Future Technology
Open Access Journal
ISSN 2832-0379

November 2026| Volume 05 | Issue 04 | Pages 199-210

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.5.4.16

Future Publishing LLC

Article

An intelligent distance-driven community detection framework for complex networks using the longest distance node head technique (LDNHT)

Srinivas Amedapu*, Leela Velusamy R

Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

A R T I C L E I N F O

Article history:
Received 22 January 2026
Received in revised form
15 August 2026
Accepted 10 September 2026

Keywords:
Community detection, Complex networks, LDNHT, Geodesic distance, Graph partitioning

*Corresponding author
Email address:
srinivasreddyamedapu@yahoo.com

DOI: 10.55670/fpll.futech.5.4.16

A B S T R A C T

Community detection in complex networks needs methods that are not only structurally good, but also computationally efficient, stable, and interpretable. This study introduces a centroid-based community detection framework called Longest Distance Node Head Technique (LDNHT), where the first centroid is chosen randomly, and the others are determined by maximizing the total shortest-path distances from existing centroids. The method was tested on Facebook, Power-Grid, Wiki-Vote, and Gnutella networks in comparison with Random Node Head Technique (RNHT), Highest Degree Node Head Technique (HDNHT), classical max-min distance seeding, Louvain, and Fast-Greedy community detection. The stochastic methods were tested on more than 35 seeded runs, and the deterministic baselines were performed under the same experimental protocol. The modularity, External Connection Ratio (ECR), conductance, coverage, and runtime were used to assess partition quality. On Power-Grid and Wiki-Vote, its performance was significantly better than RNHT, HDNHT, and MAXMIN in terms of ECR and coverage, while Louvain and Fast-Greedy generally achieved higher modularity. Sensitivity analysis showed that the number of repeated initializations had a small impact, with the dependence on the first random centroid decreasing significantly with repeated initializations, and the benefit diminishing after about 20 internal iterations. The empirical scalability experiments on sparse networks showed approximately linear behavior, with a log-log slope of 0.911. Overall, LDNHT offers a comprehensible cumulative geodesic-dispersion strategy whose effectiveness is dependent on network structure and parameter configuration.

1. Introduction

The rapid growth of digital environments, web applications, and cyberspace infrastructure has produced large-scale interconnected datasets that are naturally modeled as complex networks. These networks appear in many contexts, such as social interactions, communication systems, energy grids, and information infrastructures. In these systems, the community detection problem of finding sets of nodes that are highly connected within and sparsely connected to others outside has been a critical tool for analyzing structural organization and functional behavior [1]. Applications include recommendation systems, anomaly and fraud detection, network security, and information diffusion, as well as emerging areas such as smart grids, intelligent communication systems, and sustainable infrastructure management. Despite extensive research, existing community detection methods have major limitations when applied to large, heterogeneous networks. Modularity-based methods, such as the Louvain and Leiden algorithms, are common because they are computationally cheap and scalable, but they suffer from resolution limits and poor localization because they are sensitive to local optima, resulting in unstable or incorrect partitions [2]. Similarly, label propagation methods have linear complexity and are often random, which can lead to dissimilar community structures when replicated. Centroid-based methods, where nodes receive a representative head based on their distance, are easy and intuitive to apply; however, centroid crowding and spatial imbalance problems are likely to arise, particularly in large-scale and structurally complex networks [3]. To address these shortcomings, recent research has explored hybrid and structure-sensitive methods to improve community detection. Approaches based on central node definition and local structure optimization aim to enhance seed selection and stabilize clustering performance [4]. At the same time, domain-specific studies in energy networks have emphasized the importance of identifying self-sufficient and resilient substructures, such as microgrid partitions, and ensuring system stability and flexibility [5]. Another possible direction is distance-based analysis, which uses shortest-path distance to label structural proximity in the network. Using distance information to select centroids can enforce spatial separation between community centers, leading to interpretable and structurally clear partitions [6].

These limitations inspired this paper, and we propose a smart distance-based community detection algorithm, the Longest Distance Node Head Technique (LDNHT). It is a centroid-selection-based, topology-conscious approach to intelligent decision-making. Unlike random or degree-based algorithms, LDNHT repeatedly selects community heads by optimizing cumulative shortest-path distances, which provides optimal spatial dispersal across the network. This centroid selection is justified by a distance-driven process that induces global structural awareness to reduce overlap and centroid bias. Therefore, LDNHT generates more balanced, well-separated, and interpretable communities, with better cohesion and separation achieved without computationally expensive global optimization. The primary objectives of this work are:

- To develop a reproducible centroid-based community detection framework using cumulative geodesic distance for sequential centroid selection and to distinguish its mechanism from conventional random, degree-based, and MAXMIN/farthest-first initialization approaches.
- To evaluate the performance of LDNHT against RNHT, HDNHT, MAXMIN, Louvain, and Fast-Greedy using modularity, External Connection Ratio (ECR), conductance, coverage, runtime, and ground-truth measures such as NMI and ARI on LFR benchmark networks.
- To examine the robustness and scalability of LDNHT through sensitivity analyses of the community count, number of internal trials, and random first-centroid initialization, together with empirical evaluation across increasing network sizes and edge-density levels.

2. Related work

Several methodological approaches have been explored for community detection, such as graph partitioning, matrix factorization, modularity optimization, propagation-based methods, representation learning, and seed-based clustering. One way to do this is to use nonnegative matrix factorization (NMF) to find latent community structures in low-rank representations of adjacency relationships [7]. More broadly, community detection has gradually shifted from traditional graph-theoretical and clustering methods to hybrid, probabilistic, and learning-based methods [8]. Other approaches show that considering multiple structural levels can improve partition quality by integrating local neighborhood information with global network structure [9]. Modularity-based approaches remain among the most popular methods because they provide a clear structural goal: identifying groups with greater internal connectivity than external connectivity. Efficient local moving strategies have significantly improved the scalability of modularity optimization [10,11], and later, researchers added node attributes alongside topological information [12]. While these methods work well on many large networks, modularity optimization can be affected by resolution effects and local optima. Maximizing modularity is therefore a key criterion for community detection, but not the best one for all community structures. Propagation-based approaches offer a more computationally efficient alternative, in which community labels are passed around among neighborhoods in the local network.

To improve convergence and reproducibility, researchers developed stabilized variants of label-propagation methods, including LabelRank and semi-synchronous label-propagation methods [13,14]. However, propagation-based algorithms can still be sensitive to the ordering of node updates and tie-breaking, as well as to the initial state. This variability is especially important when results must be reproducible across repeated runs. Representation-learning approaches learn node representations before community assignment. Low-dimensional node embeddings are generated using biased random walks and then clustered [15]. Graph convolutional networks have then evolved representation learning by incorporating neighborhood information to learn node representations [16]. While these methods offer flexible representations, they tend to be more model-dependent and computationally intensive than lightweight graph-theoretic methods, and they do not offer the same level of structural interpretability.

