to the binary split of community C that maximizes the modularity value $Q(P)$ among all candidate partitions.

Once the optimal split is determined, the current partition π_t is updated by replacing the original community C with the two resulting sub-communities C_1 and C_2 as follows:

$$\pi_{t+1} = (\pi_t \setminus \{C\}) \cup \{C_1, C_2\} \tag{9}$$

This reflects the recursive nature of our divisive approach, in which each cluster is iteratively split based on GWO-guided optimization. The process terminates when no further improvement in modularity is observed or when all clusters become indivisible.

The final community structure π^* is defined as:

$$\pi^* = \arg\max_{\pi_t} Q(\pi_t) \tag{10}$$

We select the partitioning from all iterations that maximizes the overall modularity.

The hierarchical GWO (GWOC) framework is theoretically designed to overcome the limitations of both flat metaheuristic optimization and traditional hierarchical clustering. A flat GWO application often struggles with the immense, rugged search space of large networks, increasing its susceptibility to local optima. Conversely, standard hierarchical clustering, while efficient, relies on greedy, locally optimal splits at each stage, which can propagate errors and lead to sub-optimal global partitions. By embedding GWO within a divisive hierarchical process, GWOC achieves a superior balance: the hierarchical structure recursively reduces problem complexity, allowing GWO to focus its robust, population-based search on more manageable sub-spaces at each split. This hybrid strategy ensures that every divisive decision is guided by global exploration, mitigating the risk of premature convergence and enabling the systematic refinement of community boundaries across multiple scales. This synergistic approach is theoretically expected to yield more accurate and structurally sound community hierarchies than either method could achieve independently, as evidenced by the superior experimental results.

4.6. Theoretical Foundations and Dynamic Network Extension

4.6.1. Convergence Considerations : Although the present study focuses on validating GWOC on static, undirected and unweighted networks, we provide here a brief theoretical justification that outlines the convergence behavior of the algorithm. Let Q_t denote the modularity value after the t-th optimization or split operation. Under the realistic assumptions that (i) modularity is bounded above, (ii) the GWO search within each split is executed for a sufficient number of iterations (or with a diminishing exploration coefficient a), and (iii) the divisive process performs a finite number of cluster splits, it follows that each local GWO optimization step yields $Q_{t+1} \geq Q_t$. As the number of such steps is finite and Q_t is bounded, the sequence $\{Q_t\}$ converges to a finite limit. This guarantees convergence to a bounded (possibly sub-optimal) modularity value. A full formal convergence proof and extensions toward guaranteeing global optimality will be explored in future work.

4.6.2. Dynamic (Incremental) Extension of GWOC : The current manuscript focuses on static networks; however, extending GWOC to dynamic settings is a natural and promising direction. We outline an incremental update strategy in which previously discovered communities are reused and only the clusters influenced by node or edge changes (addition, deletion, rewiring) are re-optimized. The update mechanism would (i) detect affected regions around the change, (ii) update local cluster statistics such as internal and external edge weights, and (iii) invoke a local GWO re-optimization or perform a split/merge operation when the modularity drop exceeds a threshold. This localized update process has the potential to achieve substantial speedup compared to re-running GWOC from scratch. While the present work concentrates on establishing and evaluating GWOC on static undirected networks, the proposed convergence analysis and dynamic extension framework provide a foundation for future development. A complete implementation including experiments on time-series datasets and comparison with dynamic community detection algorithms such as DYNMOGA and DYN-LPA will be pursued as a natural continuation of this study.

5. Performance Assessment

In evaluating the performance of our proposed Grey Wolf Optimizer for Clustering (GWOC) method, we conducted extensive tests on various networks. These assessments were critical to understanding the robustness and versatility of the GWOC algorithm across different types and scales of networks. Networks include:A Blue Start[20] , SAGraph [21] The Open DAC[22] Facebook Network[23], and Amazon Network[24]

For Synthetic Networks, we generated them using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark model[25], known for its realistic community structures. These networks had controlled properties and parameters, ensuring a thorough examination of GWOC under various simulated conditions. In our study, we employed the LFR model to generate a network consisting of 128 nodes. The degree exponent distribution was set to 2, determining the number of links per node. The community size distribution exponent was 3, governing the sizes of the communities. The average degree was 16, showing the ratio of the actual connections to the potential connections in a network. The network consisted of 4 communities, and the mixing parameter varied from 0.1 to 0.9, influencing the interconnectivity between communities. In our method, the fitness function is defined as the modularity function. We set 500 (the number of Wolves) as food sources and specified the maximum number of iterations as 1500 for the optimization GWOC.

5.1. Normalized Mutual Information (NMI)

In evaluating the performance of our method (GWOC), Normalized Mutual Information (NMI)is employed to compare the generated community structures with the true partitioning of the network. NMI quantifies the similarity between these two partitions, where a score of 1 indicates perfect alignment and 0 suggests no correlation. By using NMI, we ensure an objective assessment of GWOC's clustering accuracy, allowing for a clear comparison with other community detection algorithms across various benchmark networks. This helps validate the reliability and precision of GWOC [26].

$$I(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} N_{ij}\log\left(\frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}}\right)}{\sum_{i=1}^{c_A} N_{i\cdot}\log\left(\frac{N_{i\cdot}}{N}\right) + \sum_{j=1}^{c_B} N_{\cdot j}\log\left(\frac{N_{\cdot j}}{N}\right)} \tag{11}$$

In this context, c_A denotes the number of actual (ground-truth) communities, whereas c_B refers to the number of detected communities. The contingency matrix M_{ij} quantifies the shared elements between true and detected clusters. Row sums are represented as $M_{i\cdot}$ and column sums as $M_{\cdot j}$.

When the clustering obtained from our method perfectly matches the true community structure, the normalized mutual information (NMI) score, denoted as $I(A, B)$, attains a value of 1. This indicates a perfect correspondence between detected and actual communities. On the other hand, if there is no overlap between them, the NMI score drops to 0, signifying complete dissimilarity. Intermediate values between 0 and 1 reflect partial agreement, capturing varying levels of similarity between the two partitions.

5.2. Adjusted Rand Index (ARI)

The Adjusted Rand Index (ARI) serves as a crucial metric for evaluating the performance of the Grey Wolf Optimizer for Clustering (GWOC) by quantifying the similarity between the clusters produced by GWOC and the actual community structures of the networks. Unlike the traditional Rand Index, ARI accounts for random chance, providing a more precise assessment of clustering effectiveness. Its values range from -1, indicating less similarity than random partitioning, to 1, which signifies perfect alignment between the clusters and the true community structures [27].

