Community Detection Algorithm Implementation And Evaluation

Pengyuan Jin
1024041112
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—With the rapid development of the big data era, community detection algorithms have become indispensable tools in network analysis and social network mining. This study focuses on exploring the implementation and evaluation of several well-known community detection algorithms, including KL, GN, FN, LPA, SLPA, COPAR, Louvain, and LFM algorithms. These algorithms are tested on a single dataset to analyze their behavior under controlled conditions. The paper provides a detailed discussion of each algorithm. Using the Python programming language, we conduct experiments to assess their efficiency. Particular attention is paid to understanding how these algorithms perform in identifying community structures within the given network topology and how stable and accurate their results are under the constraints of a single dataset.**

The experimental results reveal differences in the performance of these algorithms when applied to the specific dataset, shedding light on their adaptability to network properties. By focusing on a single dataset, the study is able to provide precise insights into the feasibility of these algorithms in handling specific data structures within the broader context of big data. The findings emphasize both the strengths and limitations of each algorithm, offering valuable guidance for researchers and practitioners aiming to apply community detection techniques to real-world network analysis tasks. This research contributes to a deeper understanding of community detection algorithms and their application in social network analysis, providing practical insights for leveraging these methods in both academic and industry settings.

***Index Terms*—Community Detection Algorithms, Network Topologies, Social Network Mining, Modularity.**

I. Introduction

In recent years, complex networks have gained widespread attention as an interdisciplinary research field, with an increasing number of problems being effectively represented through complex network theory. Complex networks have become an important tool for problem-solving. Community partitioning originated from graph partitioning in computer science and hierarchical issues in sociology. Since Girvan and Newman introduced the concept of community structure in complex networks, community detection has become a hot topic in this field. In 2002, Girvan and Newman proposed a community partitioning algorithm based on community splitting—the GN algorithm [1]. By repeatedly removing edges with the highest edge betweenness, the GN algorithm can clearly display the hierarchical structure of a network. However, the GN algorithm has a drawback: it lacks an effective termination condition, which results in uncertainty about which level of partitioning is optimal. To address this issue, Newman and Girvan introduced a new metric to measure the strength of community structure—the modularity function Q in 2004. They argued that if a network has a clear community structure, the probability of connections within communities should be greater than the average connectivity of the corresponding random network. The larger the difference, the more pronounced the community structure. By introducing the modularity function, the GN algorithm is able to return the community partition corresponding to the maximum modularity value, thus improving the algorithm. Although the GN algorithm provides precise community partitioning results, its time complexity is relatively high. For a network with n nodes and m edges, its time complexity is $O(m^2n)$, which limits its application in large-scale networks. In contrast, Newman proposed the FN algorithm [2], a community discovery algorithm based on modularity optimization. Unlike the GN algorithm, the FN algorithm is an agglomerative community discovery algorithm. It initializes each node as an independent community, and at each step, merges two communities to maximize the increase in modularity (or minimize its decrease). This process continues until only one community remains. The FN algorithm returns the community structure corresponding to the maximum modularity value as the final community division result. Compared to the GN algorithm, the FN algorithm significantly reduces complexity, with a time complexity of $O(mn)$. However, with the advent of the big data era, the rapid growth of complex network scales has limited the applicability of the FN algorithm to networks of such scales. Therefore, there is an urgent need for algorithms that can discover large-scale community structures in a short time. In this context, Blondel et al. proposed the Louvain algorithm [3], a community discovery algorithm based on modularity optimization.

Identify applicable funding agency here. If none, delete this.

The Louvain algorithm rapidly obtains community structures through hierarchical clustering and local optimization. Blondel et al. used the Louvain algorithm to process a complex network with 118 million nodes, and the experimental results showed that the Louvain algorithm completed the community structure detection in 152 minutes. The time complexity of the Louvain algorithm is O(m), close to linear, making it one of the most widely used algorithms in community discovery.

