Annales Mathematicae et Informaticae
61 (2025) pp. 229–247
DOI: 10.33039/ami.2025.10.012
URL: https://ami.uni-eszterhazy.hu

Betweenness-driven overlapping label propagation community detection*

Sylvert Prian Tahalea[ab], **Miklós Krész**[cd]

[a]Doctoral School of Computer Science, University of Szeged, Hungary
sylvert@inf.u-szeged.hu

[b]Department of Informatics, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

[c]Department of Applied Informatics, University of Szeged, Bolodgasszony sgt. 6, H-6725 Szeged, Hungary

[d]InnoRenew CoE, UP IAM and UP FAMNIT, University of Primorska, Titov trg 4, 6000 Koper, Slovenia
miklos.kresz@innorenew.eu

Abstract. Community detection holds significant value in discovering hidden structures in complex networks. In this paper, we propose a betweenness-driven community detection based on the label propagation algorithm. First, at the multiple labels' assignment phase, we detect communities using label propagation and apply labels for the nodes using the betweenness and degree centrality as references. Second, we refine the modularity and stability using several configurations, such as global modularity and stability pruning, to avoid nodes that have not changed for several iterations. This algorithm was tested with the most common datasets, such as Zachary's Karate Club Network, Polbooks, Football, and 12 LFR synthetic datasets, which resulted in improved scores on modularity, overlapping normalised mutual information, omega index, generalized F1-score, and extended various pieces of information.

Keywords: network, graph, overlapping, community detection, label propagation algorithm

*The research was supported by the BioLOG project: the second author is grateful for the support of the National Centre of Science (NCN) through grant DEC-2020/39/I/HS4/03533, the Slovenian Research and Innovation Agency (ARIS) through grant N1-0223 and the Austrian Science Fund (FWF) through grant I 5443-N. This work is supported by the ARIS research program P1-0404 and by the research program CogniCom (0013103) at the University of Primorska.

Accepted: October 8, 2025
Published online: October 28, 2025

1. Introduction

Network analysis is implemented and used in various disciplines to represent their complex interactions, such as social sciences, computer sciences, biology, and materials science. Networks consisting of nodes with edges connecting them encode information through their structural characteristics; one of the most popular research areas is the study of community structures. While it has no formal definition, a community can be considered a densely connected subgraph of a network, which means that nodes within the community have strong relationships but are sparsely connected to the rest of the network. Generally, there are two types of communities: non-overlapping or disjoint communities and overlapping communities. The non-overlapping community detection separated the nodes based on their membership, where each node only belongs to exactly one community. Meanwhile, the overlapping communities consider the nodes that belong to one or more communities.

Community detection is considered an important task because it can uncover the hidden structure of a complex network. Most community detection algorithms have been developed to solve the non-overlapping community detection problem, but some algorithms work well to solve both non-overlapping and overlapping community detection problems [4]. Many approaches have been proposed to solve the community detection problems, such as clique percolation approach [23], label propagation [25], non-negative matrix factorization (NMF) [28], fuzzy set theory [8], evolutionary algorithms [24], and even the statistical models [9]. The Newman-Girvan modularity measurements have become one of the most popular methods to measure the density of communities within the network [21], since they provide an objective way to evaluate the communities' quality. This measurement indicates that nodes are more closely connected to their community compared to other nodes in the network. A modularity score near zero indicates that there is no real community structure, while a score near one means the communities are dense and well-structured. The label propagation algorithm (LPA) approach has received a lot of interest because of its simplicity and scalability [7]. As an extension of its original non-overlapping version [25], it consists of simple steps for the overlapping community detection [8], such as 1) label every node with its unique label; 2) label the current node based on its neighbours' labels; 3) propagate for all the nodes in the network; 4) compare the label of each node in the current iteration with the previous iteration; and finally 5) labels indicate the communities for each node (a node may have multiple labels). The weighting system is usually applied in stage 3. The iteration is terminated at stage 4 if convergence occurs; otherwise, another iteration is executed.

There are several notable overlapping community algorithms based on label propagation. In 2010, Gregory [8] designed a specific algorithm called COPRA (Community Overlap Propagation Algorithm) to allow nodes to hold multiple labels simultaneously using the belonging coefficient as a degree of membership. Later, the dynamic process where each node acts as a speaker and a listener was

introduced as SLPA [29], which identified both the number and the strength of each node's community affiliation. Another approach, which combined with LPA is a local spectral method called LEMON [15]: a sparse vector is obtained by minimising the $\ell 1$ norm over a local spectral subspace with seed constraints, which ensures the seed nodes to be included and highlights additional nodes to include in the community.

In 2021, Attal et. al. [1] introduced a method to find overlapping community detection using pre-computed disjoint communities. This algorithm, leveraging density and clustering coefficient as belonging function, compares to closeness and betweenness centrality as average node measures, defining a node's memberships. These results show that communities with high density and clustering coefficient performed better than closeness and betweenness centrality measures. In the same year, Li and Sun [14] introduced a combination of local expansion and label propagation (LELP), which uses local expansion to generate some immature communities, prunes the network, and uses LPA to obtain a stable network.

In 2022, density based-label propagation algorithm (D-LPA) [31] was developed, combining the density peak clustering with traditional LPA to improve the stability and accuracy of community assignments. The vector-label propagation algorithm (vLPA) was also introduced in this year [3], where gradient descent was utilised to improve the modularity. This approach retains weak structural information, but obtains better performance when the community structure is weak. The influence-based COPRA approach introduced as INF-COPRA [30], is an algorithm that ranks the influence of nodes and labels, thereby improving the extended modularity (EQ) and normalised mutual information. One of the latest expansions of LPA was the degree and betweenness-based label propagation (DBLPA) [22], which combines the degree and betweenness centrality to provide the core nodes in layer-by-layer LPA [34].