Seed- and centrality-based approaches identify structurally important nodes that can act as starting points for community formation. Research on influential-node identification has demonstrated that highly central or influential nodes can play important roles in network structure and diffusion [17,18]. However, centrality alone does not ensure that the selected seeds are geographically or topologically dispersed across the graph. Multiple high-centrality nodes may occur within the same structural region, resulting in concentrated community representatives. Distance information provides a complementary mechanism because shortest-path relationships can explicitly represent topological separation between candidate centroids. Community detection has also been extended to dynamic and structurally complex networks. Dynamic-community methods consider changes in group structure over time [19], while the Lancichinetti-Fortunato-Radicchi (LFR) benchmark provides controlled networks with planted communities for evaluating recovery performance under varying structural difficulty [20]. Reproducible network-analysis libraries have further supported systematic experimentation across social, communication, and infrastructure graphs [21]. The field has additionally expanded toward deep-learning and parallel metaheuristic approaches [22], geometry-based methods such as discrete Ricci-flow analysis [23], probabilistic models for multiplex and multirelational networks [24], and formulations intended for directed, weighted, or overlapping community structures [25]. Theoretical studies on clustering dynamics and degree-based structure have also helped explain regularities in complex networks [26]; scalable methods have been proposed to overcome some drawbacks of traditional modularity optimization methods [27].

Although methods vary, representative-node selection remains an important problem in centroid-based community detection. Random initialization may cause run-to-run variation, while degree- or influence-based initialization may have several centroids in structurally similar areas. Classical distance-dispersion methods select a new representative sequentially based on its minimum distance from the cluster of selected representatives, such as Gonzalez's farthest-first method for metric-center clustering [28-32]. The idea of MAXMIN is the same as in the max-min principle: maximize the distance to the nearest existing centroid. Instead of this minimum-distance criterion, in LDNHT each candidate is assessed by the sum of the distances to the shortest paths of its centroids to all previously selected centroids. Therefore, the claimed methodological novelty does not lie in the use of graph distance itself, but in the aggregation of geodesic information across the complete selected-centroid set. The framework consequently examines whether cumulative distance dispersion provides a reproducible and interpretable alternative to conventional farthest-first initialization for graph community partitioning.

3. Research methodology

This section presents the methodological framework used to develop and evaluate the proposed Longest Distance Node Head Technique (LDNHT). LDNHT is formulated for simple, undirected, and unweighted graphs and uses shortest-path information to select spatially dispersed centroids before assigning each node to its nearest selected centroid. The framework was evaluated using four real-world benchmark networks and a controlled LFR synthetic benchmark with planted community labels. Comparative evaluation was conducted against the Random Node Head Technique (RNHT), Highest Degree Node Head Technique (HDNHT), classical MAXMIN distance-based initialization, Louvain modularity optimization, and Fast-Greedy community detection. Figure 1 provides a small illustrative undirected and unweighted network used only to explain the centroid-selection and node-assignment mechanism of LDNHT. The principal experimental evaluation, however, is based on the benchmark datasets and repeated computational experiments described in the following subsections.

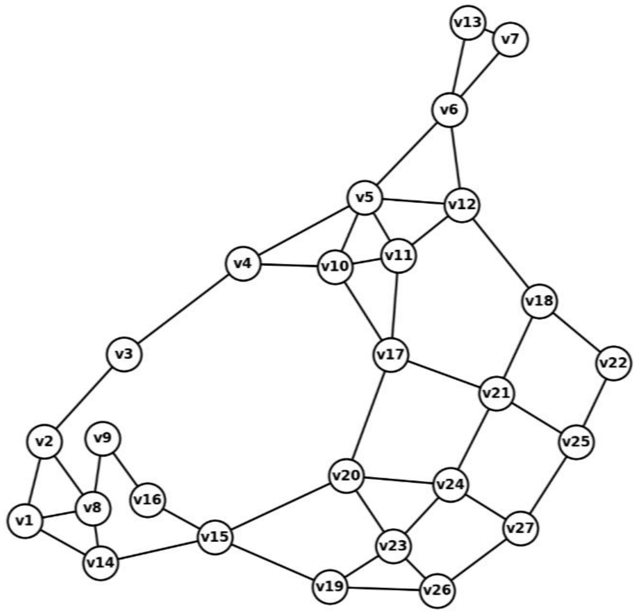

Figure 1. Sample network

To illustrate the ideas of the work, we assume a representative undirected, unweighted network (Graph-1). The framework is contrasted with three reference techniques: Random Node Head Technique (RNHT), Highest Degree Node Head Technique (HDNHT), and the Louvain modularity optimization method. Although these baselines serve as references, this section focuses on formalizing and justifying the proposed LDNHT approach.

3.1 Dataset description

Four real-world benchmark networks with distinct structural characteristics were used to evaluate the proposed framework: Facebook [28], Power-Grid [29], Wiki-Vote [30], and Gnutella [31]. Facebook represents a social interaction network, Power-Grid models the Western U.S. electrical transmission infrastructure, Wiki-Vote represents voting interactions among Wikipedia users, and Gnutella represents a peer-to-peer communication network. The diversity of these datasets enables evaluation across social, infrastructure, collaborative, and decentralized communication structures. All datasets were processed using a common graph-preparation procedure. Self-loops were removed, and the graphs were represented as simple, unweighted networks. Facebook and Power-Grid are natively undirected. Wiki-Vote and Gnutella were originally directed and were converted to undirected graphs for compatibility with the shortest-path-based centroid framework. After conversion, the largest connected component was retained so that shortest-path distances remained finite between all analyzed nodes. Consequently, Wiki-Vote was reduced from 7,115 source nodes to 7,066 analyzed nodes, while Gnutella was reduced from 6,301 to 6,299 nodes. The structural characteristics of the final analyzed networks are presented in Table 1.

For the primary real-network experiments, the number of communities required by the centroid-based methods was calibrated using 35 independent Louvain executions for each dataset. The median detected community count was used as the common primary k, resulting in $k = 16$, $k = 39$, $k = 7$, and $k = 21$ for Facebook, Power-Grid, Wiki-Vote, and Gnutella, respectively. The corresponding Louvain community-count ranges were 14–17, 37–43, 5–18, and 18–25. This calibration provides a consistent operating point for RNHT, HDNHT, MAXMIN, and LDNHT. Because Louvain-derived values may introduce dependence on the calibration procedure, we explicitly examined this issue through a separate sensitivity analysis. In addition, the LFR ground-truth experiments used the known planted community count for centroid-based methods rather than a Louvain-derived value. Louvain and Fast-Greedy were allowed to determine their own community counts during comparative evaluation.

Table 1. Dataset description

Dataset	Nodes	Edges	Average Degree	Density	Clustering Coefficient	Primary (k)
Facebook	4,039	88,234	43.6910	0.010820	0.6055	16
Power-Grid	4,941	6,594	2.6691	0.000540	0.0801	39
Wiki-Vote	7,066	100,736	28.5129	0.004036	0.1419	7
Gnutella	6,299	20,776	6.5966	0.001047	0.0109	21

Louvain is used only as a calibration mechanism to obtain a consistent value of k across methods and is not a dependency of the proposed framework. This makes the evaluation unbiased, allowing a direct and fair comparison between centroid-based (RNHT, HDNHT, LDNHT) and modularity-based methods.

3.2 Evaluation metrics

A multi-metric evaluation framework was used because no single measure fully characterizes community quality. The real-network experiments were evaluated using modularity, External Connection Ratio (ECR), mean conductance, coverage, and runtime. In addition, Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) were used for the LFR synthetic benchmark, where planted ground-truth communities were available.

Modularity: Modularity measures whether edges occur within detected communities more frequently than expected under a degree-preserving null model. For an undirected network, modularity is expressed as:

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

where A_{ij} is the adjacency-matrix element between nodes i and j, k_i and k_j denote their degrees, m is the number of edges, and $\delta(c_i, c_j) = 1$ when both nodes belong to the same community and 0 otherwise. Higher modularity indicates stronger separation relative to the null model.