In addition to the Louvain algorithm, other algorithms such as SLPA [4], KL [5], LPA [6], COPAR [7], and LFM [8] have also been widely applied, each algorithms with unique characteristics and applications. With the continuous improvement of community partitioning theory, community discovery technology has gradually been applied to many fields. Data mining through community partitioning has become a hot research direction. This paper first introduces several classic community partitioning algorithms and compares their ideas, time complexity, and other aspects from a theoretical perspective. Then, the paper compares the performance of these algorithms on real datasets from a practical perspective. Using the karate-club dataset for experiments, the paper constructs a complex similarity network using the Spearman correlation coefficient to measure similarity and applies community discovery algorithms for community partitioning. Finally, the paper uses time efficiency and modularity as evaluation metrics and compares the experimental results.

II. Related Works

Community detection is a critical topic in the study of complex networks, garnering significant attention and driving the development of various algorithms to address this challenging problem. This section review the relevant research and summarize key contributions from the literature.

A. Girvan-Newman Algorithm

The Girvan-Newman algorithm is one of the pioneering methods in community detection. Its core idea is based on the concept of edge betweenness, which measures the importance of an edge as a "bridge" in the network. By iteratively removing edges with high betweenness values, the algorithm gradually reveals the hierarchical structure of communities within the network. While the algorithm has a solid theoretical foundation and effectively demonstrates the hierarchical organization of networks, its computational complexity is relatively high due to the need to calculate edge betweenness, limiting its applicability to large-scale networks. The Girvan-Newman algorithm has laid a significant foundation for the development of subsequent community detection methods and holds substantial research value.

B. Fast Newman Algorithm

The Fast Newman Algorithm (FN Algorithm) is a community detection algorithm based on modularity optimization, proposed by Newman as an improvement upon the Girvan-Newman Algorithm (GN Algorithm). Unlike the edge-removal strategy of the GN Algorithm, the FN Algorithm adopts an agglomerative approach. Its core idea is to iteratively merge communities by maximizing the increase in modularity (or minimizing its decrease). Initially, each node is treated as an independent community, and in each step, the two most optimal communities are merged until only one community remains. The FN Algorithm then returns the community division corresponding to the maximum modularity. Compared to the GN Algorithm, the FN Algorithm significantly reduces the time complexity from $O(m^2n)$ to O(mn) , thereby improving its applicability. However, with the rapid growth of network sizes, even the FN Algorithm struggles to efficiently handle ultra-large-scale complex networks, which has driven the demand for more efficient community detection algorithms.

C. Louvain algorithm

The Louvain algorithm is an efficient community detection method based on modularity optimization. The algorithm works by iteratively optimizing the modularity function to uncover community structures. It operates in two main stages: In the first stage, nodes are locally moved to different communities to maximize the modularity; in the second stage, these communities are treated as "super nodes," and a new network is built, after which the first stage is repeated to refine the community partition. The Louvain algorithm has a time complexity close to linear, O(m), making it highly efficient for large-scale networks. Experiments have shown that the algorithm not only performs well in terms of runtime efficiency but also produces community partitions with high modularity, making it one of the most widely used algorithms in the field of community detection.

D. Label Propagation Algorithm

LPA is a simple yet effective algorithm used for community detection. It identifies community structures in a network by propagating labels between nodes. LPA starts with each node having a unique label, and in each iteration, a node adopts the most frequent label among its neighbors as its new label. If there are multiple labels with the same frequency, one is chosen randomly. This process continues until the algorithm converges or reaches a specified number of iterations.

1) Speaker-Listener Label Propagation Algorithm: SLPA is a label propagation-based community detection algorithm suitable for discovering overlapping communities in networks. Unlike traditional label propagation algorithms, SLPA simulates the process of information propagation in the network, where each node plays two roles: "Speaker" and "Listener." In each iteration, a node first acts as a Speaker, propagating its label to neighboring nodes, which then act as Listeners, receiving the labels and selecting the most frequent label as their new label. Through repeated label propagation, the labels of nodes converge, forming the community structure in the network. SLPA effectively detects overlapping communities and does not rely on predefined network structures. Instead, it adaptively finds community boundaries through label propagation, making it highly flexible and efficient, particularly for large-scale networks. However, the results of SLPA may

have some randomness, and sometimes unstable community partitions can be produced.