Label propagation algorithms have a fast runtime but have strong randomness and weak robustness causing difficulties in obtaining effective community detection results. The expansion of the label propagation algorithms conducted with several approaches, such as the multistep greedy, to increase modularity, but sacrifices the fast-running time [17]. There is also a kernel label approach proposed to reduce the complexity, but at the same time improve the randomness of the algorithm [16]. Meanwhile, others proposed the accelerated modularity gain by analysing Newman's modularity function [33].

Overlapping community detection is important because real-world networks rarely consist of cleanly separated groups. Detecting overlapping communities can lead to the nature of multiple roles of nodes, provide more accurate prediction of network behaviour, capture phenomena such as redundancy and robustness in complex systems, and define the characteristics that strengthen the interpretability and predictive utility of community detection methods. Modularity is one of the most widely accepted quality functions in community detection because it measures how well a given community separates dense intra-community connections from sparse inter-community connections. However, overlapping communities often underes-

timates the quality of the community structure because nodes can contribute to multiple groups simultaneously. If the overlapping assignments are not refined, the modularity may remain low, indicating communities are not cohesive or not well-defined. Improving modularity in overlapping community detection ensures the structural density and avoids the risk of generating fragmented communities.

Betweenness centrality quantifies the shortest paths between others, showing a strong indicator of a boundary in the networks. In overlapping community detection, betweenness nodes are precisely where community memberships are likely to overlap since they act as the bridge across different groups of nodes in the network. This betweenness-driven approach leverages the inverse betweenness, causing these nodes to retain multiple memberships in the overlap assignment phase. This approach utilises the balance between degree centrality capturing the local influences and betweenness centrality, which highlights global structure.

This paper proposes a novel overlapping community detection algorithm utilising the fast propagation through the network of LPA combined with the nature of the betweenness centrality score of a node. This algorithm is split into two phases: multi-assignment label propagation and modularity refinement. In the first phase, the algorithm will quickly build an overlapping community structure using LPA with betweenness as its voting mechanism to decide the memberships of each node. This allows strong local leaders to influence community growth while boundary nodes exert more measured influence. In the second phase, modularity refinement is executed using Newman's modularity [21] with temporal projections to refine the modularity of the communities within the network. Finally, overlapping communities are defined by assigning the nodes to all communities where their membership strength exceeds the threshold. In this proposed algorithm, several adaptation parameters are used to maximise the modularity, such as top-k filtering to make sure that the nodes can hold k maximum labels as their possible communities, and minimum gain as a threshold when maximising the modularity.

The main contributions of this paper are as follows:

1. Balancing high-speed and high-quality communities, addressing the weaknesses of the classic label propagation algorithm.

2. Flexible filtering strategies (such as Top-K filtering, minimum gain threshold, and stability pruning) which lead to an increase in the quality of communities produced.

3. Utilising the nature of betweenness nodes to define the overlapping nodes.

The remainder of the paper is organised as follows. Section 2 explains the methodology, data, and evaluation techniques used in this study. Section 3 presents the main results, with tables highlighting the main findings. Section 4 concludes the contribution, takeaways, and future research.

2. Methodology

The Betweenness-Driven Label Propagation Algorithm (BD-LPA) is an overlapping community detection method that combines local label propagation with a global modularity-based refinement. The high-betweenness nodes often sit at community boundaries and influence how strongly a node is pulled by its neighbours' labels. By seeding each node with a unique label and weighting those labels according to both how often they appear in the neighbourhood and how central their owners are, BD-LPA builds a soft membership vector for every node. This vector encodes the node's membership towards multiple communities, allowing for overlapping structures. The algorithm proceeds in two main phases—first, a fast, distributed voting scheme that spreads labels with influence proportional to neighbour degree and inverse to neighbour betweenness, and then a slower, global refinement that uses modularity gains to reinforce coherent groupings. This proposed method is split into two phases, namely the multi-assignment label propagation phase and the global modularity refinement phase.

2.1. Degree and betweenness centrality

Degree centrality measures the number of edges connected to a node [5]. This centrality shows the position of a node in the networks based on its connections and can be measured as follows. For a simple undirected graph $G = (V, E)$, the degree centrality C_d of node v is

$$C_D(v) = \deg(v),$$

where $\deg(v)$ is the degree of node v.

While degree centrality measures the node's centrality using the direct connection to the node, betweenness centrality measures the node's centrality based on how often the node lies on the shortest path between other nodes [6]. Betweenness centrality C_b of a node v can be measured as follows. Let σ_{xy} be the number of shortest path between nodes x and y, and $\sigma_{xy}(v)$ the number of paths that pass through node v, with $v \neq x \neq y$. The betweenness centrality of v is

$$C_B(v) = \sum_{v \neq x \neq y} \frac{\sigma_{xy}(v)}{\sigma_{xy}}.$$

The nature of a node with a high betweenness value is becoming a hub for the network, meaning that they are node with a high possibility of having multiple community memberships. On the contrary, nodes with low betweenness are more likely to have single community membership but not necessarily rely only on their betweenness centrality value. Thus, in this study, we propose to combine the use of degree and betweenness centrality as the community membership voting mechanism.