External Connection Ratio (ECR). The measure previously referred to as "fitness" is defined more precisely in this study as the External Connection Ratio. For node v,

$$ECR(v) = \frac{e_{\text{out}}(v)}{\deg(v)} \quad (2)$$

where $e_{\text{out}}(v)$ is the number of edges connecting v to nodes assigned to other communities and $\deg(v)$ is its total degree. The network-level ECR is calculated as:

$$ECR = \frac{1}{|V|}\sum_{v\in V} ECR(v) \quad (3)$$

Lower ECR values indicate that fewer node connections cross community boundaries and therefore represent stronger local community cohesion.

Conductance. Conductance measures how well a community is separated from the rest of the network. For community C,

$$\phi(C) = \frac{\text{cut}(C, V\setminus C)}{\min\{\text{vol}(C), \text{vol}(V\setminus C)\}} \quad (4)$$

where $\text{cut}(C, V \setminus C)$ is the number of edges crossing the community boundary and $\text{vol}(C)$ is the sum of node degrees within C. Mean conductance was obtained by averaging the conductance values across detected communities. Lower values indicate better separation.

Coverage: Coverage represents the proportion of all graph edges that occur within detected communities:

$$\text{Coverage} = \frac{m_{\text{intra}}}{m} \quad (5)$$

where m_{intra} denotes the number of intra-community edges and m is the total number of graph edges. Higher coverage indicates that a larger proportion of network connections are retained within communities.

Runtime: Runtime was measured as the wall-clock execution time required to complete each algorithmic run. It was reported in seconds and used to assess computational efficiency under the same experimental environment.

For the LFR benchmark experiments, Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) were used to compare the detected assignments with the known planted communities. Both metrics account for agreement between two partitions, while ARI additionally corrects pairwise agreement for chance. Values closer to 1 indicate greater correspondence with the planted community structure. NMI and ARI were not applied to the four real-world networks because the experimental data did not provide a common authoritative ground-truth partition for all four datasets.

3.3 Random node head technique (RNHT)

Five baseline methods were used to provide comparisons with the proposed LDNHT framework: Random Node Head Technique (RNHT), Highest Degree Node Head Technique (HDNHT), classical MAXMIN distance-based initialization, Louvain community detection, and Fast-Greedy modularity optimization. The centroid-based baselines used the same predefined values as LDNHT in the real-network experiments so that differences could be attributed primarily to the centroid-selection strategy.

RNHT: RNHT represents a random centroid-selection baseline. For each internal trial, distinct nodes are selected randomly as community centroids using a seeded random-number generator. Every remaining node is assigned to the centroid having the minimum unweighted shortest-path distance. When a node is equally distant from multiple centroids, the smaller centroid identifier is used as the deterministic tie-break. Because RNHT is stochastic, internal trials were performed for each outer experimental run. The partition with the highest modularity was retained. When two candidate partitions produced equal modularity, the partition with the lower ECR was preferred.

HDNHT: HDNHT represents a degree-based centroid-selection baseline. The nodes with the highest degree are selected as centroids, with smaller node identifiers used to resolve equal-degree ties. Each remaining node is assigned to its nearest centroid by unweighted shortest-path distance, using the smaller centroid identifier to break equal-distance ties. Because centroid selection is deterministic for a fixed graph, HDNHT produces a single partition rather than requiring repeated internal initialization.

MAXMIN: MAXMIN was included as the closest conventional distance-dispersion comparator to LDNHT. The first centroid is selected randomly. Each subsequent centroid is chosen according to:

$$c_{j+1} = \arg\max_{v\in V\setminus C_j}\min_{c\in C_j} d(v, c) \quad (6)$$

where C_j denotes the set of already selected centroids and $d(v, c)$ is the unweighted shortest-path distance. Thus, MAXMIN selects the node whose nearest existing centroid is as far away as possible. Equal maximum-distance values are resolved first by higher node degree and then by smaller node identifier. Node assignment follows the same nearest-centroid rule used by the other centroid methods. MAXMIN was executed for $T = 30$ internal trials, and the partition with the highest modularity was retained, with lower ECR used as the secondary selection criterion.

Louvain: Louvain was included as a modularity-optimization baseline and was implemented using the NetworkX Louvain community-detection routine. The experiments used an unweighted graph representation, resolution parameter, and threshold. Louvain determines the number of communities automatically rather than using the externally supplied. For stochastic evaluation, the same predefined outer seeds were used across the 35 independent runs.

Fast-Greedy: Fast-Greedy was included as an additional established modularity-based baseline. It iteratively merges communities according to the modularity gain until no further beneficial merge is available. The NetworkX greedy modularity community routine was used with an unweighted graph and a resolution parameter. Fast-Greedy determines its own number of communities and is deterministic for a fixed input graph under the implementation used in this study.

3.4 Proposed longest distance node head technique (LDNHT)

The Longest Distance Node Head Technique (LDNHT) is a centroid-based community detection method designed around cumulative geodesic dispersion. Let $G = (V, E)$ denote a connected, simple, undirected, and unweighted graph, let k denote the required number of communities, and let T denote the number of internal initialization trials. LDNHT first selects one centroid randomly and then sequentially selects additional centroids according to their cumulative shortest-path separation from all previously selected centroids. For an existing centroid set C_j, each unselected candidate node v is assigned the cumulative geodesic-dispersion score

$$S_j(v) = \sum_{c \in C_j} d(v, c) \tag{7}$$

where $d(v, c)$ denotes the unweighted shortest-path distance between candidate node v and centroid c. The next centroid is selected as

$$c_{j+1} = \arg\max_{v \in V \setminus C_j} S_j(v) = \arg\max_{v \in V \setminus C_j} \sum_{c \in C_j} d(v, c) \tag{8}$$

When multiple candidates have the same maximum cumulative score, the candidate with the higher degree is selected. If the degree is also tied, the node with the smaller identifier is selected. This deterministic two-stage tie-breaking rule ensures reproducibility once the first centroid has been established.

The key distinction from classical MAXMIN initialization lies in the aggregation of distance information. MAXMIN selects the next centroid according to the maximum of the minimum distance to the existing centroid set, whereas LDNHT maximizes the sum of distances to all previously selected centroids. Thus, the methodological contribution is the cumulative geodesic-dispersion criterion rather than the general use of graph distance. Shortest-path distances are computed using Breadth-First Search (BFS), which is appropriate because all analyzed networks are represented as unweighted graphs. After the first centroid is selected, its BFS distance map is computed once. Whenever a new centroid is added, a single additional BFS traversal is performed, and the corresponding distances are added incrementally to the cumulative score $S_j(v)$. Previously computed BFS distance maps are therefore reused rather than recomputed from the beginning. After k centroids have been obtained, each graph node is assigned to the centroid with the minimum shortest-path distance:

$$a(v) = \arg\min_{c \in C_k} d(v, c) \tag{9}$$

If a node is equally distant from more than one centroid, the centroid with the smaller identifier is selected.

Because the first centroid is randomized, the complete centroid-selection and assignment process is repeated for $T = 30$ internal trials in the primary experiments. Each trial produces a candidate partition P_t. The retained partition is the one with the largest modularity:

$$P^* = \arg\max_{P_t, t=1,\ldots,T} Q(P_t) \tag{10}$$

If two candidate partitions have equal modularity, the partition with the lower External Connection Ratio is retained. If an exact tie remains, deterministic centroid ordering provides the final tie-break. ECR is therefore used only as a secondary restart-selection criterion rather than as the primary optimization objective.