2) Kernighan-Lin algorithm: KL algorithm is a classic graph partitioning method widely used in community detection and graph partitioning problems. The algorithm optimizes graph partitioning by minimizing the cut-set of edges in the graph, and was initially proposed by Kernighan and Lin in 1970. The basic idea of the algorithm is to divide the graph's nodes into two subsets and iteratively swap nodes to reduce the number of edges between the two subsets. The specific steps involve randomly dividing the nodes into two subsets, calculating the cut-set of edges for the current partition, and swapping nodes between the two subsets, choosing the node pairs that reduce the cut-set the most, until no further reduction of the cut-set edges can be made. Although the KL algorithm can effectively optimize graph partitioning, its computational complexity is high, particularly when handling large graphs, which limits its application in large-scale networks. Nonetheless, the KL algorithm remains the foundation of many modern graph partitioning algorithms and plays an important role in various fields.

3) Community Overlap Propagation Algorithm: COPRA is an overlapping community detection algorithm based on label propagation. Similar to SLPA, COPRA also allows nodes to have multiple labels, but its characteristic is that each node is assigned a pair of values (c, b), where c is the community identifier and b is the membership coefficient indicating the degree of belonging to the community. However, COPRA may yield unstable results in some cases due to its high randomness.

E. LFM algorithm

LFM is an algorithm that can detect both overlapping communities and hierarchical structures. The method is based on the local optimization of a fitness function, and community structure is revealed by peaks in the fitness histogram. The resolution can be adjusted through a parameter, allowing for the investigation of different hierarchical levels of organization.

III. Algorithm

A. Construction of Complex Networks

Complex networks are an essential tool for studying the relationships among components within complex systems, widely applied in natural, social, and technological domains such as social networks, biological networks, transportation systems, the internet, and power grids. A complex network consists of nodes and edges, where nodes represent fundamental units of a system, and edges signify the relationships between these units. The connectivity of nodes is described by "degree," often exhibiting a scale-free property, where a few nodes have exceptionally high connectivity (hub nodes). Complex networks typically exhibit the small-world effect, meaning that most nodes can be connected through just a few intermediaries, and they may also display community structures that reflect clusters of functional modules. The dynamic and diverse nature of complex networks makes them vital in understanding information dissemination, epidemic spread, gene interactions, and transportation optimization. Research methods include graph theory, statistical physics, computer science, and dynamical modeling. Current studies focus not only on static structures but also on exploring the dynamic evolution of networks, multi-layer interactions, and functional optimization, addressing challenges such as processing massive-scale network data and designing efficient and stable systems, thereby advancing the understanding of complex systems and fostering innovative applications.

In complex networks, methods for defining relationships are primarily based on node connectivity and network structural properties. Common approaches include using topology and weight information to describe associations between nodes. For instance, the adjacency matrix is the most basic representation, directly indicating whether a connection exists between nodes. Furthermore, degree centrality can be used to measure the direct associations of a node with other nodes. In weighted networks, weight information plays a crucial role, such as reflecting strong associations between nodes through edge weights or node strength (i.e., the total weight of all edges connected to a node). Path-based metrics also provide important measures, such as calculating the probability of reaching one node from another using random walk models or evaluating potential relationships through resource allocation methods. For dynamic networks, mutual information or dynamic activity over time series can be employed to measure correlations.

B. Community Detection Algorithms

Community discovery, also referred to as community detection or graph clustering, involves identifying and unveiling closely connected groups of nodes within a network or graph. In this context, nodes represent entities (such as individuals or organizations), while edges signify relationships or interactions between them. The purpose of community discovery is to expose meaningful and relatively independent substructures, or communities, within a broader network.

The core concept behind community discovery is that nodes within a community are more likely to be interconnected compared to their connections with nodes outside the community. Communities often represent functional units, social groups, or modules within various complex systems. Recognizing these communities aids in understanding the underlying structure and organizational principles of a network.

A variety of algorithms and methods have been developed to perform community discovery, each with distinct assumptions and techniques. These methods aim to segment the network into subsets of nodes that exhibit higher internal connectivity than their connections to the rest of the network. Metrics such as modularity Q, which evaluates the strength of the identified community structure, are commonly used to assess the effectiveness of these algorithms. In essence, community discovery is a pivotal aspect of network analysis, social network mining, and the study of organizational patterns in complex systems.Modularity measures the difference between the actual internal connection strength of nodes within com-

munities and the expected connection strength under random distribution, with higher values indicating a more meaningful community division.