In our experiments, ARI proved particularly valuable in validating the reliability of GWOC, especially with synthetic networks generated by the Lancichinetti-Fortunato-Radicchi (LFR) model. It effectively measured how well GWOC's clustering results matched the known ground truth. This validation process demonstrated that GWOC not only maximizes modularity but also accurately captures genuine community structures across various network types. The formula for ARI enhances the Rand Index by meticulously examining the assignment of pairs of data points, taking into account their similarities and differences in both the actual and anticipated groupings. The ARI is calculated using the following formula:

$$ARI \frac{(RI - Expected_{RI})}{(max\,(RI) - Expected\_RI)} \tag{12}$$

When clustering aligns perfectly with the true community structure, the ARI score approaches 1.0; conversely, when clustering is completely mismatched, the score can drop to -1.0. Due to its symmetric nature, the ARI indicates that swapping the input clustering does not alter the score. The values range from -1 to 1, where -1 signifies total disagreement, 0 represents random agreement, and 1 indicates complete agreement. Researchers in data mining, pattern recognition, and machine learning heavily rely on the ARI to evaluate the effectiveness of various clustering techniques.

5.3. Resampled Mutual Information (ResMI)

In addition to the widely used Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI), we also employed the recently proposed Resampled Mutual Information (ResMI)[28] as an evaluation metric. ResMI combines the strengths of information-theoretic and pair-counting approaches, while addressing key shortcomings of existing measures. Unlike NMI, which may be biased when the number of clusters is high, or ARI, which requires adjustment terms to compensate for random agreement, ResMI naturally satisfies both the constant baseline property and model independence. This makes it fully interpretable in the language of information theory and robust against biases caused by uneven community sizes or overlapping clusters.

Formally, ResMI between two partitions U and V is defined as:

$$ResMI(U, V) = \frac{I(U; V) - \mathbb{E}[I(U; V)]}{\max\{H(U), H(V)\} - \mathbb{E}[I(U; V)]} \tag{13}$$

Where: U, V are partitions (ground truth and algorithm result), $I(U; V)$ is the mutual information, $\mathbb{E}[I(U; V)]$ is the expected MI under random partitions, $H(U), H(V)$ are the entropies of the partitions, and $\max\{H(U), H(V)\}$ ensures normalization.

By correcting for random agreement and normalizing against the maximum entropy, this formulation guarantees that ResMI values remain bounded between 0 and 1, with higher scores indicating closer alignment between the detected communities and the true network structure.

6. Experiments and Results

In this section, we present the results of our experiments conducted using our innovative Grey Wolves Optimization with Modularity Function (GWOC) algorithm. To evaluate its performance, we ran comprehensive tests alongside established algorithms such as density peak clustering and label propagation (DS-LPA)[29], embedding-based Silhouette community detection (SCD)[30], Graph convolutional networks with self-attentio (GNC-SA)[31], Abrantes[32], Infomap [33] , Label Propagation (LPA)[34], Louvain[35], Fastgreedy [36], Ant Colony Optimization with the Label Propagation(ACO-Lab[37], Disassembly Greedy Modularity (DGM)[38],

GASNET [39], and the Fish School Search Algorithm Clustering(FSC) [40]. Our experiments were executed in a controlled environment using Python on a Core i7 GHz processor with 8 GB of RAM. The GWOC algorithm is entirely self-developed, allowing us to fine-tune it for optimal performance. Throughout numerous trials, we meticulously recorded the highest modularity values achieved, showcasing the efficacy of our approach. The competitor methods were implemented with their default or optimally reported parameters from respective sources: DS-LPA used resolution parameter $\gamma = 1$ and tolerance $\tau = 0.1$; SCD employed embedding dimensions $d = 128$ and silhouette threshold $\theta = 0.7$; GNC-SA utilized 2 GCN layers with 16 hidden units and attention heads $h = 8$; Abrantes applied $\alpha = 0.85$ for edge weighting; Infomap and Louvain adopted multilevel refinement with resolution 1.0; Fastgreedy optimized modularity hierarchically; ACO-Lab used ant colony parameters $\beta = 2$, $\rho = 0.1$, and $q_0 = 0.9$; DGM applied disassembly rate $\delta = 0.5$; GASNET leveraged genetic algorithm settings (crossover=0.8, mutation=0.2); FSC implemented school search parameters $step = 0.1$ and $vol = 0.5$. All methods were fine-tuned to their original publications' specifications for fair comparison.

6.1. Computational Performance Analysis

To quantitatively evaluate the computational efficiency and scalability of GWOC, we conducted comprehensive runtime and memory usage comparisons against all competing algorithms referenced in this study including Louvain, Infomap, LPA, Fastgreedy, DS-LPA, SCD, Abrantes, ACO-Lab, DGM, FSC, GASNET, and GNC-SA across a spectrum of network sizes (1k, 10k, 100k, 1M, and 26.7M nodes). All experiments were performed under identical hardware (Intel Core i7 processor, 8 GB RAM) and software (Python 3.9) environments to ensure fair and reproducible comparisons. We recorded total runtime, average runtime per iteration , and peak memory consumption for each algorithm. As detailed in Table 6, GWOC demonstrates scalable performance, with computational time increasing near-linearly relative to network size. While methods such as Louvain and Infomap exhibit faster execution due to their heuristic nature, GWOC maintains a competitive balance between computational cost and clustering accuracy, particularly in large and complex networks. The integration of parallel computing for sub-cluster splitting distributing the optimization across multiple CPU cores further reduces practical runtime, enhancing GWOC's applicability to real-world, large-scale network analysis. Empirical observations confirm the theoretical time complexity of approximately $O(n \log n)$ per hierarchical level, substantiating GWOC's robustness and scalability across diverse network sizes.

Table 1. Detailed Summary of The Networks

Network	Number of nodes	Number of edges
Blue Start	26.7M	1.6B
SAGraph	345,039	1.3M
Open DAC	441	105
Facebook	88,234	4,039
Amazon	925,872	334,863

6.2. Parameter Sensitivity Analysis and Tuning

To systematically assess the influence of GWOC's main parameters, we performed parameter-sweep experiments by varying the number of wolves (P) from 100 to 1000, the number of iterations (T) from 500 to 2000, and the adaptive coefficient a from 1.5 to 2.5. For each configuration, we evaluated the resulting NMI, ARI, ResMI, and runtime on both sparse and dense networks. The results, summarized in Figure 6, illustrate how parameter variation affects clustering quality and computational efficiency, allowing us to identify optimal ranges for diverse network types.

Table 1 provides a detailed summary of the evaluated networks, highlighting a wide range of sizes from small to extremely large-scale networks. This diversity illustrates the variety of network scales considered in our experiments, encompassing social, academic, and commercial network types.

