Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 2, pp. 173–180, DOI 10.12694/scpe.v24i2.2102

EVALUATING THE IGRAPH COMMUNITY DETECTION ALGORITHMS ON DIFFERENT REAL NETWORKS

PARITA OZA, SMITA AGRAWAL, DHRUV RAVALIYA, AND RIYA KAKKAR*

Abstract. Complex networks are an essential tool in machine learning and data mining. The underlying information can help understand the system and reveal new information. Community is sub-groups in networks that are densely connected. This community can help us reveal a lot of information. The community detection problem is a method to find communities in the network. The igraph library is used by many researchers due to the utilization of various community detection algorithms implemented in both Python and R language. The algorithms are implemented using various methods showing various performance results. We have evaluated the community detection algorithm and ranked it based on its performance in different scenarios and various performance metrics. The results show that the Multi-level, Leiden community detection algorithm, and Walk trap got the highest performance compared to spin glass and leading eigenvector algorithms. The findings based on these algorithms help researchers to choose algorithms from the igraph library according to their requirements.

Key words: Community detection, igraph, Multi-level, Walk trap, Leiden

1. Introduction. In computer science, network theory is a remarkable field that uses various techniques to utilize the vast scale of graphs consisting of numerous nodes and patterns connected in a network. Network theory helps in modeling computer networks, biological neural networks, traffic networks, protein networks, etc. One significant feature of these networks is community structure, i.e., densely connected components networks. The most crucial task in network analysis involves detecting communities in a network. We may have millions of nodes and edges in a colossal network connected in a quiet complex manner [17]. For example, nowadays, people use various online social networks such as Twitter, Facebook, and Instagram, which connect massive numbers of people forming a complex network [11]. So, it becomes a tremendous effort to detect communities in such networks. For that, many research work worldwide have been discussed and compared considering the community detection algorithms in which each of them associates various techniques for community detection [3] [18] [4]. Many research works have utilized machine learning [15] and deep learning networks [16] in community detection. For example, artificial networks [13] [14] or generating benchmark networks [10] can be considered with their implementation of various algorithms. However, most of the research works did not consider the performance, modularity, and coverage of the network [24] [5]. Therefore, we have evaluated the different algorithms implemented in the igraph library on different sizes of real networks. igraph is a set of tools for generating, altering, and analyzing graphs and networks. The techniques are commonly used in academic network science research. Further, we have applied the igraph library algorithms on real networks and ranked them based on their performance measures. We present the comparison of various igraph community detection algorithms considering the factors such as modularity, coverage, and performance to evaluate the impact of the algorithms on the network. We have computed the final rank considering the mean rank of all the possible community detection algorithms to evaluate the performance of the network. In nutshell, Our main research objective is to present a comparative analysis of the various algorithm on different datasets of sparse to dense networks and their applicability hence we utilized algorithms pertaining to the same library.

1.1. Research Contributions.
- We present the analysis of different igraph community detection algorithms such as Multi-level, Leiden, and Walk trap on different real networks.
- We discuss the various community detection algorithms implemented in the igraph library.

Department of Computer Science and Engineering, Nirma University, Ahmedabad, India (18bce195@nirmauni.ac.in, 21ftphde56@nirmauni.ac.in, parita.prajapati@nirmauni.ac.in,smita.agrawal@nirmauni.ac.in)

Table 1.1: Abbreviations for the community detection algorithms

EB	Edge Betweenness Centrality Measure
LE	Leading Eigenvector
LP	Label Propagation
IM	Map of information
SG	Spin-Glass
LC	Leiden algorithm based on optimization of modularity
FG	Fast-Greedy
ML	Multi-level Modularity Optimization
WT	Walk-Trap

- We present a comparative study of the various algorithm on different dataset of sparse to dense network and it's applicability .
- Finally, the performance results have been analyzed considering the modularity, performance measure, and coverage of community detection algorithms.

1.2. Paper Organization. The rest of the paper is structured as follows: Section 2 presents summary of the evaluated algorithms. In section 3 we discuss various datasets utilized in this work. Section 4 presents the methodology we adopted for evaluation. Section 5 presents the result and discussion. We finally end with conclusion in section 6.

2. The evaluated algorithms. The igraph library implements various community detection algorithms, which are discussed in the next section. In the algorithms, V is considered the set of vertices/nodes in the network, and E is considered the set of edges.