Algorithm 1. Longest distance node head technique

Input:
Connected unweighted graph G = (V,E)
Number of communities k
Number of internal trials T
Random seed s

Output:
Selected partition P*

1. Set $Q*$ = -infinity
2. Set $ECR*$ = +infinity
3. Set $P*$ = null

Initialize pseudo-random generator RNG using seed s
4. for $t = 1,...,T$ do
5. Select first centroid randomly from the sorted node set using RNG.
6. $C = \{c1\}$
7. Run BFS from $c1$
8. For each v not in C:
$S(v) = d(v, c1)$
9. while $|C| < k$ do
10. Select $v*$ maximizing $S(v)$
11. If cumulative scores are tied:
select the candidate with higher degree
12. If degree is also tied:
select the candidate with smaller node ID
13. $C = C \ union\ \{v*\}$
14. Run one BFS from $v*$
15. For each unselected node v:
$S(v) = S(v) + d(v, v*)$
16. end while
17. For each node v in V:
assign v to the centroid minimizing $d(v, c)$
18. If nearest-centroid distances are tied:
assign v to the centroid with smaller node ID
19. Construct candidate partition Pt
20. Compute modularity $Q(Pt)$
21. Compute $ECR(Pt)$
22. If $Q(Pt) > Q*$:
$P* = Pt$
$Q* = Q(Pt)$
$ECR* = ECR(Pt)$
23. Else if $Q(Pt) = Q*$ and $ECR(Pt) < ECR*$:
$P* = Pt$
$ECR* = ECR(Pt)$
24. If an exact tie remains:
use deterministic centroid ordering
25. end for
26. Return $P*$

3.5 Computational complexity

For an unweighted graph $G = (V, E)$, LDNHT computes geodesic distances using Breadth-First Search (BFS). A single BFS traversal requires $O(|V| + |E|)$ time. During one internal LDNHT trial, the first centroid requires one BFS traversal, and each subsequently selected centroid requires one additional

BFS traversal. Therefore, selecting k centroids requires approximately k BFS traversals, giving $O(k(|V| + |E|))$ time for shortest-path computation within one trial. The cumulative geodesic scores of the candidate nodes are updated after each newly selected centroid. These updates require $O(k|V|)$ operations. After centroid selection, assigning every node to its nearest centroid also requires $O(k|V|)$ comparisons when the previously computed centroid-distance maps are reused. Consequently, the overall complexity of one internal trial is $O(k(|V| + |E|) + k|V|)$, which simplifies to $O(k(|V| + |E|))$.

Since the complete procedure is repeated for T internal trials, the overall runtime complexity becomes $O(Tk(|V| + |E|))$ for the implemented unweighted-graph setting. The implementation reuses shortest-path distance maps from previous BFS traversals rather than recomputing all distances at each centroid-selection step. If the distance maps for the k selected centroids are retained, their auxiliary storage requirement is approximately $O(k|V|)$, in addition to the $O(|V| + |E|)$ storage required for the graph itself. For sparse networks, where $|E| = O(|V|)$, and when k and T are treated as fixed experimental parameters, the expected scaling behavior approaches linear growth with the number of vertices. However, this theoretical property should not be interpreted as evidence of linear behavior for arbitrary dense graphs. In a dense network, $|E|$ may approach $O(|V|^2)$, which correspondingly increases the cost of each BFS traversal.

To examine practical scalability, a supplementary experiment was conducted using Barabási-Albert networks with $k = 20$ and $T = 10$. Network size was varied from 1,000 to 8,000 nodes while maintaining sparse connectivity. A separate edge-density experiment retained 3,000 nodes while increasing the Barabási-Albert attachment parameter from 2 to 12. These experiments were used to evaluate the empirical effects of increasing graph size and edge load without confounding the analysis through simultaneous variation in k or T.

3.6 Experimental protocol and statistical analysis

The primary real-network evaluation used 35 outer executions for each dataset-algorithm combination. For stochastic methods, these consisted of 35 seeded runs using a fixed predefined seed sequence to support reproducibility and matched comparison. RNHT, MAXMIN, and LDNHT additionally used 30 internal initialization trials within each outer run, whereas Louvain used the corresponding outer seed. HDNHT and Fast-Greedy were deterministic under the adopted implementation; therefore, their repeated executions produced identical structural partitions, although runtime was recorded across executions. For the centroid-based methods, the primary values of k were fixed at 16 for Facebook, 39 for Power-Grid, 7 for Wiki-Vote, and 21 for Gnutella. These values were obtained from the median number of communities identified across 35 Louvain calibration runs, as described in Section 3.1. Louvain and Fast-Greedy were allowed to determine their own community counts during comparative evaluation.

The structural performance of each method was summarized using the arithmetic mean, sample standard deviation, minimum, maximum, and 95% confidence interval across the repeated runs. Because the distributions could not be assumed to satisfy parametric assumptions and multiple algorithms were compared on matched graph instances, non-parametric statistical procedures were adopted. The Friedman test was first applied separately to each dataset and evaluation metric to determine whether overall differences existed among the six methods. Kendall's W was reported as an effect-size measure for the Friedman test. Where an overall difference was present, pairwise comparisons between LDNHT and each baseline were performed using two-sided Wilcoxon signed-rank tests. The family-wise error rate was controlled using the Holm correction, with statistical significance evaluated at $\alpha = 0.05$. For stochastic methods sharing the same outer seeds, the comparisons represented matched seeded outcomes. HDNHT and Fast-Greedy produced identical structural partitions across repeated executions; therefore, their zero structural variance reflects deterministic behavior rather than independent stochastic replication. Comparisons involving these methods were consequently interpreted as comparisons of the LDNHT run distribution against a fixed deterministic reference.

Runtime was recorded as wall-clock execution time for the community-detection routine. For centroid-based methods, this timing included centroid initialization, BFS shortest-path calculations, centroid selection, and node assignment. Post-hoc computation of evaluation metrics and CSV file writing were excluded from the algorithm runtime. All experiments were conducted on a Lenovo 21 KJ system equipped with an AMD Ryzen 57430 U processor with 6 physical cores and 12 logical processors and approximately 32 GB of RAM. The system ran 64-bit Microsoft Windows 11 Pro, version 10.0.26200. The implementation used Python 3.10.0, NetworkX 3.4.2, NumPy 2.2.6, SciPy 1.15.3, and scikit-learn 1.7.2. For reproducibility, the same graph-preprocessing rules, primary k values, algorithm parameters, and predefined seed sequence were retained throughout the primary experiment. No algorithm-specific post-hoc tuning was performed after observing the final comparative results.

4. Results

LDNHT was evaluated on the Facebook, Power-Grid, Wiki-Vote, and Gnutella networks against RNHT, HDNHT, MAXMIN, Louvain, and Fast-Greedy. The comparative assessment considered External Connection Ratio (ECR), modularity, conductance, coverage, and runtime. Statistical significance was evaluated using Friedman tests followed by Holm-adjusted Wilcoxon comparisons where appropriate. The following results summarize the principal structural and computational findings. Each of the methods has been reiterated 35 times on each of the datasets to make it robust and fair. The reason that led to this repetition-run protocol was that the four protocols have either stochastic or initialization-dependent items. The results provided, therefore, are the average 35-run performance that has the benefit of reducing the impact of random initialization, tie-breaking variation, and single outliers. Therefore, there is predictable algorithmic variability in the comparisons in this part, yet not the effects of a particular single run. Community count k was configured to the median community count of 35 Louvain runs per dataset in the centroid-based methods to provide a common calibration structure to be able to do comparative analysis.