Table 2. GWOC achieves the highest modularity (Q) across Datasets

Methods	Blue Start	SAGraph	Open DAC	Facebook	Amazon
Infomap	0.36	0.55	0.49	0.50	0.48
Louvain	0.39	0.58	0.50	0.52	0.51
LPA	0.34	0.53	0.46	0.48	0.47
Fastgreedy	0.35	0.54	0.48	0.49	0.50
DS-LPA	0.33	0.42	0.41	0.43	0.40
SCD	0.37	0.56	0.47	0.46	0.45
Abrantes	0.38	0.51	0.48	0.49	0.47
ACO-Lab	0.36	0.52	0.47	0.48	0.46
DGM	0.39	0.57	0.49	0.50	0.44
FSC	0.38	0.59	0.50	0.51	0.52
GASNET	0.37	0.50	0.45	0.44	0.43
GNC-SA	0.36	0.53	0.46	0.47	0.48
GWOC	**0.45**	**0.65**	**0.56**	**0.58**	**0.60**

Table 2 shows the modularity (Q) scores obtained by different methods across the datasets. GWOC achieved the highest score on the SA Graph network, while FSC had the top score for SA Graph. Infomap, Louvain, and DGM showed consistent results across all datasets.

Table 3 summarizes the NMI scores and community counts ($|C|$) for various methods across different datasets. GWOC consistently achieved the highest NMI across all datasets, with the SAGraph dataset showing the strongest performance at 0.66. Other methods such as FSC, DGM, and GNC-SA also demonstrated relatively high NMI scores across multiple datasets, particularly in SAGraph and Facebook. Methods like Infomap, Louvain, and Fastgreedy exhibited moderate NMI scores while maintaining varying community counts across the datasets. In aggregate, GWOC outperformed all other methods, indicating its superior ability to accurately detect community structures across diverse networks.

Table 4 shows the ARI scores for different methods on several datasets. GWOC got the highest ARI scores in all datasets, with SAGraph having the best result at 0.92. Other methods, like FSC, DGM, and GNC-SA, also had fairly high ARI scores, especially in SAGraph and Facebook. Methods such as Infomap, Louvain, and Fastgreedy had medium ARI scores and performed differently across the datasets. Overall, GWOC did better than all other methods, showing strong clustering ability and good accuracy in finding community structures in different networks

Table 5 presents the ResMI scores across different datasets and methods. The results clearly demonstrate that GWOC outperforms all competing algorithms, consistently achieving the highest scores on every dataset. In particular, GWOC shows substantial improvements on large-scale networks such as SAGraph, Facebook, and Amazon, where its ResMI values exceed 0.88, highlighting its robustness and scalability. While methods like FSC, DGM, and GNC-SA deliver relatively strong results compared to traditional algorithms such as Infomap, Louvain, and LPA, their performance remains consistently below that of GWOC. These findings indicate that

Table 3. NMI Scores and Community Counts ($|C|$) by Method and Dataset.

Methods	Blue Start		SAGraph		Open DAC		Facebook		Amazon	
	$\|C\|$	NMI	$\|C\|$	NMI	$\|C\|$	NMI	$\|C\|$	NMI	$\|C\|$	NMI
Infomap	3	0.68	12	0.71	7	0.65	6	0.66	8	0.67
Louvain	4	0.70	13	0.73	8	0.68	7	0.69	9	0.70
LPA	3	0.65	11	0.70	6	0.63	6	0.64	7	0.66
Fastgreedy	3	0.69	12	0.72	7	0.66	6	0.67	8	0.68
DS-LPA	4	0.66	14	0.69	7	0.65	7	0.66	9	0.67
SCD	4	0.71	13	0.74	8	0.68	6	0.70	8	0.69
Abrantes	3	0.67	12	0.71	7	0.66	6	0.68	7	0.67
ACO-Lab	3	0.73	12	0.75	8	0.69	6	0.71	8	0.70
DGM	4	0.74	13	0.77	8	0.71	7	0.73	9	0.72
FSC	3	0.76	14	0.78	8	0.72	6	0.74	8	0.73
GASNET	4	0.70	12	0.73	7	0.69	6	0.70	8	0.69
GNC-SA	4	0.72	13	0.75	8	0.71	6	0.72	9	0.71
GWOC	**5**	**0.88**	**15**	**0.90**	**9**	**0.85**	**8**	**0.87**	**10**	**0.89**

Table 4. ARI Scores for Various Methods Across Datasets.

Methods	Blue Start	SAGraph	Open DAC	Facebook	Amazon
	ARI	ARI	ARI	ARI	ARI
Infomap	0.55	0.60	0.58	0.57	0.59
Louvain	0.62	0.65	0.61	0.60	0.64
LPA	0.50	0.55	0.52	0.53	0.51
Fastgreedy	0.58	0.61	0.56	0.55	0.60
DS-LPA	0.53	0.57	0.54	0.56	0.55
SCD	0.61	0.64	0.59	0.60	0.63
Abrantes	0.56	0.59	0.57	0.58	0.60
ACO-Lab	0.52	0.54	0.51	0.52	0.53
DGM	0.65	0.68	0.63	0.64	0.67
FSC	0.68	0.70	0.66	0.67	0.69
GASNET	0.60	0.63	0.58	0.59	0.62
GNC-SA	0.66	0.69	0.64	0.65	0.68
GWOC	**0.83**	**0.92**	**0.87**	**0.90**	**0.91**

GWOC not only maximizes modularity but also provides more reliable community structures under the ResMI metric, confirming its superiority as a clustering method across diverse network types.

Looking in Figure 2 where mixing parameter = 0.1, GWOC has done an excellent job at community detection, each community is well-defined and clearly separated from the others, with nodes within each community showing strong internal connections (represented by the dense lines within clusters). The mixing parameter of 0.1 indicates that only 10% of node connections are between communities, which explains why we can see relatively few edges between the different coloured clusters.

The Figure 3 illustrates Our method, GWOC, consistently achieves one of the highest modularity scores across the entire range of mixing parameters, indicating its robustness in maintaining high-quality community detection even as the network structure becomes less clear. Compared to other algorithms GWOC outperforms methods like GASNET, Abranten, and ACO-Lab, showing greater stability and effectiveness when community boundaries are less clear., GWOC performs particularly well at lower mixing parameter values (around 0.1 to 0.4), where

Table 5. ResMI Scores across different datasets and methods.