Edge betweenness centrality measure. This algorithm is called edge_betweenness in the igraph library. It can work on directed and weighted edges and can also handle multiple components in the network. It was initially proposed by Girvan and Newman in 2002 [8]. Later, in 2004, they presented a new version using the modularity measure, which is a method that estimates how modular network partitions in a given network. The main idea is to calculate the betweenness centrality for all edges and gradually remove the ones with the highest value to form a dendrogram [8]. The time complexity of the algorithm is $O(VE^2)$.

Leading Eigenvector. This algorithm is called the leading eigenvector and it can handle multiple components. This algorithm was proposed by Newman *et al.* [12] in 2006 to find optimal modularity using eigenvectors. The graph is separated into two segments in each step so that the separation generates a considerable rise in modularity. The split is decided by assessing the modularity matrix leading to the eigenvector and a halting condition that prohibits tightly related groups from being divided further. The complexity of algorithm can be computed as $O(V^2 + E)$.

Label Propagation. In the igraph library, this method is known as label-propagation and works with weighted edges. Raghavan *et al.* [20] first proposed it in 2007, in which every node in this algorithm is given a unique label at the start. Further, the algorithm involves iteration, re-assigning labels to nodes so that each node receives the label that its neighbor most frequently uses. The procedure ends when each node's label is one of the most common labels in its neighborhood. Communities are connected subgroups of nodes with the same labeled neighbors. The complexity of the algorithm can be calculated as $O(E + V)$.

Map of information. This algorithm is known as InfoMAP in the igraph library and works on weighted and directed edges. In 2008, Rosvall *et al.* [22] proposed the algorithm based on information-theoretic principles. With the help of random walks, it builds a grouping that provides the shortest description length for a random walk on the graph. The description length is measured by the expected number of bits per vertex required to encode the path of a random walk[8]. The complexity of the algorithm is $O(V * (E + V))$.

Spin Glass. In the igraph library, this algorithm is known as spin glass, and it can function with both directed and weighted edges and numerous components. Reichardt *et al.* [21] proposed this method in 2006 by introducing the technique based on the Potts model. In this model, each particle/vertex can be in one of the multiple spin states, and the interactions between the particles/edges decide which vertices prefer to be in

Table 3.1: Summary of datasets

Database –>	**Zachary's Karate Club**	**Football Network**	**HEP-TH Network**
Nodes	34	35	8400
Edges	78	118	15800
Density	0.139037	0.198319	0.000450686
Maximum degree	17	19	50
Minimum degree	1	1	0
Average degree	4	6	3
Average clustering coefficient	0.570638	0.338986	0.441964
Number of triangles	135	351	39900

the same spin state and which prefer to be in other spin states. The spin configuration reduces the spin glass's energy, and further spin states of the particles define the communities at the end.

Leiden algorithm based on optimization of modularity. The igraph library name for this algorithm is Leiden, and it works on both weighted and unweighted edges. In 2019, Traag *et al.* [23] proposed the Leiden optimization algorithm, which resembles the Multi-level algorithm in terms of functioning, except it is faster and yields better results. It can help with modularity and the Constant Potts Model, unaffected by the resolution constraint. If we consider the sparse graph for computing the complexity of the algorithm, it is found to be linear.

Fast-Greedy. This algorithm is known as fast greedy in the igraph library and it can work on weighted edges and handle multiple components. In 2004, Newman proposed this algorithm for detecting communities through a hierarchical approach rather than betweenness centrality; it is completely based on the modularity measure [7]. Later, Clauset *et al.* [7] suggested an optimized greedy version of Newman's proposed algorithm, which uses a more efficient data structure and an improved version of the Modularity measure. The complexity of algorithm can be determined as $O(VElogV)$.

Multi-level Modularity Optimization. This algorithm is termed multi-level modularity optimization in the igraph library and works on the weighted edges. This algorithm was proposed by Blondel *et al.* [6] in 2008, which resembles the fast greedy algorithm, but at each level of the dendogram, it considers local modularity, where a node is merged with the local neighbor to achieve the highest contribution to modularity. The complexity of algorithm is linear and can be calculated as $O(E)$.