The modularity Q is defined as follows:

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

In this formula:

- m is the total number of edges in the network, i.e.,$[m = \frac{1}{2} \sum_{i,j} A_{ij}]$
- $[A_{ij}]$ is the adjacency matrix, where $[A_{ij} = 1]$ if there is an edge between node i and node j, and $[A_{ij} = 0]$ otherwise.
- $[k_i]$ and $[k_j]$ are the degrees of nodes i and j, respectively, i.e.,$[k_i = \sum_j A_{ij}]$ and $[k_j = \sum_i A_{ij}]$
- $[\delta(c_i, c_j)]$ is the Kronecker delta, which is 1 if nodes i and j belong to the same community $[(c_i = c_j)]$, and 0 if they belong to different communities.

Next, we will examine notable community discovery algorithms, including the KL algorithm, GN algorithm, FN algorithm, LPA, SLPA, COPRA algorithm, Louvain algorithm, and LFM algorithm. This discussion will focus on analyzing the strengths and limitations of each algorithm, providing a detailed overview of their features and applications in uncovering network community structures.

IV. Experimental

A. Dataset

The Karate Club dataset is one of the classic examples widely used in social network analysis, particularly for demonstrating and testing community detection algorithms. This dataset originates from a university karate club and records the social relationships and interaction patterns among club members. It consists of 34 nodes, with each node representing a member of the club, and 78 edges representing the social connections between members, reflecting their interactions in daily activities. What makes the dataset particularly interesting is that it not only describes simple connections between members but also captures the community structure that can emerge in a social network.

The social structure of the Karate Club is significant in algorithmic research. The relationships among members exhibit a certain degree of fragmentation, making it an ideal case for studying how this fragmentation changes over time or in response to social events. For example, the social dynamics within the club may change after specific events, with members forming new subgroups or reorganizing existing social circles. This transformation provides an excellent experimental platform for community detection algorithms, allowing researchers to test how sensitive algorithms are to changes in network structure and community evolution.

The dataset is frequently used in the teaching and research of graph theory, social network analysis, and community detection, especially for analyzing smaller networks. Due to its moderate size and intuitive structure, it helps researchers gain a deeper understanding of the mechanisms behind community formation and its evolution in social networks. The Karate Club dataset has become one of the most widely used benchmark datasets for testing various community detection algorithms in academic research.

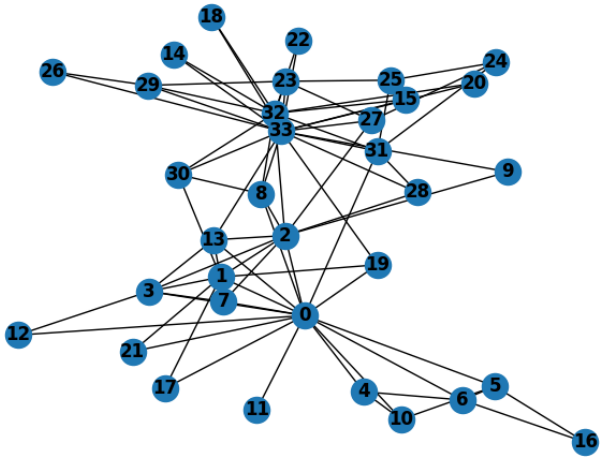


Fig. 1: Karate-Club dataset visualization

In addition, the dataset also demonstrates how the connections between nodes in a social network influence the formation of community structures, providing an intuitive perspective on social interactions. For researchers using different algorithms, this dataset not only holds educational value but also aids in validating the effectiveness and robustness of those algorithms.