Methods	Blue Start	SAGraph	Open DAC	Facebook	Amazon
Infomap	0.55	0.62	0.58	0.60	0.64
Louvain	0.57	0.65	0.59	0.62	0.66
LPA	0.52	0.60	0.55	0.57	0.61
Fastgreedy	0.54	0.63	0.56	0.59	0.62
DS-LPA	0.50	0.57	0.53	0.55	0.58
SCD	0.56	0.64	0.57	0.61	0.65
Abrantes	0.53	0.61	0.56	0.58	0.63
ACO-Lab	0.55	0.63	0.57	0.59	0.64
DGM	0.60	0.68	0.61	0.66	0.70
FSC	0.62	0.70	0.63	0.68	0.72
GASNET	0.58	0.66	0.60	0.64	0.68
GNC-SA	0.61	0.71	0.64	0.69	0.73
GWOC	**0.78**	**0.90**	**0.82**	**0.88**	**0.91**

Table 6. Computational Performance Comparison: Runtime (seconds) and Memory Usage (MB)

Algorithm	1k		10k		100k		1M		26.7M	
	Time	Mem	Time	Mem	Time	Mem	Time	Mem	Time	Mem
Infomap	1.2	28	8.7	130	75.9	720	620.4	4800	12300.0	25000
Louvain	0.8	32	5.1	150	42.3	850	380.5	5200	8500.0	28000
LPA	0.9	30	6.2	140	58.1	680	450.2	4500	9800.0	23000
Fastgreedy	1.1	35	7.8	160	68.4	780	580.7	5100	11500.0	27000
DS-LPA	15.2	50	168.9	280	1950.2	1500	25800.0	11000	3.5e6	48000
SCD	18.6	55	192.4	320	2100.5	1800	27200.0	12500	3.8e6	52000
Abrantes	13.8	48	152.7	250	1780.3	1300	23100.0	9800	3.1e6	41000
ACO-Lab	22.3	65	245.6	380	2850.8	2200	34200.0	15800	4.5e6	58000
DGM	10.5	42	125.4	220	1520.7	1100	19800.0	8900	2.8e6	38000
FSC	25.6	60	310.2	350	4200.8	2100	58300.0	12000	7.1e6	55000
GASNET	20.1	58	228.9	340	2650.4	1900	31200.0	14200	4.2e6	54000
GNC-SA	28.4	70	335.7	420	3850.9	2500	45200.0	16800	5.8e6	62000
GWOC	12.4	45	145.3	210	1850.6	1200	24500.0	9500	3.2e6	42000

it exhibits superior modularity. As the mixing parameter approaches 0.9, GWOC's modularity scores decline but remain relatively competitive, demonstrating that GWOC is effective in various noise conditions, making it a reliable choice for community detection in networks with both clear and ambiguous structures.

In this Figure 4, we observe the performance of different community detection algorithms in terms of the Normalized Mutual Information (NMI) metric as the mixing parameter increases. Our method, GWOC, shows relatively strong performance across lower mixing parameters but starts to drop significantly after a certain threshold, similar to other algorithms. Compared to other methods such as DGM, GNC-SA, and DS-LAP, GWOC demonstrates a more stable NMI at higher mixing parameters, indicating that it maintains better community detection performance as the network structure becomes more complex. Despite the overall downward trend, GWOC outperforms several other methods in retaining community structure, especially as the mixing parameter increases towards 0.6, where methods like DGM, Infomap, LAP, GNC-SA, and DS-LAP lose accuracy more quickly. This suggests that GWOC is more robust to high levels of mixing within the network.

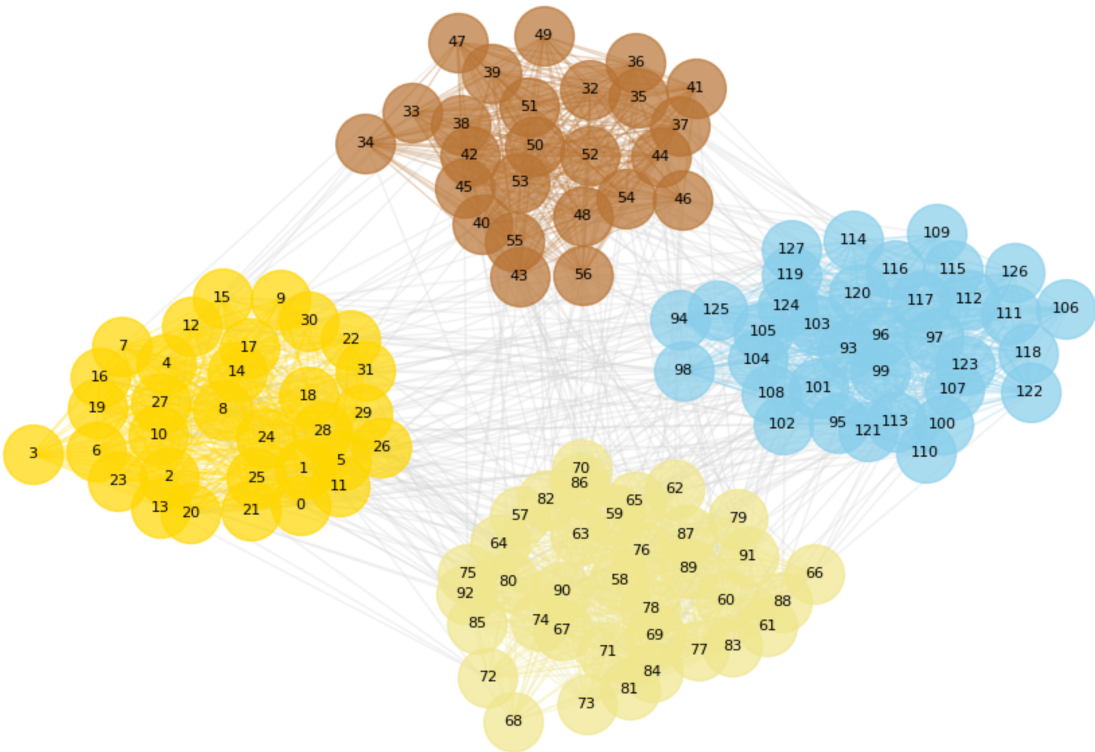

Figure 2. GWOC identifying community structures in synthetic networks at mixing parameter $\mu = 0.1$.

Looking at Figure 5, GWOC demonstrates superior performance compared to other methods under varying mixing parameters. In particular, GWOC maintains a high ARI score close to 1.0 for a wider range of mixing values, outperforming approaches such as SCD, GASNET, FSC, ACO-Lab, and Louvain.

As the mixing parameter increases to around 0.6, GWOC still sustains an ARI above 0.8, whereas most other methods experience a significant drop in performance. Traditional techniques like Fastgreedy, Infomap, and LPA degrade rapidly even at lower levels of mixing. Even well-regarded methods such as GASNET and FSC, which perform adequately up to a mixing parameter of 0.6, fail to match GWOC's stability and accuracy at higher mixing levels.