2.2. Multi-assignment label propagation

In Phase 1, each node maintains a weight vector over candidate community labels. In each iteration, nodes are visited in random order and vote for labels: each neighbour contributes to the vote for its own dominant label in proportion to its degree divided by one plus its betweenness centrality. The candidate labels with the highest votes are then added to the node's weight vector with a fixed gain factor, and the vector is renormalised as probabilities, retaining only the top L labels. Over multiple fast iterations, this process diffuses label influence through the graph, enabling nodes at the fringes of communities to accumulate membership probabilities in several nearby groups. Since the voting weight uses degree and penalises high-betweenness nodes, labels spread more readily within densely connected regions while respecting bottlenecks. The node membership voting mechanism can be computed as follows.

$$vote[\ell_u^*] \mathrel{+}= \frac{\deg(u)}{1 + \mathrm{btw}(u)},$$

where ℓ_u^* is the dominant label of neighbor u, $\deg(u)$ is the degree of node u, $\mathrm{btw}(u)$ is the betweenness centrality of node u, and $vote[\ell_u^*]$ is the cumulative vote weight for label received by node u from its neighbors. The algorithm for phase 1 is presented in Algorithm 1.

2.3. Modularity refinement

In Phase 2, modularity refinement measurement utilises Newmann's modularity with temporary projection or hard mapping[26, 27]. This shifts the measurement from local diffusion to global optimisation by temporarily hard-assigning each node to its highest-weight label and computing the resulting modularity. It then attempts to improve modularity by considering, for each node in random order, reassigning it to one of its neighbours' labels if it results in a modularity gain above a small threshold. Whenever a better community label is found, the node's weight vector is reset to that single label before moving on. Iterations continue until no single-node swap can further increase modularity. Crucially, these hard-assignment trials only guide the search; at the end of refinement, the algorithm reverts to the soft weight vectors and applies thresholds to produce overlapping communities, preserving multi-membership while ensuring that each switch meaningfully boosts global cohesion. The algorithm for phase 2 is presented in Algorithm 2, and the modularity measurement is conducted as follows.

1. After phase-1, each node v has membership weight $w_v(\ell)$
2. Temporay projection (hard mapping) performed as $c_i^* = \arg\max_\ell w_i(\ell)$
3. Compute the modularity using Newman's modularity [21] for the temporary hard mapping as follows.

$$Q^{proj} = \frac{1}{2m} \sum [A_{ij} \frac{k_i k_j}{2m}] \delta(c_i^*, c_j^*), \tag{2.1}$$

Algorithm 1 Phase 1: Multi-Assignment Label Propagation

Require: $G = (V, E)$ ▷ undirected graph
Require: T_f ▷ fast propagation iterations
Require: k ▷ top-k votes per node
Require: $L_{\max}$ ▷ max labels per node
Require: η ▷ vote gain factor
Ensure: $\{w_v\}$ ▷ soft label weights

// Initialisation
for each $v \in V$ **do**
    assign unique label ℓ_v
    $w_v \leftarrow \{\ell_v \mapsto 1.0\}$
end for
// Propagation
for $t = 1$ to T_f **do**
    SHUFFLE(V)
    for each $v \in V$ **do**
        $vote \leftarrow \{\}$
        for each $u \in N(v)$ **do**
            $\ell_u^* \leftarrow \arg\max_\ell w_u[\ell]$
            $vote[\ell_u^*] \mathrel{+}= \dfrac{\deg(u)}{1 + \mathrm{btw}(u)}$
        end for
        $T \leftarrow$ top-k labels by $vote$
        for each $\ell \in T$ **do**
            $w_v[\ell] \mathrel{+}= \eta\, vote[\ell]$
        end for
        $w_v \leftarrow \mathrm{TruncatedSoftmax}(w_v, L_{\max})$
    end for
end for

where A_{ij} is actual adjacency between node i and j; k_i, k_j are the degrees of node i and j; m is the total number of edges; and $\delta(c_i^*, c*_j)$ shows the position of node i and j on their temporary mapping, 1 means both nodes are in the same community, and 0 otherwise.

Here is an example of the approach.

1. Graph: G with $V = \{A, B, C, D\}$, a 4-cycle, $m = 4$
2. Communities: $C_1 = \{A, B, C\}, C_2 = \{A, C, D\}$
3. Assumed that memberships after phase-1:
    - $A : C_1 : 0.7, C_2 : 0.3$
    - $B : C_1 : 0.6, C_2 : 0.4$
    - $C : C_1 : 0.2, C_2 : 0.8$
    - $D : C_1 : 0.1, C_2 : 0.9$

4. Temporal projection:
   - $A \mapsto C_1$
   - $B \mapsto C_1$
   - $C \mapsto C_2$
   - $D \mapsto C_2$
   - Projected community $C_1^* = \{A, B, C\}, C_2^* = \{D\}$
5. Using equation (2.1) $Q^{proj} = 0.08$

2.4. Evaluation

This proposed algorithm was evaluated using several standard metrics that are widely adopted, such as modularity, overlapping normalized mutual information (ONMI), omega index, generalized F1-score, and extended variation of information (extended VI) as the proper evaluation for overlapping community detection [7, 11]. Modularity introduced by Newman and Girvan [21] to quantify the internal structure of the community, while ONMI measures the similarity between the results and the ground truth [18] as follows:

$$NMI_{max} = \frac{I(X:Y)}{\max(H(X), H(Y))},$$

where $I(X:Y) = \frac{1}{2}[H(X) - H(X|Y) + H(Y) - H(X|Y)]$is the mutual information and $H(X|Y) = \sum_{i \in 1,\dots K_X}$ is the total information.