Walk-Trap. This type of algorithm can work with weighted edges. Pons *et al.* [19] reportedly introduced it in 2006 to utilize the features of random walks. Moreover, there is a high probability of random walks in the same community. The complexity of the algorithm can be determined as $O(EV^2)$.

3. Datasets. We utilized three datasets in our work; the Karate network, the football network, and the Hep-Th network. This dataset represents a real network scenario. Zachary's Karate Club is being used for more minor test cases. It is based on Zachary's 1977 model of a friendship network between 34 karate club members at a US university with 78 interrelationships [25], which can be referred to as the Karate Network. The football network [8] is the American football college team dataset and has medium larger test cases. The dataset, HEP-TH (high energy physics-Theory) [9] is a large network consisting of details of authors and related papers submitted to the high energy physics-Theory category. We present a summary of this dataset in table 3.1.

4. Methodology for Evaluation. The metrics considered for evaluation are modularity, time, and coverage. The modularity measurement Q and the running time t returned by the algorithms are used for the evaluation. Alternatively, they can be arranged based on the Q/t ratio. According to the different running times of algorithms, a sigmoid function is being used to rescale the t values to a smaller interval and enhance the weight of the modularity metric in the ranking process. The above associations can be represented as follows:

$$S(x) = \frac{1}{(1 + e^{-x})} \tag{4.1}$$

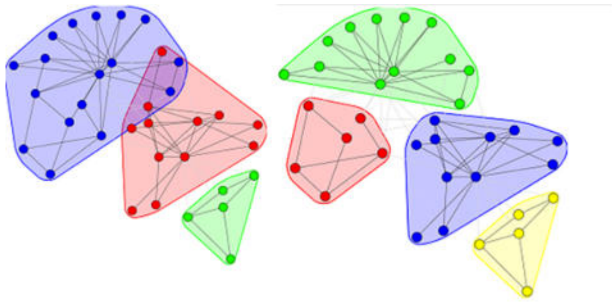

Fig. 4.1: Communities detected by the Leiden and Multi-level detection algorithm for Karate Network.

Table 4.1: Different igraph's community detection algorithm performance and results for the Karate Network.

Algorithm	**C**	**Q**	**T**	**P**	**Rank**
EB	5	0.40	1.99 ms	0.80	4
LE	4	0.39	6.98ms	0.78	5
LP	2	0.35	991µs	0.71	8
IM	3	0.40	6.98 ms	0.80	3
SG	4	0.41	190ms	0.76	6
LC	3	0.40	996µs	0.80	2
FG	3	0.38	998µs	0.76	7
ML	4	0.41	999 µs	0.83	1
WT	5	0.35	995µs	0.70	9

The modularity Q can be calculated with the help of the igraph method, i.e., self graph modularity. It can be evaluated based on the performance measure P, which can be mentioned as follows:

$$P_k = \frac{Q_k}{Sig(t_k)} \tag{4.2}$$

where Q_k is the modularity measure and t_k is the running time of algorithm k.

Coverage can be considered to detect the communities close to the desirable communities, which can be calculated with the help of partitions. Each partition's coverage is defined by the ratio of inter-communities edges to the total number of edges in the graph. The intra-community edges are those formed by joining a pair of nodes in the same partition block. In an ideal community, each node is connected to every other node of all the edges of the graph within clusters that lead to the coverage of one [2, 1].

The final rank can be calculated considering the average of the partial ranks algorithms obtained across all the test cases. This helps us to compare and rank the overall results of the algorithms.

5. Result and Discussion. The simulation has been performed considering the various community detection algorithms along with their parameters. The results of the Karate Network have been analyzed as shown in Table 4.1 in which C denotes the amount of detected community. It can be observed that multi-level and infomap achieve the same modularity; however, multi-level community detection yields better performance. While the Label propagation algorithm achieves the lowest modularity and the spin glass algorithm achieves the highest modularity. Further Fig. 4.1 shows a visual representation of the community detected by the Multi-level and Leiden community detection algorithm for Karate Network. Then, Fig. 4.2 and Fig. 4.3 visualize the community detection by Multi-level and Leiden community detection algorithm for Football and Hep-Th Network, respectively.