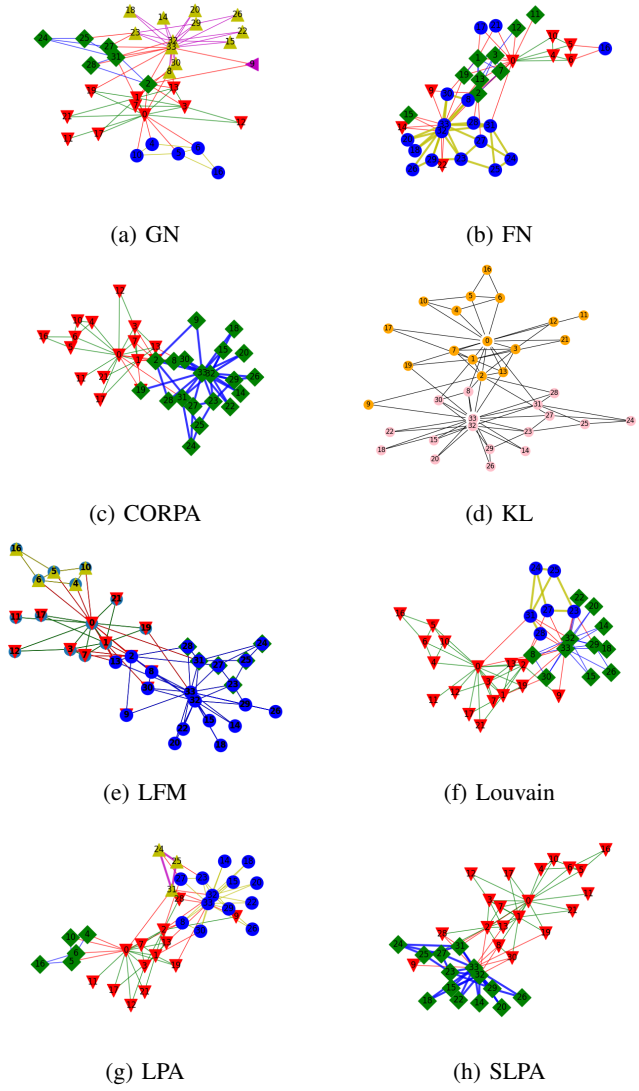
B. Evaluation metrics

The paper uses modularity and execution time as the evaluation criteria.Modularity values range from [-1, 1], usually being positive. A higher value indicates a more reasonable community assignment, stronger internal connections within communities, and a more significant community structure. Negative values suggest weaker internal connections, and nodes are more likely to be randomly distributed, resulting in lower modularity. Values close to zero imply that the community structure of the network is similar to that of a random network, leading to lower modularity.

1) Result: This paper implements the algorithm using Python 3.8. This study applied the KL, GN, FN, LPA, SLPA, COPAR, Louvain, and LFM community detection algorithms to the Karate-Club dataset. The resulting community classifications are shown in Figure 2.

C. Analysis of Results

In this study, we applied several community detection algorithms to analyze the Karate Club dataset, which consists of 34 nodes and 78 edges representing social relationships among the club members. The algorithms used in this analysis include GN, FN, SLPA, Louvain, KL, LPA, COPAR, and



(a) GN (b) FN (c) CORPA (d) KL (e) LFM (f) Louvain (g) LPA (h) SLPA

Fig. 2: Result

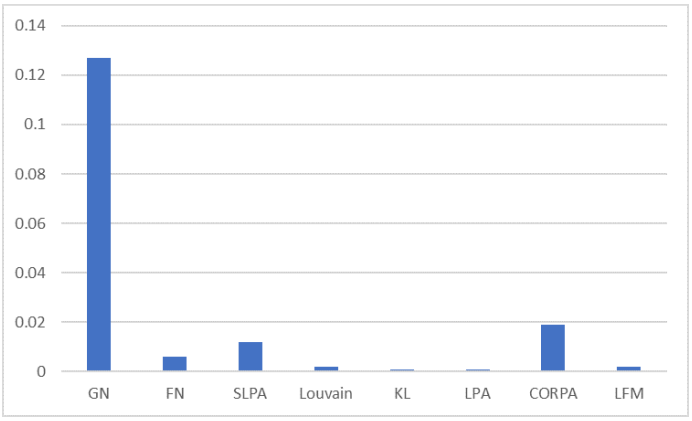


Fig. 3: Time Cost

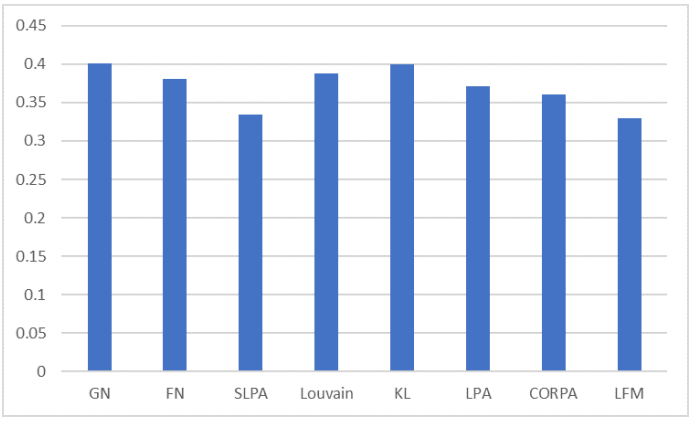


Fig. 4: modularity

LFM. We evaluated each algorithm based on its running time and modularity values to assess its performance on this dataset.