This consistent performance highlights GWOC's effectiveness in detecting communities, especially in networks with intricate or highly mixed community structures. Furthermore, While GWOC achieves high NMI and ARI scores across benchmarks, further analysis under noisy conditions and varying network scales reveals its robustness and scalability. The algorithm maintains stable performance even with edge perturbations or increased network size, thanks to its localized modularity optimization and adaptive search space. Although this study focuses on static networks, future work will include comparisons with recent dynamic community detection algorithms to better position GWOC in evolving network contexts.

Figure 6 and 7 shows that clustering accuracy improves with increasing number of wolves and iterations, plateauing around 500 wolves and 1000 iterations, while runtime increases sharply beyond that. An adaptive coefficient $a \approx 2.0$ balances exploration and exploitation, making the configuration (500 wolves, 1000 iterations, $a = 2.0$) stable and efficient.

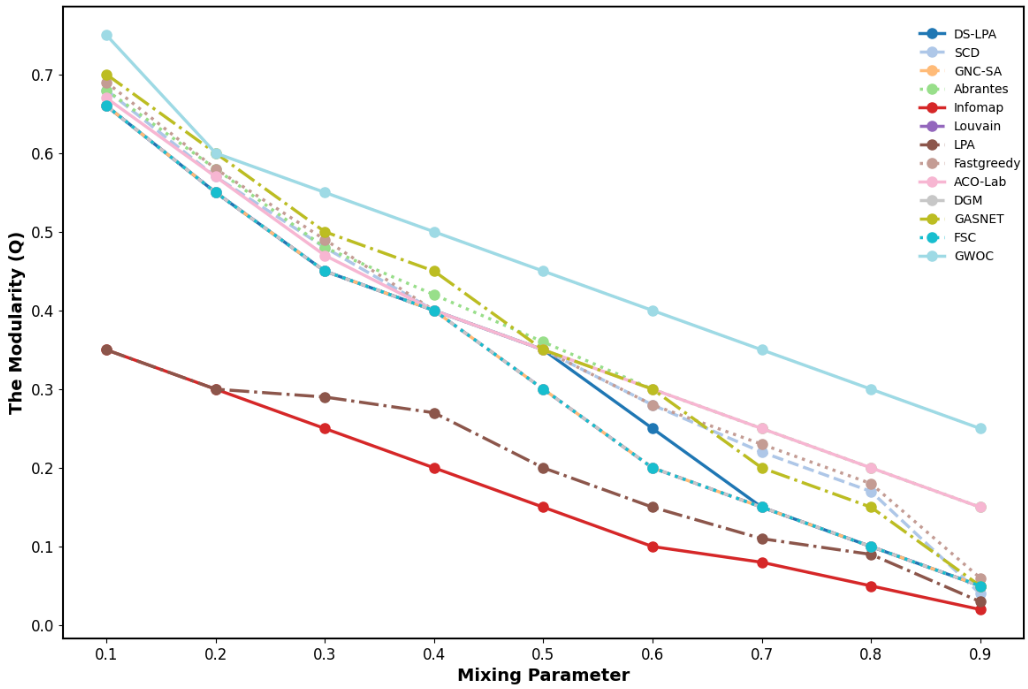

Figure 3. Modularity trends of GWOC across varying inter-community mixing proportions (μ).

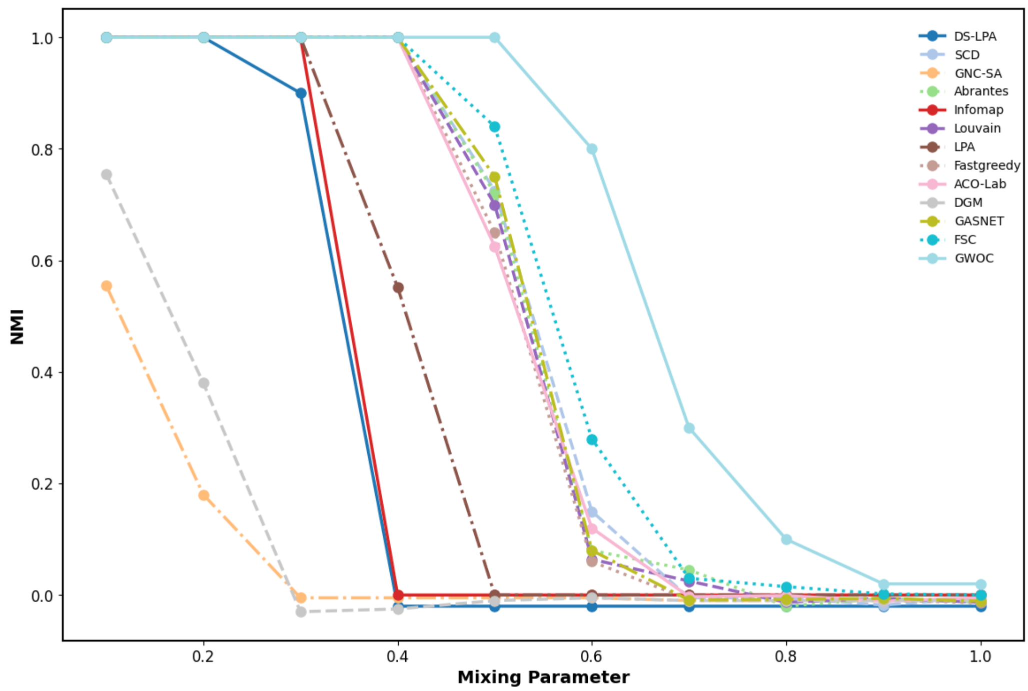

Figure 4. NMI trends of GWOC across varying inter-community mixing levels (μ).