Statistical differences among the six algorithms were evaluated using dataset-specific Friedman tests, followed by Holm-adjusted Wilcoxon signed-rank comparisons between LDNHT and each baseline. Descriptive results are reported as mean ± standard deviation across the 35 outer executions. For deterministic methods, a zero structural standard deviation indicates identical partitions across repeated executions rather than independent stochastic replication. Statistical significance was evaluated at $p < 0.05$.

4.1 External connection ratio (ECR)

External Connection Ratio (ECR) was used to assess the proportion of node connections extending outside the assigned community. Lower ECR values indicate stronger local cohesion. The corrected results show that LDNHT's relative performance depends on the network's structural characteristics rather than showing uniform superiority across all datasets (Table 2). On the Facebook network, LDNHT obtained a mean ECR of 0.099438. This was significantly lower than MAXMIN (0.132779; Holm-adjusted $p < 0.001$), but significantly higher than RNHT, HDNHT, Louvain, and Fast-Greedy. Therefore, the Facebook results provide evidence that cumulative geodesic dispersion improves ECR relative to classical MAXMIN initialization, but not relative to the remaining baselines. A stronger result was observed for Power-Grid. LDNHT achieved an ECR of 0.040626, compared with 0.046403 for RNHT, 0.055683 for HDNHT, and 0.050313 for MAXMIN. All three improvements were statistically significant after Holm correction ($p < 0.001$).

However, Louvain (0.023138) and Fast-Greedy (0.022545) retained significantly lower ECR values. The Wiki-Vote network also produced favorable centroid-baseline results. LDNHT achieved an ECR of 0.185444, significantly improving on RNHT (0.297806), HDNHT (0.323327), and MAXMIN (0.231051). The corresponding Holm-adjusted p-values were below 0.001. Nevertheless, Louvain (0.160078) and Fast-Greedy (0.110571) produced significantly lower ECR values than LDNHT. In contrast, Gnutella represents an important negative case. LDNHT produced the highest mean ECR (0.428068) among the six methods, compared with 0.414336 for RNHT, 0.423643 for HDNHT, 0.398213 for MAXMIN, 0.323322 for Louvain, and 0.287701 for Fast-Greedy. The pairwise differences between LDNHT and all five baselines were statistically significant after Holm correction. Overall, ECR results indicate that LDNHT is particularly effective relative to the other centroid-based strategies on Power-Grid and Wiki-Vote, while its advantage does not generalize to Facebook or Gnutella. These findings support a network-dependent interpretation of cumulative geodesic centroid selection rather than a claim of universal improvement.

4.2 Modularity performance

Modularity was used to evaluate the extent to which the detected partitions contained more within-community connections than expected under the corresponding null model. Higher modularity values indicate stronger global community structure. Table 3 presents the mean modularity obtained by the six methods across the four real-world networks.

On Facebook, Louvain achieved the highest mean modularity of 0.834943, followed by Fast-Greedy (0.777378) and MAXMIN (0.710121). LDNHT produced a mean modularity of 0.628204, which was significantly lower than RNHT, HDNHT, MAXMIN, Louvain, and Fast-Greedy after Holm correction. Thus, cumulative geodesic centroid selection did not provide a modularity advantage on this network.

For Power-Grid, Louvain again produced the highest modularity (0.935995), closely followed by Fast-Greedy (0.934566). MAXMIN and RNHT obtained values of 0.905257 and 0.899126, respectively. LDNHT achieved 0.878877. Although this value was significantly higher than HDNHT (0.875422, Holm-adjusted), it was significantly lower than RNHT, MAXMIN, Louvain, and Fast-Greedy.

On Wiki-Vote, Louvain produced the highest modularity of 0.423083, followed by Fast-Greedy at 0.339913. LDNHT obtained 0.166133. Its difference from RNHT (0.158843) was not statistically significant after Holm correction ($p = 0.295$), whereas LDNHT significantly outperformed HDNHT (0.119510). MAXMIN achieved a significantly higher modularity of 0.191287.

For Gnutella, Louvain (0.463550) and Fast-Greedy (0.461630) substantially outperformed the centroid-based approaches. Among the centroid strategies, RNHT produced 0.303798, MAXMIN 0.296548, LDNHT 0.281866, and HDNHT 0.259772. LDNHT significantly exceeded HDNHT but remained significantly below RNHT, MAXMIN, Louvain, and Fast-Greedy.

Overall, the modularity results show that LDNHT should not be characterized as a modularity-maximizing alternative to Louvain or Fast-Greedy. Its principal contribution lies in the cumulative geodesic centroid-selection mechanism and the structural trade-offs observed across different network types. In particular, improvements in ECR or coverage on some datasets do not necessarily translate into higher modularity.

4.3 Conductance performance

Conductance was used to assess the separation between detected communities and the remainder of the network. Lower conductance values indicate better-defined community boundaries. Table 4 shows that LDNHT does not consistently minimize conductance and that its relative performance varies substantially across network structures. For Facebook, Louvain achieved the lowest mean conductance of 0.054277, followed closely by Fast-Greedy at 0.054633. LDNHT produced a mean conductance of 0.538496. It significantly outperformed HDNHT (0.638840), but its conductance was significantly higher than RNHT, MAXMIN, Louvain, and Fast-Greedy.

Table 2. External connection ratio (mean ± SD) across the primary experiments

Dataset	RNHT	HDNHT	MAXMIN	LDNHT	Louvain	Fast-Greedy
Facebook	0.091855 ± 0.018515	0.029221 ± 0.000000	0.132779 ± 0.013725	0.099438 ± 0.014902	0.041121 ± 0.000528	0.031039 ± 0.000000
Power-Grid	0.046403 ± 0.002816	0.055683 ± 0.000000	0.050313 ± 0.001335	0.040626 ± 0.002185	0.023138 ± 0.000815	0.022545 ± 0.000000
Wiki-Vote	0.297806 ± 0.041790	0.323327 ± 0.000000	0.231051 ± 0.050037	0.185444 ± 0.039908	0.160078 ± 0.015435	0.110571 ± 0.000000
Gnutella	0.414336 ± 0.012124	0.423643 ± 0.000000	0.398213 ± 0.019776	0.428068 ± 0.025699	0.323322 ± 0.006470	0.287701 ± 0.000000

Note: Lower ECR indicates better local cohesion. Bold values represent the lowest mean ECR within each dataset.

Table 3. Modularity (mean ± SD) across the primary experiments

Dataset	RNHT	HDNHT	MAXMIN	LDNHT	Louvain	Fast-Greedy
Facebook	0.692720 ± 0.015750	0.692781 ± 0.000000	0.710121 ± 0.015977	0.628204 ± 0.014026	0.834943 ± 0.000189	0.777378 ± 0.000000
Power-Grid	0.899126 ± 0.002377	0.875422 ± 0.000000	0.905257 ± 0.002293	0.878877 ± 0.001696	0.935995 ± 0.000467	0.934566 ± 0.000000
Wiki-Vote	0.158843 ± 0.021283	0.119510 ± 0.000000	0.191287 ± 0.017742	0.166133 ± 0.028479	0.423083 ± 0.004724	0.339913 ± 0.000000
Gnutella	0.303798 ± 0.007506	0.259772 ± 0.000000	0.296548 ± 0.008761	0.281866 ± 0.004015	0.463550 ± 0.003070	0.461630 ± 0.000000

Note: Higher modularity indicates stronger global community structure. Bold values represent the highest mean modularity within each dataset.