First, regarding the running time of the algorithms, the KL algorithm proved to be the most efficient, taking only 0.001 seconds. It is followed by LPA and Louvain, which took 0.001 seconds and 0.002 seconds, respectively. These algorithms are capable of quickly identifying community structures and are well-suited for larger networks. In contrast, the GN and COPAR algorithms took considerably longer, with running times of 0.127 seconds and 0.019 seconds, respectively. This indicates that these algorithms have higher computational costs when applied to the Karate Club dataset, likely due to their more complex iterative processes.The time costs are shown in Figure 3.

Modularity is an important metric for evaluating the quality of community detection, reflecting the strength of internal connections within communities and the degree of separation between communities. In this experiment, the KL algorithm performed the best, with a modularity value of 0.401, suggesting it was particularly effective at identifying the true community structure within the Karate Club. The GN algorithm had a modularity value of 0.401, slightly lower than KL, but still demonstrated strong performance in community detection. The FN and Louvain algorithms had modularity values of 0.381 and 0.388, respectively, which are a bit lower than KL and GN, but still produced reasonable community divisions.

The SLPA algorithm had a modularity value of 0.334, which is lower compared to other algorithms, indicating that its community detection was more scattered and less precise. The LPA and LFM algorithms had modularity values of 0.371 and 0.33, respectively, showing weaker community structures. Despite their efficiency in terms of running time, these algorithms did not perform as well in terms of modularity.The modularity are shown in Figure 4.

Overall, the KL and GN algorithms not only had higher efficiency in terms of runtime, but also exhibited the best modularity, indicating they were more capable of identifying the true community structure in the Karate Club dataset. The Louvain algorithm, although slightly slower, achieved modularity close to the optimal value and is suitable when better community detection is required. In comparison, the SLPA and

LFM algorithms showed lower modularity, suggesting they may be more suited for more complex or irregular networks, and might not perform as well on simpler social networks.

In conclusion, these algorithms display varying performances across different metrics. KL and GN are ideal for cases where accuracy is important, even though they require longer computational time. Louvain, LPA, and COPAR algorithms offer better speed, making them ideal for large-scale networks or real-time processing scenarios. SLPA and LFM, though more efficient, exhibited lower modularity, and could benefit from further improvement.

V. CONCLUSION

In this study, various community detection algorithms—GN, FN, SLPA, Louvain, KL, LPA, COPAR, and LFM—were evaluated using the Karate Club dataset. The focus was on comparing their performance in terms of computational time and the quality of the identified community structures, with modularity used as the key metric.

The algorithms demonstrated notable differences in both execution time and modularity. Some algorithms achieved high computational efficiency while still delivering good community detection results, making them ideal for large-scale or time-sensitive applications. On the other hand, algorithms that produced higher modularity values, indicating stronger internal community cohesion, required significantly more computational time. This highlights a typical trade-off between time efficiency and community quality.

Interestingly, some of the faster algorithms still provided competitive modularity scores, suggesting that quick execution does not always compromise the quality of community detection. However, slower algorithms tended to reveal more accurate community structures, making them more suitable for tasks where precision is more important than speed.

Overall, this study emphasizes the need to balance computational efficiency with the quality of community detection. While faster algorithms can perform well in many cases, slower, more accurate methods may be preferable in scenarios where detailed community structures are crucial. This analysis offers insights into the strengths and limitations of each algorithm, guiding the selection of the most suitable method based on specific requirements, such as network size, time constraints, and the desired quality of community detection.

REFERENCES

[1] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, p. 066133, 2004.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[4] J. Xie, B. K. Szymanski, and 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.

[5] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

[6] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, p. 036106, 2007.

[7] S. Gregory, "Finding overlapping communities in networks by label propagation," *New journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[8] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New journal of physics*, vol. 11, no. 3, p. 033015, 2009.