In the next scenario, as shown in Table 4.2, different community detection algorithms have been analyzed

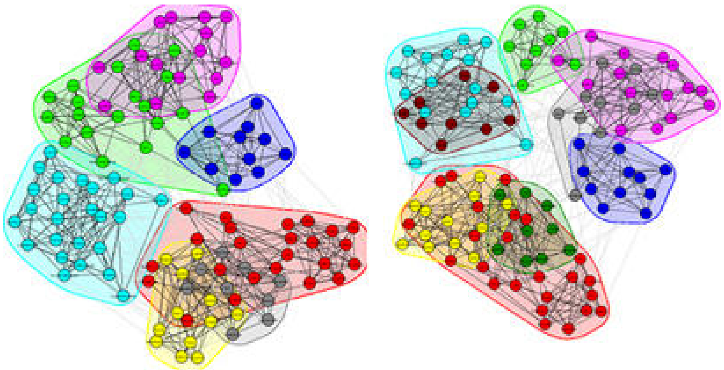

Fig. 4.2: Communities detected by the Leiden and Multi-level detection algorithm for Football Network

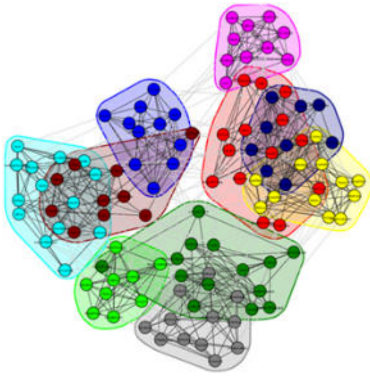

Fig. 4.3: Communities detected by the Leiden and Multi-level detection algorithm for Hep-Th Network

Table 4.2: Different igraph's community detection algorithm performance and results for Football Network.

Algorithm	**C**	**Q**	**t**	**P**	**Rank**
EB	10	0.60	234 ms	1.18	6
LE	8	0.48	23.9 ms	0.97	8
LP	10	0.60	987 µs	1.20	3
IM	13	0.55	18 ms	1.11	7
SG	11	0.60	736 ms	1.20	4
LC	11	0.60	998 µs	1.20	5
FG	-	-	-	-	9
ML	10	0.60	998 µs	1.21	1
WT	10	0.60	2 ms	1.20	2

for Football Network in which the multi-level detection algorithm yields the highest modularity leading to the highest rank. Compared to the previous scenario, Label propagation gives good results in scaling leading into a higher rank. In the final scenario, community detection algorithms have been analyzed for Hep-Th Network which is a large network comprising of various nodes and edges as shown in Table 4.3 in which multi-level detection algorithm results into improved modularity which results into the highest rank. However, it can effect the peformance of the network due to the high complexity of computation time as it contains large dataset of

Table 4.3: Different igraph's community detection algorithm performance and results for Hep-Th Network.

Algorithm	**C**	**Q**	**t**	**P**	**Rank**
EB	-	-	-	-	8
LE	1366	0.75	3.17 s	1.30	6
LP	1859	0.76	66.8 ms	1.52	4
IM	1847	0.76	6.46 s	1.16	7
SG	-	-	-	-	9
LC	2187	0.71	63.8 ms	1.42	5
FG	1411	0.81	151 ms	1.61	2
ML	1379	0.84	57.8 ms	1.69	1
WT	1411	0.81	703 ms	1.56	3

Table 4.4: Different igraph's community detection algorithm final rank on various network.

Algorithm	**Karate**	**Football**	**Hep-Th**	**Mean**	**Rank**
EB	4	6	8	6	6
LE	5	8	6	6.33	8
LP	8	3	4	5	4
IM	3	7	7	5.67	5
SG	6	4	9	6.33	9
LC	2	5	5	4	2
FG	7	9	2	6	7
ML	1	1	1	1	1
WT	9	2	3	4.677	3

Table 4.5: Different igraph's community detection algorithm coverage on various datasets

Algorithm	**Karate Network**	**Football Network**	**Hep-Th Network**
EB	0.69	0.71	0
LE	0.66	0.63	0.82
LP	0.85	0.71	0.78
IM	0.82	0.63	0.77
SG	0.73	0.69	0
LC	0.82	0.60	0.71
FG	0.75	0	0.90
ML	0.73	0.71	0.88
WT	0.58	0.70	0.85

networks which increases the overall complexity of the network. Label propagation detection continues to be good at scaling on the larger dataset. Similarly, Multi-level and Fast greedy seems to provide more modularity to the network improving the performance of the network. Nevertheless, we have considered small and large dataset to evaluate the performance of the various community detection algorithms considering Karate, Football, and Hep-Th network.