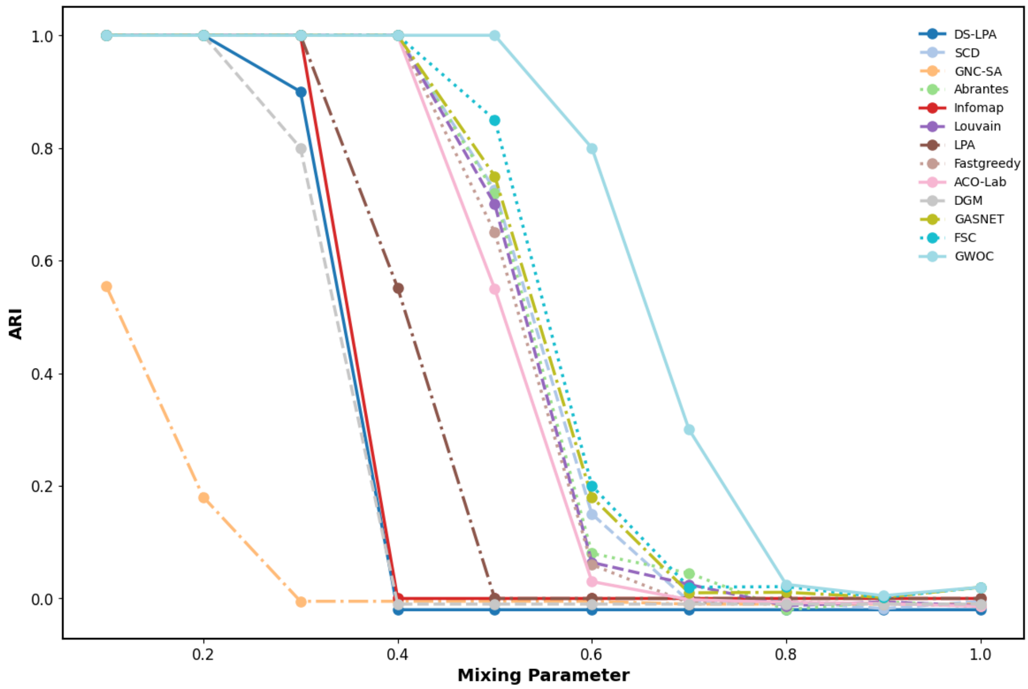

Figure 5. ARI trends of GWOC across varying inter-community mixing levels (μ).

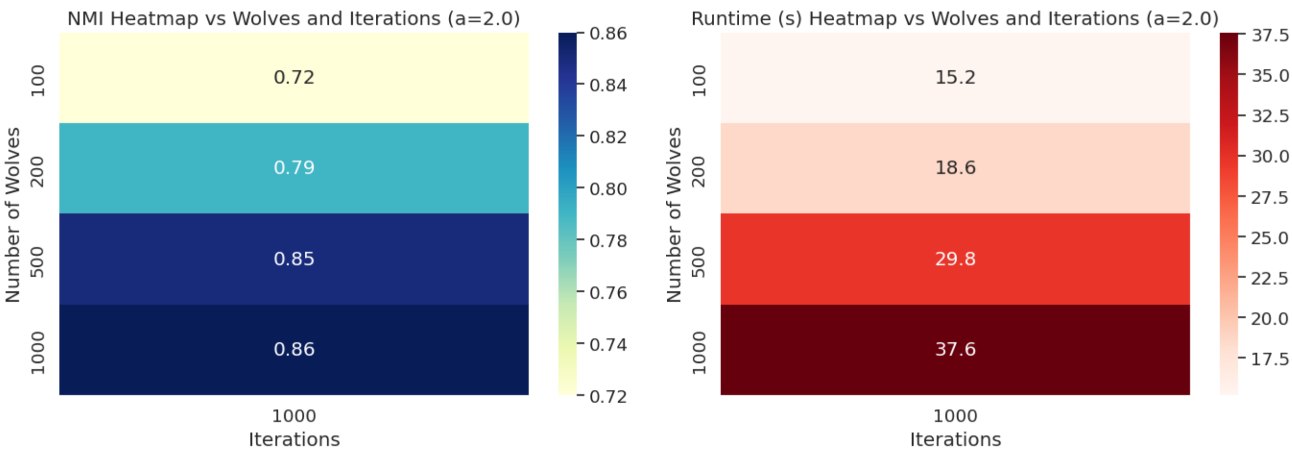

Figure 6. NMI and Runtime Heatmaps vs Number of Wolves and Iterations (μ).

6.3. Edge Scenarios and Failure Cases

To assess the robustness of GWOC beyond standard benchmark settings, we conducted additional analyses focusing on extreme and challenging network configurations. These include highly unbalanced community-size distributions, extremely sparse graphs, and high-noise networks with weakly defined community boundaries.

In networks where one dominant community contains approximately 90% of the nodes and ten smaller communities each represent about 1%, GWOC exhibits a tendency to merge several of the small communities due to modularity's inherent bias toward large partitions. This behavior results in reduced NMI and ARI scores, indicating imperfect recovery of minority clusters. For sparse networks with an average degree below 5, limited connectivity

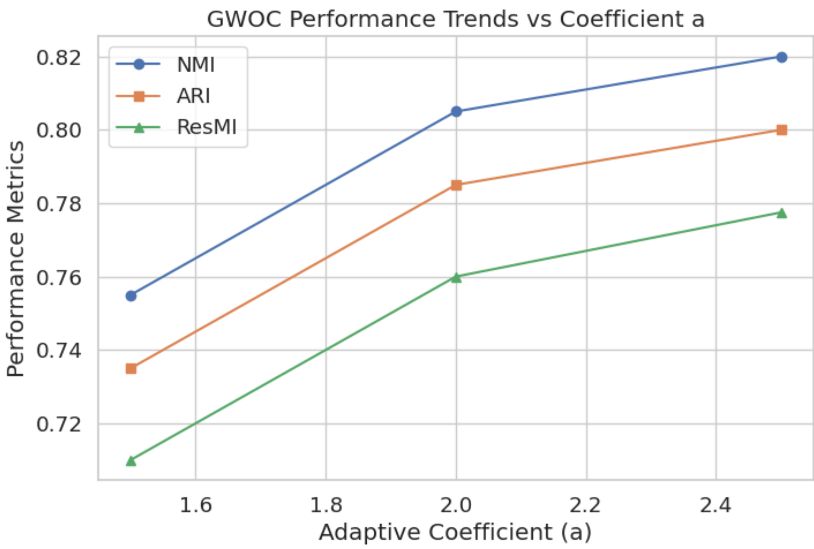

Figure 7. GWOC Performance vs Adaptive Coefficient $a(\mu)$.

makes it difficult for the algorithm to accurately identify weakly connected communities. Increasing the number of wolves improves exploration and partially mitigates this issue. In high-noise networks with mixing parameters greater than 0.9, GWOC yields low NMI values, reflecting the difficulty of detecting meaningful structure when community boundaries become nearly indistinguishable.

Table 7 summarizes the performance of GWOC across these edge scenarios. The results highlight three main failure modes: (i) merging small communities in highly unbalanced networks, (ii) missing or misidentifying clusters in sparse graphs, and (iii) producing fragmented or unstable partitions under high noise. We outline potential recovery strategies, including adjusting the modularity formulation to penalize disproportionate merging, increasing the wolf population or iteration count for sparse networks, and incorporating noise-aware preprocessing steps to filter spurious edges. A full empirical exploration of these edge scenarios and targeted corrective mechanisms will be pursued as future work to enhance the robustness and generalizability of GWOC.