To evaluate the accuracy, imbalance community, and discrepancy in shared information, the omega index was used [20] as follows.

$$\Omega(C_1, C_2) = \frac{o_u(C_1, C_2) - o_e(C_1, C_2)}{1 - o_e(C_1, C_2)},$$

where $o_u(C_1, C_2)$ is the fraction of pairs that occur in the same number of communities in both communities, and $o_e(C_1, C_2)$ is the expected fraction under random assignment.

The generalized F1-score [32] is also used to measure the best-matching community produced by the algorithm compared with the ground truth, which can be measured as follows.

$$\frac{1}{2}\left(\frac{1}{|C^*|}\sum_{C_i \in C^*} F1(C_i, \hat{C}_{g(i)}) + \frac{1}{|\hat{C}|}\sum_{\hat{C}_i \in \hat{C}} F1(C_{g'(i)}, \hat{C}_i)\right),$$

where the best matching g and g' is defined as $g(i) = \arg\max_j F1(C_i, \hat{C}_j)$ and $g'(i) = \arg\max_j F1(C_j, \hat{C}_i)$.

Algorithm 2 Phase 2: Modularity Refinement

Require: $G = (V, E)$ ▷ undirected graph
Require: $\{w_v\}_{v \in V}$ ▷ weight vectors from Phase 1
Require: T_r ▷ refinement iterations
Require: ε ▷ modularity-gain threshold
Ensure: $\{w_v\}_{v \in V}$ ▷ refined weight vectors

function HardMap(v)
  return $\arg\max_\ell w_v[\ell]$
end function
function BuildCommunities
  $C \leftarrow \{\}$ ▷ map label→node list
  for each $v \in V$ **do**
    $\ell \leftarrow$ HardMap(v)
    append v to $C[\ell]$
  end for
  return list of communities in C
end function
$Q \leftarrow \text{modularity}\big(G, \text{BuildCommunities}\big)$
for $t \leftarrow 1$ **to** T_r **do**
  changed $\leftarrow$ false; Shuffle(V)
  for each $v \in V$ **do**
    $c_0 \leftarrow$ HardMap(v)
    Cand $\leftarrow \{\text{HardMap}(u) \mid u \in N(v)\} \setminus \{c_0\}$
    best, $\Delta_{\text{best}} \leftarrow c_0, 0$
    for each $\ell \in$ Cand **do**
      save $w_v^{\text{old}} \leftarrow w_v$
      $w_v \leftarrow \{\ell \mapsto 1.0\}$
      $Q' \leftarrow \text{modularity}\big(G, \text{BuildCommunities}\big)$
      $\Delta \leftarrow Q' - Q$
      if $\Delta > \varepsilon$ **and** $\Delta > \Delta_{\text{best}}$ **then**
        best, $\Delta_{\text{best}} \leftarrow \ell, \Delta$
      end if
      restore $w_v \leftarrow w_v^{\text{old}}$
    end for
    if best $\neq c_0$ **then**
      $w_v \leftarrow \{\text{best} \mapsto 1.0\}$
      $Q \leftarrow Q + \Delta_{\text{best}}$
      changed $\leftarrow$ true
    end if
  end for
  if not changed **then**
    break
  end if
end for

Another evaluation measurement used was the extended VI [19] as follows.

$$VI(X, Y) = H(X|Y) + H(Y|X),$$

where $H(X|Y)$ is the conditional entropy of community X given community Y and vice versa. Conditional entropy measures the certainty of a community assignment given knowledge of the other communities. It captures the average amount of extra information to describe the communities. A lower conditional entropy implies that knowing one community almost fully explains the other (strong agreement), while a higher value means the communities disagree more, reflecting greater divergence in how overlaps are assigned.

The datasets used in this evaluation are Zachary's Karate Club Network [5], Football [5], Polbooks [10], and synthetic datasets generated using the LFR framework [12].

3. Experimental results and analysis

The proposed algorithms were benchmarked using four standard metrics, such as modularity, ONMI, Omega Index, Generalized F1-score and Extended VI, over the four datasets mentioned before. The results were compared to three overlapping community detection algorithms such as COPRA [8], SLPA [29], and Motif-LPA [13], to evaluate their performance to the LPA-based algorithm.

- **COPRA**: Expansion of traditional LPA using membership coefficients to allow nodes to belong to multiple communities.
- **SLPA**: Dynamic model of LPA where nodes act as speakers and listeners, allowing nodes to remember the community label and become its member.
- **Motif-LPA**: Introducing a motif-based approach (recurring structural patterns) and capturing high-order connectivity to allow nodes to become members of more than one community

3.1. Datasets

Three real-world datasets are commonly used in community detection benchmarking, utilised in this research. The Zachary's Karate Club network describes relationships between club members, which are divided into two communities. The Football dataset was created from the American College Football League, where nodes represent the football team and edges represent the game played between them. The Polbooks network was created based on the interaction between readers of American politics books. Alongside the real-world datasets, we generated 12 synthetic datasets using the LFR model [12] with different settings. The LFR model is widely adopted in community detection research because it produces networks with realistic structural features (heterogeneous degree distributions and community sizes) while allowing precise control over the embedded community structure.

Unlike real-world networks, where communities are inferred from contextual meaning, in LFR networks, communities are explicitly defined by the generation process, making them suitable for controlled benchmarking. Each synthetic dataset

in Table 1 is characterised by: n (Nodes) and Edges: the size and connectivity of the generated network; c (Communities): the number of planted communities generated by the model; k (Average degree): the average number of edges per node; μ (Mixing parameter): the proportion of edges that connect a node to nodes outside its community. A low μ (close to 0) indicates well-separated communities, while a higher μ implies stronger inter-community mixing and weaker community structure; on (Overlapping nodes): the number of nodes that belong to more than one community; om (Overlapping memberships): the number of communities each overlapping node belongs to. For example, om = 2 indicates that overlapping nodes belong to exactly two communities, while om = 3 allows nodes to be shared across three communities.