Table 4. Conductance (mean ± SD) across the primary experiments

Dataset	RNHT	HDNHT	MAXMIN	LDNHT	Louvain	Fast-Greedy
Facebook	0.402033 ± 0.045719	0.638840 ± 0.000000	0.298859 ± 0.018249	0.538496 ± 0.007605	0.054277 ± 0.003953	0.054633 ± 0.000000
Power-Grid	0.088795 ± 0.015028	0.161687 ± 0.000000	0.069143 ± 0.001749	0.306074 ± 0.012922	0.035620 ± 0.001519	0.037147 ± 0.000000
Wiki-Vote	0.729875 ± 0.028457	0.779926 ± 0.000000	0.676806 ± 0.062938	0.710734 ± 0.039397	0.365306 ± 0.037413	0.562429 ± 0.000000
Gnutella	0.619457 ± 0.011436	0.687681 ± 0.000000	0.611463 ± 0.012410	0.651098 ± 0.012174	0.490551 ± 0.017881	0.439177 ± 0.000000

Note: Lower conductance represents stronger separation between communities. Bold values indicate the lowest mean conductance within each dataset.

For Power-Grid, LDNHT recorded a conductance of 0.306074, which was the highest among the six methods. Louvain produced the strongest separation with 0.035620, followed by Fast-Greedy (0.037147), MAXMIN (0.069143), RNHT (0.088795), and HDNHT (0.161687). The differences between LDNHT and all five baselines were statistically significant after Holm correction. On Wiki-Vote, Louvain again achieved the lowest conductance at 0.365306, followed by Fast-Greedy at 0.562429. LDNHT obtained 0.710734. Its conductance was significantly lower than HDNHT (0.779926), while the differences relative to RNHT (0.729875) and MAXMIN (0.676806) were not statistically significant after Holm correction. Louvain and Fast-Greedy remained significantly better than LDNHT.

For Gnutella, Fast-Greedy produced the lowest conductance of 0.439177, followed by Louvain at 0.490551. LDNHT achieved 0.651098. It significantly improved on HDNHT (0.687681) but remained significantly worse than RNHT, MAXMIN, Louvain, and Fast-Greedy.

Overall, the conductance results indicate that cumulative geodesic centroid selection does not systematically improve boundary separation. LDNHT shows some advantage over HDNHT on Facebook, Wiki-Vote, and Gnutella, but modularity-based methods generally produce substantially lower conductance. This reinforces the interpretation of LDNHT as a structurally distinct centroid-selection mechanism rather than a universally superior community-detection method.

4.4 Coverage performance

Coverage measures the proportion of network edges that fall within the detected communities. Higher coverage indicates that a larger fraction of edges is retained as intra-community connections. Table 5 shows that LDNHT performs comparatively well on Power-Grid and Wiki-Vote, although its performance remains network-dependent. For Facebook, Fast-Greedy achieved the highest mean coverage of 0.961364, closely followed by Louvain at 0.961079 and HDNHT at 0.952399. LDNHT obtained 0.919682. It significantly outperformed RNHT (0.892644) and MAXMIN (0.842197), but remained significantly below HDNHT, Louvain, and Fast-Greedy.

For Power-Grid, Fast-Greedy again produced the highest coverage at 0.967091, followed closely by Louvain at 0.966502. LDNHT achieved 0.949968 and significantly exceeded all three centroid-based baselines: RNHT (0.942177), HDNHT (0.921595), and MAXMIN (0.938078). However, its coverage remained significantly below Louvain and Fast-Greedy.

The Wiki-Vote results provide the strongest coverage evidence for LDNHT. It achieved a mean coverage of 0.676352 ,substantially exceeding RNHT (0.507955), HDNHT (0.426809), and MAXMIN (0.600922), with all three improvements statistically significant after Holm correction. LDNHT was also statistically indistinguishable from Louvain, which obtained 0.678628(p_{Holm} = 0.877899). Fast-Greedy remained significantly higher at 0.805561.

For Gnutella, LDNHT achieved a coverage of 0.392878 . This was descriptively higher than HDNHT (0.388670), but the difference was not statistically significant (p_{Holm} = 0.190061). RNHT (0.431574), MAXMIN (0.461630), Louvain (0.572732), and Fast-Greedy (0.615614) all produced higher coverage, with these differences statistically significant. Overall, the coverage analysis shows that LDNHT provides its clearest advantages over alternative centroid-based initialization strategies on Power-Grid and Wiki-Vote. In particular, the Wiki-Vote result is notable because LDNHT achieved coverage statistically comparable to Louvain despite substantially different centroid-selection and partitioning mechanisms. Nevertheless, the Gnutella results again confirm that the effectiveness of cumulative geodesic dispersion depends on network structure.

4.5 Runtime and robustness analysis

Runtime was evaluated using wall-clock execution time in seconds (Table 6). For the centroid-based methods, the measured time included centroid initialization, shortest-path computations, centroid selection, and node assignment. Lower values indicate greater computational efficiency. The results demonstrate substantial differences across algorithms and datasets.

For Facebook, HDNHT was the fastest method with a mean runtime of 0.156732 s, followed by Louvain at 1.445546 s. LDNHT required 7.670764 s, making it significantly slower than RNHT (7.385707 s), HDNHT, and Louvain. However, LDNHT was significantly faster than MAXMIN (9.572425 s) and Fast-Greedy (22.497992 s).

For Power-Grid, HDNHT again produced the lowest runtime at 0.125446 s, followed by Louvain (0.263400 s) and Fast-Greedy (0.554820 s). LDNHT required 6.925687 s, compared with 6.894272 s for MAXMIN. The difference between LDNHT and MAXMIN was not statistically significant after Holm correction ($p = 0.501$). LDNHT was significantly slower than RNHT, HDNHT, Louvain, and Fast-Greedy.

On Wiki-Vote, HDNHT achieved the lowest runtime at 0.129808 s, followed by Louvain at 1.278974 s. LDNHT required 7.534413 s, while MAXMIN required 7.476026 s. Their difference was not statistically significant ($p = 0.865$). LDNHT was significantly slower than RNHT but substantially and significantly faster than Fast-Greedy, which required 148.030023 s.

For Gnutella, HDNHT and Louvain again provided the lowest mean runtimes at 0.132328 s and 0.954031 s, respectively. LDNHT required 6.599244 s, compared with 6.764204 s for MAXMIN. Although LDNHT was descriptively faster than MAXMIN, this difference was not statistically significant ($p = 0.174$). LDNHT was significantly faster than Fast-Greedy (15.624927 s), but significantly slower than RNHT, HDNHT, and Louvain.

Overall, LDNHT does not provide the lowest computational runtime. Its execution cost is broadly comparable with MAXMIN because both methods repeatedly perform geodesic-distance calculations during centroid initialization. In contrast, HDNHT and Louvain were consistently faster under the adopted implementation. Nevertheless, LDNHT was substantially faster than Fast-Greedy on Facebook, Wiki-Vote, and Gnutella. Therefore, computational efficiency should be interpreted together with the structural characteristics of the resulting partitions rather than as evidence of uniform runtime superiority.

Robustness checks were performed to examine initialization sensitivity, the number of internal trials T, the predefined community count k, ground-truth recovery, and empirical scalability. Increasing T reduced dependence on the randomly selected first centroid. Mean LDNHT modularity increased from 0.5052 to 0.6291 on Facebook, 0.8532 to 0.8783 on Power-Grid, 0.0621 to 0.1454 on Wiki-Vote, and 0.1697 to 0.2806 on Gnutella between $T = 1$ and $T = 30$. However, the additional gain from $T = 20$ to $T = 30$ was small on all four networks, indicating diminishing returns beyond approximately 20 trials. The k-sensitivity analysis further showed that the Louvain-calibrated primary k was not necessarily optimal for LDNHT. Power-Grid reached 0.8834 at $k = 43$, compared with 0.8783 at $k = 39$. These results confirm that k should be treated as a model parameter rather than an intrinsically optimal value.