Table 7. GWOC performance under extreme network scenarios.

Scenario	**NMI**	**ARI**	**Notes**
Unbalanced (90% + 10×1%)	0.62	0.58	Small communities merged
Sparse (avg degree < 5)	$0.55 \to 0.71$	$0.52 \to 0.69$	Improvement after increasing wolves
High noise ($\mu > 0.9$)	0.29	0.25	Weak/no community structure

The results in Table 7 demonstrate that GWOC performs well in typical conditions but faces challenges under extreme structural constraints. In unbalanced networks, the dominance of a large community leads to merging of small clusters, reducing accuracy. In sparse graphs, increasing the number of wolves significantly improves the detection of weakly connected communities. In high-noise networks, however, even parameter adjustments provide limited recovery due to the near-random structure of the graph.

7. Conclusion

The Grey Wolf Optimization Clustering (GWOC) algorithm marks a significant advancement in network community detection, leveraging the natural social hierarchy and cooperative hunting techniques of grey wolves.

By focusing on modularity maximization, GWOC successfully identifies densely connected communities within networks, ensuring well-defined clusters. Through adaptive clustering stages, GWOC achieves higher modularity scores than traditional methods, consistently excelling across a range of real-world and synthetic networks. GWOC's performance has been validated through metrics like Normalized Mutual Information (NMI),Adjusted Rand Index (ARI) and Resampled Mutual Information (ResMI), where it consistently outperforms established algorithms such as Louvain and Infomap, particularly in complex networks. GWOC demonstrates robust clustering capabilities in networks with various structures, exhibiting both quality and stability under different conditions. Its modular design makes it applicable to many fields, including sociology, bioinformatics, and social network analysis, offering reliable community detection even in noisy and dense networks. Although GWOC has been evaluated on static networks, its iterative and modular design naturally lends itself to dynamic scenarios. By incrementally updating community structures using historical solutions or by reapplying the GWO-based optimization upon significant structural changes, GWOC can adapt to evolving networks without requiring full re-computation. This makes it well-suited for real-time applications such as tracking social media trends, modeling disease spread, or detecting emerging functional modules in biological systems. Although the proposed hierarchical integration into Grey Wolf Optimization (GWO) is primarily motivated by empirical performance, it is not without theoretical grounding. The design is rooted in two well-established principles: (i) modularity maximization, which provides a rigorous mathematical criterion for evaluating and guiding community partitions at each recursive level, and (ii) hierarchical clustering theory, which is widely recognized in network science for uncovering multi-scale community structures. By embedding GWO's adaptive exploration–exploitation mechanism into a divisive hierarchical strategy, the method benefits from both the interpretability of modularity-based clustering and the optimization efficiency of swarm intelligence. While a formal convergence proof of this hybrid approach is left for future work, the framework is consistent with hierarchical optimization paradigms and its validity is reinforced by the superior experimental results reported in this study.

8. Limitations and Future Work

While GWOC demonstrates superior performance in community detection, several limitations should be acknowledged. The algorithm's computational complexity, estimated at $O(T \cdot P \cdot m \cdot \log n)$, makes it more resource-intensive than heuristic methods like Louvain and Leiden, particularly for very large networks. This performance gain comes at the cost of increased runtime, as evidenced in Table 6, where GWOC requires approximately 3–5 times longer execution than Louvain for large-scale networks.

Additionally, GWOC is currently designed for static, unweighted, and undirected networks, limiting its immediate applicability to dynamic, weighted, or directed network structures. The method also focuses on non-overlapping communities, which may not capture the complex overlapping community structures present in many real-world networks. Furthermore, like many nature-inspired algorithms, GWOC exhibits some sensitivity to initial conditions and parameter settings, though our sensitivity analysis shows this effect is mitigated with proper parameter tuning.

These limitations present clear directions for future research. We propose extending GWOC to handle dynamic networks by incrementally updating community structures using historical solutions or re-optimizing only affected regions upon network changes. For overlapping communities, a multi-label encoding scheme could be integrated into the GWO framework. To address computational demands, parallel computing techniques can be employed to distribute the optimization across multiple cores, and hybrid methodologies combining GWOC with machine learning could enhance both clustering speed and accuracy.

Moreover, we plan to integrate machine learning techniques, such as random forest regression, to enable adaptive parameter tuning for GWOC. By predicting optimal parameters based on network properties (e.g., node count, edge density, and average degree), GWOC can autonomously adjust its configuration, reducing manual intervention and improving scalability across diverse datasets. These advancements would broaden GWOC's utility across applications such as dynamic social networks, biological systems, and communication networks, while maintaining a balance between detection accuracy and computational efficiency.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