The parameterisation across the LFR datasets was chosen to cover different scenarios of network size from 1000 to 10000 nodes, degree distributions, mixing levels, and varying degrees of community overlap. This diversity ensures the algorithm's robustness under a wide range of structural complexities and community structures. Table 1 provides the summary of the datasets, where c is the number of communities, k is the average degree, μ is the mixing parameter, *on* is overlapping nodes, and *om* is overlapping memberships.

Table 1. Comparative metadata of benchmark datasets.

Dataset	**Nodes**	**Edges**	c	k	μ	**on**	**om**
Karate	34	78	2	-	-	-	-
Football	115	613	3	-	-	-	-
Polbooks	105	441	12	-	-	-	-
LFR1	1000	10455	44	20.91	0.10	100	0
LFR2	1000	12555	35	25.11	0.30	100	2
LFR3	1000	6003	27	12.01	0.15	100	3
LFR4	2000	24555	40	24.55	0.20	200	0
LFR5	2000	31461	39	31.46	0.30	200	2
LFR6	2000	15495	32	15.49	0.15	200	3
LFR7	5000	70518	57	28.21	0.10	500	0
LFR8	5000	84899	61	33.96	0.25	500	2
LFR9	5000	93743	56	37.50	0.30	500	3
LFR10	10000	113290	61	22.66	0.10	1000	0
LFR11	10000	120047	75	24.01	0.20	1000	2
LFR12	10000	190440	84	38.09	0.25	1000	3

3.2. Parameter settings

The proposed algorithm is an adaptive algorithm by nature, meaning that the parameters can be tuned to achieve good community detection results. The configured parameters are the number of fast iterations in the first phase, the number of

refinement iterations, top-k labels, max labels, overlapping threshold, gain threshold, and gain factor. The fast iteration converged well within 8 iterations, while the modularity can be stabilised in 6 iterations. The top-3 neighbour labels is a good balance between noise and diversity, and limiting 3 labels per node helps avoid fragmentation. The low number of τ, ε, and η is to control the overlapping sets, prevent overfitting, and ensure effective propagation speed. The best average setting for the dataset used in this experiment is presented in Table 2.

Table 2. Parameter settings.

Name	**Symbol**	**Value**	**Function/Role**
Fast iterations	T_f	8	Number of label-voting passes
Refinement iterations	T_r	6	Number of modularity-refinement
Top-k labels	k	3	Number of neighbor can be considered
Max labels	L_{max}	3	Number of labels a node can hold
Overlapping threshold	τ	0.25	Weight threshold to allow a label into overlapping set
Gain threshold	ε	1e-4	Minimum modularity gain to allow a node to change label
Gain factor	η	0.6	Scaling factor for the neighboring votes
Random seed	-	75	Ensure the reproducibility

3.3. Modularity evaluation

The modularity measurement is a common and widely used method to evaluate community detection algorithms to show the density or sparsity of the results. The larger value indicates a densely connected structure within the results, but it does not indicate the accuracy of the community detection. The results presented in Table 3 show that compared to other algorithms, BD-LPA performed well and almost achieved the highest score for most datasets, with a peak score of 0.853 in LFR12. Motif-LPA performs competitively in synthetic datasets but struggles in real datasets, while SLPA and COPRA show moderate results. The average modularity across all datasets shows the superiority of BD-LPA with an average score of 0.647. This indicates BD-LPA is not only consistent but also adaptable to different network structures.

3.4. ONMI evaluation

The ONMI evaluation assesses the results of overlapping community detection algorithm against the ground truth of the datasets. The ONMI results of BD-LPA achieve the highest score on real-world datasets such as Karate, Football, and Polbooks. The results presented in Table 4 show that BD-LPA performs well for most

Table 3. Modularity score obtained from the datasets.

Dataset	SLPA	COPRA	Motif-LPA	BD-LPA
Karate	0.419	**0.420**	0.226	0.419
Football	0.520	0.530	0.557	**0.571**
Polbooks	0.370	0.380	0.415	**0.457**
LFR1.gml	0.803	0.802	0.802	**0.897**
LFR2.gml	0.000	0.000	0.494	**0.580**
LFR3.gml	0.000	0.000	0.713	**0.718**
LFR4.gml	0.501	0.000	0.509	**0.519**
LFR5.gml	0.665	0.667	**0.668**	0.662
LFR6.gml	0.000	0.000	0.736	**0.749**
LFR7.gml	0.529	0.000	0.532	**0.536**
LFR8.gml	0.610	0.595	0.610	**0.624**
LFR9.gml	0.827	0.811	**0.829**	0.812
LFR10.gml	0.000	0.000	0.613	**0.621**
LFR11.gml	0.000	0.000	**0.689**	**0.689**
LFR12.gml	0.000	0.000	0.840	**0.853**
Average	0.349	0.280	0.616	**0.647**

datasets, competing with Motif-LPA which performs strongly on synthetic datasets but shows poorly in real-world datasets. Considering its average performance, BD-LPA is superior to other algorithms with an average of 0.922 ONMI scores. This indicates that BD-LPA not only finds community with high modularity (as seen in Table 3) but also closely aligns its detected structure with ground truth across datasets.