Table 5. Average (mean ± SD) across the primary experiments

Dataset	RNHT	HDNHT	MAXMIN	LDNHT	Louvain	Fast-Greedy
Facebook	0.892644 ± 0.030006	0.952399 ± 0.000000	0.842197 ± 0.014628	0.919682 ± 0.017339	0.961079 ± 0.000248	0.961364 ± 0.000000
Power-Grid	0.942177 ± 0.003646	0.921595 ± 0.000000	0.938078 ± 0.001599	0.949968 ± 0.003253	0.966502 ± 0.001110	0.967091 ± 0.000000
Wiki-Vote	0.507955 ± 0.073165	0.426809 ± 0.000000	0.600922 ± 0.083516	0.676352 ± 0.068428	0.678628 ± 0.029654	0.805561 ± 0.000000
Gnutella	0.431574 ± 0.026581	0.388670 ± 0.000000	0.461630 ± 0.047085	0.392878 ± 0.050509	0.572732 ± 0.008457	0.615614 ± 0.000000

Note: Higher coverage indicates a larger proportion of edges contained within detected communities. Bold values indicate the highest mean coverage for each dataset.

Table 6. Runtime in seconds (mean ± SD) across the primary experiments

Dataset	RNHT	HDNHT	MAXMIN	LDNHT	Louvain	Fast-Greedy
Facebook	7.385707 ± 1.069961	0.156732 ± 0.007164	9.572425 ± 0.787289	7.670764 ± 0.388959	1.445546 ± 0.453330	22.497992 ± 8.109993
Power-Grid	4.716530 ± 0.156448	0.125446 ± 0.009878	6.894272 ± 0.079844	6.925687 ± 0.121214	0.263400 ± 0.018428	0.554820 ± 0.026476
Wiki-Vote	6.937632 ± 0.360562	0.129808 ± 0.016445	7.476026 ± 0.491588	7.534413 ± 0.751673	1.278974 ± 0.191128	148.030023 ± 11.026161
Gnutella	4.990771 ± 0.230826	0.132328 ± 0.008605	6.764204 ± 0.484551	6.599244 ± 0.477791	0.954031 ± 0.148281	15.624927 ± 0.820724

For example, Facebook achieved modularity of 0.6514 at $k = 12$, compared with 0.6291 at the primary $k = 16$, while Ground-truth recovery was further evaluated on LFR benchmark networks using ten independently generated instances at each mixing level $\mu = 0.10, 0.30$, and 0.50. Table 7 reports the mean NMI and ARI values for all six algorithms. Recovery declined for all centroid-based methods as community mixing increased. At $\mu = 0.10$, LDNHT achieved NMI/ARI of 0.5179/0.3378, comparable with RNHT but below MAXMIN. At $\mu = 0.30$, LDNHT obtained 0.2792/ 0.1114, while at $\mu = 0.50$ its performance decreased to 0.1414/0.0226. Louvain achieved the strongest ground-truth recovery across all three mixing levels, while Fast-Greedy was also substantially stronger under low and moderate mixing. These results indicate that cumulative geodesic dispersion is less effective as planted community boundaries weaken.

Finally, the sparse-network scalability experiment from 1,000 to 8,000 nodes produced an approximately linear empirical runtime trend for LDNHT, with a log-log slope of 0.911. This supports practical scalability under sparse-network conditions but does not imply the same behavior for genuinely dense graphs.

5. Discussion

The revised experiments indicate that LDNHT provides a distinct cumulative geodesic-dispersion strategy for centroid selection, but its effectiveness depends strongly on network structure. The main contribution should therefore be viewed as a different centroid-initialization method rather than a generally better community-detection algorithm. Among centroid-based methods, Power-Grid and Wiki-Vote performed best; Wiki-Vote achieved the lowest ECR and highest coverage, while Power-Grid outperformed the other centroid-based methods. However, its strength was more limited on Facebook, and Gnutella was a negative example in which LDNHT's ECR and coverage were relatively low. These differences show that partitions are not necessarily better with greater centroid dispersion. The modularity results are an important qualification of the proposed method. The modularity of Louvain and Fast-Greedy was typically much higher than LDNHT, especially for Facebook, Wiki-Vote, and Gnutella. This is expected because these methods optimize modularity, whereas LDNHT builds communities using shortest-path-based centroid placement and nearest-centroid assignment. Therefore, rather than maximizing a global community-quality goal, LDNHT focuses on whether cumulative geodesic separation yields interpretable and reproducible community representatives.

This interpretation is supported by the conductance results, as LDNHT was not the best at consistently separating the boundaries. The robustness analyses also help elucidate the method's behavior. With more internal trials, fewer trials depended on the randomly selected first centroid, but it did not appear to make much difference after about 20 trials. The sensitivity analysis also revealed that the Louvain-calibrated community count was not necessarily best for LDNHT. As a result, the initialization and the chosen value remain pertinent model-design considerations. A further conservative test of Ground-truth LFR experiments showed that LDNHT recovered well at low community mixing, while NMI and ARI decreased significantly with increasing mixing. This indicates that cumulative geodesic dispersion becomes less informative when community boundaries are weak. Computationally, LDNHT was broadly comparable to MAXMIN because both require repeated shortest-path calculations. It was slower than HDNHT and Louvain under the adopted implementation, although it remained considerably faster than Fast-Greedy on several of the larger real networks. The scalability experiment nevertheless showed approximately linear empirical runtime growth on the tested sparse graphs, which is consistent with the BFS-based complexity analysis when k, T, and graph sparsity remain controlled.

From an application perspective, the cumulative-distance centroid strategy may be useful in infrastructure networks that require geographically or topologically dispersed representative nodes. For example, in a smart-grid communication network, communities may represent groups of electrically or communicationally related devices, while selected centroids may serve as candidate representative nodes for monitoring, aggregation, or decentralized coordination. By favoring nodes that are collectively distant from previously selected representatives, LDNHT can distribute such representatives across different regions of the network rather than concentrating them within a single dense area. The Power-Grid results provide preliminary support for this interpretation because LDNHT achieved favorable ECR and coverage relative to the other centroid-based methods. Similar structural partitioning may be relevant to ICT and cyber-physical networks for distributed monitoring, network segmentation, and representative-node placement. These applications remain prospective, however, and deployment-specific validation would be required before operational use. Several limitations should be acknowledged. The current implementation assumes simple, connected, undirected, and unweighted networks and requires a predefined value of k.

Table 7. Mean NMI and ARI on LFR benchmark networks across different mixing levels

Algorithm	μ = 0.10 NMI	μ = 0.10 ARI	μ = 0.30 NMI	μ = 0.30 ARI	μ = 0.50 NMI	μ = 0.50 ARI
RNHT	0.5209	0.3263	0.2668	0.0966	0.1485	0.0252
HDNHT	0.4506	0.2593	0.3194	0.1162	0.1952	0.0416
MAXMIN	0.6194	0.4517	0.3256	0.1347	0.1808	0.0386
LDNHT	0.5179	0.3378	0.2792	0.1114	0.1414	0.0226
Louvain	1.0000	1.0000	0.9745	0.9582	0.3771	0.2316
Fast-Greedy	0.9053	0.7593	0.5774	0.3102	0.1853	0.0785

Extension to weighted, directed, disconnected, overlapping, or genuinely dense networks requires additional methodological development. Moreover, modularity was used both to select the best internal restart and as one of the reported evaluation measures; therefore, modularity results for stochastic centroid methods should be interpreted with this selection procedure in mind. Overall, LDNHT is best positioned as an interpretable cumulative-distance centroid framework whose usefulness varies with topology and parameter configuration, rather than as a replacement for established modularity-optimization methods.