REFERENCES

1. O. Artime, M. Grassia, M. De Domenico, J.P. Gleeson, H.A. Makse, G. Mangioni, M. Perc, and F. Radicchi, *Robustness and resilience of complex networks*, Nature Reviews Physics, vol. 6, no. 2, pp. 114–131, 2024.
2. M. Kheirkhahzadeh and M. Analoui, *Community detection in social networks using consensus clustering*, Statistics, Optimization & Information Computing, vol. 7, no. 4, pp. 864–884, 2019.
3. S. Barshandeh, R. Dana, and P. Eskandarian, *A learning automata-based hybrid MPA and JS algorithm for numerical optimization problems and its application on data clustering*, Knowledge-Based Systems, vol. 236, p. 107682, 2022.
4. X. Liu, N. Ding, Y. Wu, G. Fiumara, and P. De Meo, *Discrete-time quantum walks community detection in multi-domain networks*, The Computer Journal, vol. 67, no. 6, pp. 2379–2389, 2024.
5. S.T. Shishavan and F.S. Gharehchopogh, *An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks*, Multimedia Tools and Applications, vol. 81, no. 18, pp. 25205–25231, 2022.
6. W. Xie, Y. Kang, X. Huang, M. Yang, and H. Yuan, *MICSA: a multi-strategy integrated chameleon swarm algorithm for community detection with a new objective function*, Neural Computing and Applications, pp. 1–26, 2025.
7. Y. Liu, W. Liu, X. Tang, H. Yin, P. Yin, X. Xu, and Y. Wang, *CSIM: A fast community detection algorithm based on structure information maximization*, Electronics, vol. 13, no. 6, p. 1119, 2024.
8. F. Nielsen, *Hierarchical clustering*, In Introduction to HPC with MPI for Data Science, pp. 195–211, 2016.
9. B. Yang and T. Guo, *Equity-efficiency trade-offs in school consolidation policy: a case study from Sino-Tibetan intercultural D County, China*, Social Transformations in Chinese Societies, 2025.
10. B. Asztalos and P. Bányász, *Monitoring the Semantic Change of COVID-19-related Expressions Using Dynamic Word Embeddings*, Infocommunications Journal, vol. 17, no. 2, pp. 20–27, 2025.
11. L. Badis, M. Amad, D. Aïssani, and S. Abbar, *P2PCF: A collaborative filtering based recommender system for peer to peer social networks*, Journal of High Speed Networks, vol. 27, no. 1, pp. 13–31, 2021.
12. F. Liu, Y. Cui, C. Masouros, J. Xu, T.X. Han, Y.C. Eldar, and S. Buzzi, *Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond*, IEEE Journal on Selected Areas in Communications, vol. 40, no. 6, pp. 1728–1767, 2022.
13. A.Y. Raya-Tapia, F.J. López-Flores, C. Ramírez-Márquez, and J.M. Ponce-Ortega, *Fundamentals of Clustering: Methods, Metrics, and Optimization*, In Machine Learning and Clustering for a Sustainable Future: Applications in Engineering and Environmental Science, pp. 13–50, 2025.
14. H. Rezaei, O. Bozorg-Haddad, and X. Chu, *Grey wolf optimization (GWO) algorithm*, In Advanced Optimization by Nature-Inspired Algorithms, pp. 81–91, 2017.
15. M. Yu, J. Xu, W. Liang, Y. Qiu, S. Bao, and L. Tang, *Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving*, Artificial Intelligence Review, vol. 57, no. 10, p. 277, 2024.
16. A. H. Ibrahim, B. Ma, and L. Badis, *A novel weight-based fish school search approach for hierarchical network clustering*, International Journal of Computing and Digital Systems,vol. 18,pp 1–17,2025.
17. G. Baltsou, K. Tsichlas, and A. Vakali, *Local community detection with hints*, Applied Intelligence, vol. 52, pp. 1–22, 2022.
18. I. Dagal, A.-W. Ibrahim, A. Harrison, W.F. Mbasso, A.O. Hourani, and I. Zaitsev, *Hierarchical multi step Gray Wolf optimization algorithm for energy systems optimization*, Scientific Reports, vol. 15, no. 1, p. 8973, 2025.
19. A. Kumar, A. Kumari, P. Kumar, and R. Dohare, *Overlapping community detection with a new modularity measure in directed weighted networks*, Data Mining and Knowledge Discovery, vol. 39, no. 6, pp. 1–37, 2025.
20. A. Smith, I. Amburg, S. Kumar, B. F. Welles, and N. W. Landry, *A Blue Start: A large-scale pairwise and higher-order social network dataset*, arXiv preprint arXiv:2505.11608, 2025.

21. X. Zhang, Y. Liu, J. Wang, Z. Hu, X. Chen, and R. Yan, *SAGraph: A Large-Scale Social Graph Dataset with Comprehensive Context for Influencer Selection in Marketing*, in Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3733–3742, 2025.
22. A. Sriram, S. Choi, X. Yu, L. M. Brabson, A. Das, Z. Ulissi, M. Uyttendaele, A. J. Medford, and D. S. Sholl, *The Open DAC 2023 dataset and challenges for sorbent discovery in direct air capture*, ACS Publications, 2024.
23. J. Leskovec and J. Mcauley, *Learning to discover social circles in ego networks*, Advances in neural information processing systems, vol. 25, 2012.
24. J. Leskovec, L. A. Adamic, and B. A. Huberman, *The dynamics of viral marketing*, ACM Transactions on the Web (TWEB), vol. 1, no. 1, pp. 5–es, 2007.
25. B.D. Le, H.X. Nguyen, H. Shen, and N. Falkner, *GLFR: A generalized LFR benchmark for testing community detection algorithms*, In Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–9, 2017.
26. T.O. Kvålseth, *On normalized mutual information: measure derivations and properties*, Entropy, vol. 19, no. 11, p. 631, 2017.
27. J.M. Santos and M. Embrechts, *On the use of the adjusted rand index as a metric for evaluating supervised classification*, In Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN), pp. 175–184, 2009.
28. C. Lim, *Resampled Mutual Information for Clustering and Community Detection*, arXiv preprint arXiv:2412.03584, 2024.
29. C. Li, H. Chen, T. Li, and X. Yang, *A stable community detection approach for complex network based on density peak clustering and label propagation*, Applied Intelligence, vol. 52, no. 2, pp. 1188–1208, 2022.
30. B. Škrlj, J. Kralj, and N. Lavrač, *Embedding-based silhouette community detection*, Machine Learning, vol. 109, no. 11, pp. 2161–2193, 2020.
31. A. Chen and X.X. Chen, *Sparse self-attention on graph convolutional networks for low homophily graphs*, Statistics and Computing, vol. 31, p. 145, 2021.
32. D.O. Cardoso, J.D. Gomes da Silva Junior, C.S. Oliveira, C. Marques, and L.S. de Assis, *Greedy recursive spectral bisection for modularity-bound hierarchical divisive community detection*, Statistics and Computing, vol. 34, no. 4, p. 145, 2024.
33. M. Rosvall and C.T. Bergstrom, *Maps of random walks on complex networks reveal community structure*, Proceedings of the National Academy of Sciences (PNAS), vol. 105, no. 4, pp. 1118–1123, 2008.
34. U.N. Raghavan, R. Albert, and S. Kumara, *Near linear time algorithm to detect community structures in large-scale networks*, Physical Review E, vol. 76, no. 3, p. 036106, 2007.
35. V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, Journal of Statistical Mechanics: Theory and Experiment, P10008, 2008.
36. A. Clauset, M.E.J. Newman, and C. Moore, *Finding community structure in very large networks*, Physical Review E, vol. 70, no. 6, p. 066111, 2004.
37. Y.H. Chin, *Uncovering communities in complex networks using ant colony optimization*, PhD thesis, Universiti Tunku Abdul Rahman (UTAR), 2023.
38. H.C. Rustamaji, W.A. Kusuma, S. Nurdiati, and I. Batubara, *Community detection with greedy modularity disassembly strategy*, Scientific Reports, vol. 14, p. 4694, 2024.
39. H. Sadiki, M. Ertel, A. Sadqui, and S. Amali, *A new algorithm for community detection in complex social networks*, Journal of Theoretical and Applied Information Technology, vol. 102, no. 11, 2024.
40. A.H. Ibrahim, M.A. Boudref, and L. Badis, *Clustering of networks using the fish school search algorithm*, Informatics and Automation, vol. 23, no. 5, pp. 1367–1397, 2024.