3.5. Omega index

The omega index is specifically designed to handle overlapping community detection as the extension of the adjusted rand index (ARI) [2, 20]. This metric measures the similarity between the node memberships in predicted communities and the ground-truth communities. The range is from -1 to 1, where 1 indicates perfect similarity, 0 means random communities, and -1 means that pairwise co-membership assignments are as discordant as possible relative to chance. Table 5 presents the omega index score from the datasets. BD-LPA performed well for the real-world datasets and almost all the synthetic datasets. BD-LPA consistently achieves high scores across datasets, notably reaching the highest scores for LFR6, LFR7, LFR8, LFR9, and LFR11. Motif-LPA performs competitively and reaches high scores in the synthetic dataset; however, it performed poorly on real-world datasets. Averaging across datasets, BD-LPA attains the highest mean omega index of 0.930, indicating its ability in overlap-sensitive agreement metrics and showing its reliability across scenarios.

Table 4. ONMI score.

Dataset	**SLPA**	**COPRA**	**Motif-LPA**	**BD-LPA**
Karate	0.334	0.048	0.178	**0.837**
Football	0.283	0.283	0.339	**0.771**
Polbooks	0.296	0.429	0.444	**0.569**
LFR1.gml	0.892	0.946	**1.000**	0.947
LFR2.gml	0.649	0.838	0.980	**0.986**
LFR3.gml	0.725	0.624	**0.990**	0.981
LFR4.gml	0.743	0.876	**0.998**	0.929
LFR5.gml	0.799	0.860	**1.000**	0.939
LFR6.gml	0.750	0.672	0.980	**0.999**
LFR7.gml	0.780	0.865	**0.998**	0.987
LFR8.gml	0.805	0.859	**0.999**	0.983
LFR9.gml	0.857	0.826	**1.000**	0.950
LFR10.gml	0.806	0.892	**1.000**	0.981
LFR11.gml	0.777	0.681	0.990	**1.000**
LFR12.gml	0.814	0.748	**1.000**	0.965
Average	0.687	0.696	0.860	**0.922**

Table 5. Omega index score.

Dataset	**SLPA**	**COPRA**	**Motif-LPA**	**BD-LPA**
Karate	0.210	0.048	0.247	**0.882**
Football	0.671	0.080	0.536	**0.778**
Polbooks	0.546	**0.700**	0.568	0.637
LFR1.gml	0.954	0.425	**1.000**	0.977
LFR2.gml	0.612	-0.033	**0.989**	0.973
LFR3.gml	0.790	0.155	**0.997**	0.991
LFR4.gml	0.760	-0.024	**0.998**	0.936
LFR5.gml	0.905	0.244	**1.000**	0.964
LFR6.gml	0.855	0.166	0.985	**1.000**
LFR7.gml	0.905	-0.022	0.998	**1.000**
LFR8.gml	0.940	-0.003	**1.000**	**1.000**
LFR9.gml	0.972	0.287	**1.000**	**1.000**
LFR10.gml	0.957	0.002	**1.000**	0.900
LFR11.gml	0.946	0.184	0.991	**1.000**
LFR12.gml	0.972	0.281	**1.000**	0.957
Average	0.800	0.166	0.887	**0.930**

3.6. Generalized F1-score

The generalized F1-score compares the similarity between the predicted community with the ground-truth community. This metric focuses on quality rather than the structure of the compared communities. Table 6 presents the results of generalized F1-score across the datasets and shows that BD-LPA consistently deliver high scores across the datasets and reaches perfect alignment in LFR6. Its performance in real-world datasets is mixed, with an exceptional score of 0.971 in the Karate dataset but lower in other datasets; however, it performed exceptionally well in structured synthetic datasets. Motif-LPA also performed well in most datasets and reached perfect alignment in several cases. In terms of averages, BD-LPA leads with a mean score of 0.899, followed by Motif-LPA (0.837), COPRA (0.803), and SLPA (0.599). This indicates BD-LPA as the most accurate and balanced method across datasets, maintaining its strong precision-recall trade-offs.

Table 6. Generalized F1-score.

Dataset	SLPA	Copra	Motif-LPA	BD-LPA
Karate	0.609	0.600	0.278	**0.971**
Football	**0.616**	0.215	0.277	0.489
Polbooks	0.456	**0.726**	0.047	0.596
LFR1.gml	0.688	0.976	**1.000**	0.772
LFR2.gml	0.541	0.924	**0.993**	0.923
LFR3.gml	0.570	0.765	0.983	**0.984**
LFR4.gml	0.554	0.934	**0.999**	0.942
LFR5.gml	0.572	0.935	**1.000**	0.982
LFR6.gml	0.543	0.809	0.985	**1.000**
LFR7.gml	0.536	0.948	**0.999**	0.985
LFR8.gml	0.548	0.952	**0.999**	0.985
LFR9.gml	0.563	0.906	**1.000**	0.983
LFR10.gml	0.531	0.951	**1.000**	0.900
LFR11.gml	0.524	0.766	**0.994**	**0.994**
LFR12.gml	0.532	0.636	**1.000**	0.985
Average	0.559	0.803	0.837	**0.899**

3.7. Extended variation of information

This metric quantifies the amount of information lost and gained when the nodes move from one community to another. The extended VI score ranges from 0 upwards, where 0 means identical communities, while the larger value indicates the dissimilarity between predicted communities and the ground-truth. Table 7 presents the extended VI across the benchmark datasets. BD-LPA has the best extended VI score, indicating there is no discrepancy between its detected com-

munities and ground truth. The average score shows that BD-LPA is far superior to other algorithms, indicating its precision and consistency in producing ground-truth communities across diverse datasets.