6. Conclusion

The Longest Distance Node Head Technique (LDNHT) introduces a centroid-based community detection framework in which community representatives are selected through cumulative geodesic dispersion. However, evaluation on Facebook, Power-Grid, Wiki-Vote, and Gnutella showed that effectiveness depends heavily on network structure and is not consistently better than alternative networks. The greatest benefits were seen on Power-Grid and Wiki-Vote, where the LDNHT was significantly better than the RNHT, HDNHT, and MAXMIN in terms of ECR and coverage. Louvain and Fast-Greedy, by contrast, typically had higher modularity, whereas Gnutella revealed problems with the proposed distance-based approach. The sensitivity analysis indicated that repeated initialization reduces sensitivity to the first random centroid chosen, and that more than about 20 internal trials do not yield significant improvements. Recovery performance also decreased as community mixing increased, as confirmed by the ground-truth evaluation on LFR benchmark networks. Empirical scalability analysis showed nearly linear runtime growth over the sparse networks tested. In conclusion, LDNHT is best suited as an interpretable cumulative-distance centroid-selection framework, but not as a standalone alternative to existing modularity-optimization algorithms. It is currently available only for simple, connected, unweighted, and undirected graphs and for a fixed value of k. In the future, adaptive estimation of k and extensions to weighted, directed, disconnected, overlapping, and dynamic network settings could be considered.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with research ethics policies. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere. All survey participants provided informed consent before participating, and the anonymity and confidentiality of all respondents were strictly maintained throughout the research process.

Data availability statement

The manuscript contains all the data. However, additional data will be provided by the corresponding author upon reasonable request.

Conflict of interest

The authors declare no potential conflict of interest.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," Sci. Rep., vol. 9, art. 5233, 2019.

[3] S. Gupta and S. Deodhar, "Understanding digitally enabled complex networks: A plural granulation based hybrid community detection approach," Inf. Technol. People, vol. 37, no. 2, pp. 919–943, 2024.

[4] A. Bouyer, P. Shahgholi, B. Arasteh, A. G. Oskouei, and X. Liu, "Community detection via core node identification and local label diffusion with GraphSAGE boundary refinement in complex networks," J. Netw. Comput. Appl., vol. 235, art. 104399, 2025.

[5] P. Danner and H. de Meer, "Community detection in energy networks based on energy self-sufficiency and dynamic flexibility activation," in Proc. 16th ACM Int. Conf. Future Sustain. Energy Syst. (E-Energy), 2025, pp. 564–576.

[6] T. Hachaj and J. Wąs, "Rough neighborhood graph: A method for proximity modeling and data clustering," Appl. Soft Comput., vol. 171, art. 112789, 2025.

[7] C. He, X. Fei, Q. Cheng, H. Li, and Y. Tang, "A survey of community detection in complex networks using nonnegative matrix factorization," IEEE Access, vol. 10, pp. 103232–103257, 2022.

[8] B. Saha, "Complex networks, communities, and clustering: A survey of community detection techniques," J. Netw. Comput. Appl., vol. 215, art. 103612, 2023.

[9] F. Dabaghi-Zarandi, M. Sadeghi, and M. R. Meybodi, "Community detection in complex networks using local and global information," Appl. Intell., vol. 52, no. 10, pp. 11423–11438, 2022.

[10] J. Yao, H. Liu, and Y. Zhang, "Community detection method for complex networks based on node influence analysis," Symmetry, vol. 16, no. 6, art. 754, 2024.

[11] L. Waltman and N. J. van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," Eur. Phys. J. B, vol. 86, no. 11, art. 471, 2013.

[12] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in Proc. IEEE 13th Int. Conf. Data Mining (ICDM), 2013, pp. 1151–1156.

[13] J. Xie and B. K. Szymanski, "LabelRank: A stabilized label propagation algorithm for community detection in networks," in Proc. IEEE 2nd Netw. Sci. Workshop (NSW), 2013, pp. 138–143.

[14] G. Cordasco and L. Gargano, "Label propagation algorithm: A semi-synchronous approach," Int. J. Soc. Netw. Min., vol. 1, no. 1, pp. 3–26, 2012.

[15] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, pp. 855–864.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

[17] L. Lü, D. Chen, X. L. Ren, Q. M. Zhang, Y. C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," Phys. Rep., vol. 650, pp. 1–63, 2016.

[18] D. Chen, L. Lü, M. S. Shang, Y. C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," Physica A, vol. 391, no. 4, pp. 1777–1787, 2012.

[19] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," ACM Comput. Surv., vol. 51, no. 2, art. 35, 2018.

[20] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Phys. Rev. E, vol. 78, no. 4, art. 046110, 2008.

[21] J. Leskovec and R. Sosič, "SNAP: A general-purpose network analysis and graph-mining library," ACM Trans. Intell. Syst. Technol., vol. 8, no. 1, pp. 1–20, 2016.

[22] M. N. Al-Andoli, S. C. Tan, W. P. Cheah, and S. Y. Tan, "A review on community detection in large complex networks from conventional to deep learning methods: A call for the use of parallel meta-heuristic algorithms," IEEE Access, vol. 9, pp. 96501–96527, 2021.

[23] E. Karampour, M. R. Malek, and M. Eidi, "Discrete Ricci flow: A powerful method for community detection in location-based social networks," Comput. Electr. Eng., vol. 123, art. 110302, 2025.

[24] A. Amini, M. Paez, and L. Lin, "Hierarchical stochastic block model for community detection in multiplex networks," Bayesian Anal., vol. 19, no. 1, pp. 319–345, 2024.

[25] A. Kumar, A. Kumari, P. Kumar, and R. Dohare, "Overlapping community detection with a new modularity measure in directed weighted networks," Data Min. Knowl. Discov., vol. 39, no. 6, art. 77, 2025.

[26] Ü. Işlak and B. Yeşiloğlu, "Analysis of clustering and degree index in random graphs and complex networks," J. Appl. Probab., vol. 61, no. 4, pp. 1–31, 2024.

[27] M. V. Sankar, B. Ravindran, and S. Shivashankar, "CEIL: A scalable, resolution-limit-free approach for detecting communities in large networks," in Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI), 2015, pp. 2097–2103.

[28] J. Leskovec and R. Sosič, "SNAP: A general-purpose network analysis and graph-mining library," ACM Trans. Intell. Syst. Technol., vol. 8, no. 1, pp. 1–20, 2016. [Online]. Available: https://snap.stanford.edu/data/

[29] J. Leskovec and A. Krevl, "SNAP Datasets: Wiki-Vote network dataset," 2014. [Online]. Available: https://snap.stanford.edu/data/wiki-Vote.html

[30] J. Leskovec and A. Krevl, "SNAP Datasets: Gnutella peer-to-peer network (p2p-Gnutella08)," 2014. [Online]. Available: https://snap.stanford.edu/data/p2p-Gnutella08.html

[31] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in Proc. AAAI Conf. Artif. Intell., 2015. [Online]. Available: https://networkrepository.com/inf-power.php

[32] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," Theor. Comput. Sci., vol. 38, pp. 293–306, 1985, doi: 10.1016/0304-3975(85)90224-5.