Table 7. Extended VI score.

Dataset	SLPA	COPRA	Motif-LPA	BD-LPA
Karate	1.000	1.252	1.000	**0.306**
Football	1.000	3.276	1.000	**0**
Polbooks	1.000	0.954	1.000	**0.364**
LFR1.gml	0.008	0.012	**0.000**	**0.000**
LFR2.gml	0.0413	0.0431	0.004	**0.000**
LFR3.gml	0.028	0.125	0.003	**0.000**
LFR4.gml	0.018	0.029	**0.000**	**0.000**
LFR5.gml	0.012	0.037	**0.000**	**0.000**
LFR6.gml	0.016	0.111	0.005	**0.000**
LFR7.gml	0.007	0.024	**0.000**	**0.000**
LFR8.gml	0.005	0.024	**0.000**	**0.000**
LFR9.gml	0.004	0.033	**0.000**	**0.000**
LFR10.gml	0.003	0.014	**0.000**	**0.000**
LFR11.gml	0.004	0.048	0.001	**0.000**
LFR12.gml	0.003	0.026	**0.000**	**0.000**
Average	0.210	0.401	0.200	**0.045**

4. Conclusion and future research

In this paper, we propose an overlapping community detection method based on a label propagation algorithm (LPA) and betweenness centrality. In the presence of numerous communities, nodes that exhibit high betweenness are prospective overlapping nodes, which may lead to a reduced membership score within a community. Multilabel assignment are quickly applied to nodes using label propagation, and using the inverse betweenness as the weight for community assignment. The modularity refinement phase uses the detected communities as the base community to be evaluated with the modularity measurement. If the modularity of the node exceeds the threshold, the designation will be altered.

The utilisation of betweenness centrality as the voting mechanism provides a simple computation yet better structure for the proposed communities. The use of betweenness suppresses label flow across communities, resulting in higher modularity. This avoids false overlapping memberships since nodes with low betweenness are most likely to become a single community member node, thereby affecting the label purity, and improving ONMI scores. The limitation of a high betweenness score also contributes to pairwise consistency in co-membership across all pairs of

nodes. Betweenness also minimises label diffusion, which leads to reducing the label entropy for better performance.

Across evaluation metrics such as Modularity, ONMI, Omega Index, Generalized F1-score, and Extended VI, BD-LPA consistently performs well. It produces highly accurate and robust community structures compared to the ground truth, optimally balancing precision and recall. This enables effective and accurate identification of overlapping communities without sacrificing detection sensitivity while minimising information loss between detected and actual community structures.

This study used parameter settings that were adjusted for all datasets and produced good results across all evaluations. For specific implementation tasks, such as biological networks, social networks, or communication systems, addition parameter settings may need to be adjusted to optimise community detection tasks achieve better results. Overall, these results suggest that this approach is particularly powerful in enhancing community detection quality for structured networks, although further refinements could improve accuracy on datasets with ambiguous or noisy metadata.

Future work on this algorithm could focus on three key areas. First, **real-world network adaptation** should be conducted, as performance on irregular graphs suggests potential sensitivity to noisy or incomplete structures. Second, **the scalability and efficiency** need to be explored for handling massive networks with millions of nodes potentially through parallelised or distributed implementation. Third, **dynamic and temporal network extension** offers a promising direction for the algorithm to track evolving overlapping communities over time, which is potentially useful for applications in social media, biological networks, and communication systems. Additionally, integration with a graph neural network pipeline could allow the algorithm to serve as a high-quality label generator or preprocessor for deep learning-based community detection.

References

[1] J.-P. Attal, M. Malek, M. Zolghadri: *Overlapping community detection using core label propagation algorithm and belonging functions*, Applied Intelligence 51.11 (2021), pp. 8067–8087, doi: 10.1007/s10489-021-02250-4.

[2] L. M. Collins, C. W. Dent: *Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions*, Multivariate behavioral research 23.2 (1988), pp. 231–242, doi: 10.1207/s15327906mbr2302_6.

[3] W. Fang, X. Wang, L. Liu, Z. Wu, S. Tang, Z. Zheng: *Community detection through vector-label propagation algorithms*, Chaos, Solitons & Fractals 158 (2022), p. 112066, doi: 10.1016/j.chaos.2022.112066.

[4] S. Fortunato, D. Hric: *Community detection in networks: A user guide*, Physics reports 659 (2016), pp. 1–44, doi: 10.1016/j.physrep.2016.09.002.

[5] L. C. Freeman: *A set of measures of centrality based on betweenness*, Sociometry (1977), pp. 35–41, doi: 10.2307/3033543.

[6] L. C. Freeman: *Centrality in social networks conceptual clarification*, Social networks 1.3 (1978), pp. 215–239.

[7] S. Goswami, A. K. Singh: *A survey on overlapping community detection: label propagation*, Multimedia Tools and Applications (2024), pp. 1–30, doi: 10.1007/s11042-024-20485-4.

[8] S. Gregory: *Fuzzy overlapping communities in networks*, Journal of Statistical Mechanics: Theory and Experiment 2011.02 (2011), P02017, doi: 10.1088/1742-5468/2011/02/P02017.

[9] P. W. Holland, K. B. Laskey, S. Leinhardt: *Stochastic blockmodels: First steps*, Social networks 5.2 (1983), pp. 109–137, doi: 10.1016/0378-8733(83)90021-7.

[10] V. Krebs: *Books about US politics*, unpublished, http://www. orgnet. com (2004).

[11] A. Kumari, A. Kumar, P. Kumar: *A new modularity metric for disjoint and overlapping community structure evaluation in weighted complex networks*, International Journal of Data Science and Analytics (2025), pp. 1–25.

[12] A. Lancichinetti, S. Fortunato, F. Radicchi: *Benchmark graphs for testing community detection algorithms*, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics 78.4 (2008), p. 046110, doi: 10.1103/PhysRevE.78.046110.

[13] P.-Z. Li, L. Huang, C.-D. Wang, J.-H. Lai, D. Huang: *Community detection by motif-aware label propagation*, ACM Transactions on Knowledge Discovery from Data (TKDD) 14.2 (2020), pp. 1–19, doi: 10.1145/3378537.

[14] X. Li, Q. Sun: *Detect Overlapping Community Based on the Combination of Local Expansion and Label Propagation*, Algorithms 14.8 (2021), p. 237, doi: 10.3390/a14080237.

[15] Y. Li, K. He, D. Bindel, J. E. Hopcroft: *Uncovering the small community structure in large networks: A local spectral approach*, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 658–668, doi: 10.1145/2736277.2741676.

[16] Z. Lin, X. Zheng, N. Xin, D. Chen: *CK-LPA: Efficient community detection algorithm based on label propagation with community kernel*, Physica A: Statistical Mechanics and its Applications 416 (2014), pp. 386–399, doi: 10.1016/j.physa.2014.09.023.

[17] X. Liu, T. Murata: *Advanced modularity-specialized label propagation algorithm for detecting communities in networks*, Physica A: Statistical Mechanics and its Applications 389.7 (2010), pp. 1493–1500, doi: 10.1016/j.physa.2009.12.019.

[18] A. F. McDaid, D. Greene, N. Hurley: *Normalized mutual information to evaluate overlapping community finding algorithms*, arXiv preprint arXiv:1110.2515 (2011), doi: 10.48550/arXiv.1110.2515.

[19] M. Meilă: *Comparing clusterings by the variation of information*, in: Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings, Springer, 2003, pp. 173–187, doi: 10.1007/978-3-540-45167-9_14.

[20] G. Murray, G. Carenini, R. Ng: *Using the omega index for evaluating abstractive community detection*, in: Proceedings of workshop on evaluation metrics and system comparison for automatic summarization, 2012, pp. 10–18.

[21] M. E. Newman, M. Girvan: *Finding and evaluating community structure in networks*, Physical review E 69.2 (2004), p. 026113, doi: 10.1103/PhysRevE.69.026113.

[22] Q. Ni, J. Wang, Z. Tang: *Degree and betweenness-based label propagation for community detection*, Journal of Combinatorial Optimization 49.2 (2025), p. 21, doi: 10.1007/s10878-024-01254-3.

[23] G. Palla, I. Derényi, I. Farkas, T. Vicsek: *Uncovering the overlapping community structure of complex networks in nature and society*, nature 435.7043 (2005), pp. 814–818, doi: 10.1038/nature03607.

[24] C. Pizzuti: *Ga-net: A genetic algorithm for community detection in social networks*, in: International conference on parallel problem solving from nature, Springer, 2008, pp. 1081–1090, doi: 10.1007/978-3-540-87700-4_107.

[25] U. N. Raghavan, R. Albert, S. Kumara: *Near linear time algorithm to detect community structures in large-scale networks*, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics 76.3 (2007), p. 036106, doi: 10.1103/PhysRevE.76.036106.

[26] A. Sarswat, V. Jami, R. M. R. Guddeti: *A novel two-step approach for overlapping community detection in social networks*, Social Network Analysis and Mining 7.1 (2017), p. 47, doi: 10.1007/s13278-017-0469-7.

[27] H.-W. Shen, X.-Q. Cheng, J.-F. Guo: *Quantifying and identifying the overlapping community structure in networks*, Journal of Statistical Mechanics: Theory and Experiment 2009.07 (2009), P07042, doi: 10.1088/1742-5468/2009/07/P07042.

[28] F. Wang, T. Li, X. Wang, S. Zhu, C. Ding: *Community discovery using nonnegative matrix factorization*, Data Mining and Knowledge Discovery 22.3 (2011), pp. 493–521, doi: 10.1007/s10618-010-0181-y.

[29] J. Xie, B. K. Szymanski, X. Liu: *Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process*, in: 2011 ieee 11th international conference on data mining workshops, IEEE, 2011, pp. 344–349, doi: 10.1109/ICDMW.2011.154.

[30] H. Xu, Y. Ran, J. Xing, L. Tao: *An influence-based label propagation algorithm for overlapping community detection*, Mathematics 11.9 (2023), p. 2133, doi: 10.3390/math11092133.

[31] M. Yan, C. Guoqiang: *Label propagation community detection algorithm based on density peak optimization*, in: 2021 17th International Conference on Computational Intelligence and Security (CIS), IEEE, 2021, pp. 80–84, doi: 10.1109/CIS54983.2021.00025.

[32] J. Yang, J. Leskovec: *Overlapping community detection at scale: a nonnegative matrix factorization approach*, in: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp. 587–596, doi: 10.1145/2433396.2433471.

[33] S. Yazdanparast, M. Jamalabdollahi, T. C. Havens: *Linear time community detection by a novel modularity gain acceleration in label propagation*, IEEE Transactions on Big Data 7.6 (2020), pp. 961–966, doi: 10.1109/TBDATA.2020.2995621.

[34] W. Zhang, R. Shang, L. Jiao: *Large-scale community detection based on core node and layer-by-layer label propagation*, Information Sciences 632 (2023), pp. 1–18, doi: 10.1016/j.ins.2023.02.090.