The results can be estimated by evaluating the final rank for all the community detection algorithms. Table 4.4 shows the comparison of final rank evaluated by considering the mean of rank in all the possible scenarios discussed. Further, Table 4.5 presents the coverage measure of various community detection algorithms. For the smaller dataset, Label propagation, Infomap and Ledian algorithm has one of highest results for denser

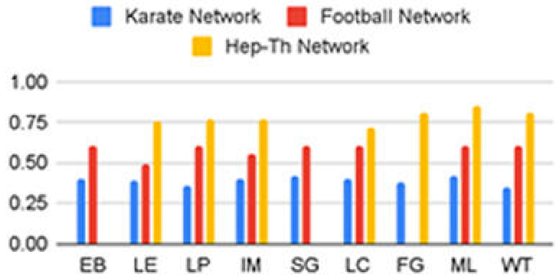

Fig. 4.4: Modularity of different algorithms

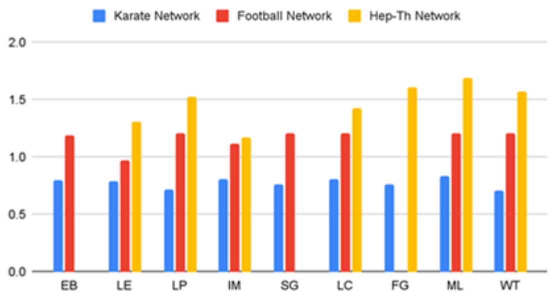

Fig. 4.5: Performance measure of the different algorithms

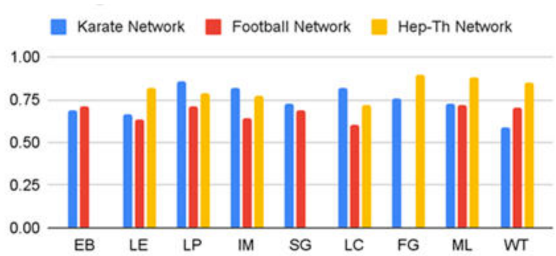

Fig. 4.6: Coverage of the different algorithms

partitions. While Fast greedy, Multi-level and Walktrap performs better for larger dataset. On the other hand,figures 4.4, 4.5 and 4.6 depicts the modularity, performance measure, and coverage of different algorithms for community detection.

6. Conclusion. In this paper, We have considered the community detection algorithms and estimated performance-based evaluation for their implementation in the igraph library. The evaluation was done based on their strength to detect communities in different types and sizes of datasets. The performance metrics to evaluate these algorithms used are modularity, performance measure, and coverage. Further, the algorithms are ranked based on performance measures. Results indicate that Multi-level performed best, followed by Leiden and Walk trap, which achieved lower performance in the smaller test case. Therefore, the evaluation of various igraph community detection algorithms has been estimated on the real network to show the impact of modularity, coverage, and performance measure.

REFERENCES

[1] S. Agrawal and A. Patel, *Csg cluster: A collaborative similarity based graph clustering for community detection in complex networks*, International Journal of Engineering and Advanced Technology, 8 (2019), pp. 1682–1687.

[2] S. Agrawal and A. Patel, *Clustering algorithm for community detection in complex network: a comprehensive review*, Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 13 (2020), pp. 542–549.

[3] S. Agrawal and A. Patel, *Sag cluster: An unsupervised graph clustering based on collaborative similarity for community detection in complex networks*, Physica A: Statistical Mechanics and its Applications, 563 (2021), p. 125459.

[4] S. E. Amrahov and B. Tugrul, *A community detection algorithm on graph data*, in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1–4.

[5] F. Berardo de Sousa and L. Zhao, *Evaluating and comparing the igraph community detection algorithms*, in 2014 Brazilian Conference on Intelligent Systems, 2014, pp. 408–413.

[6] 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, 2008 (2008), p. P10008.

[7] A. Clauset, M. E. J. Newman, and C. Moore, *Finding community structure in very large networks*, Phys. Rev. E, 70 (2004), p. 066111.

[8] M. Girvan and M. E. Newman, *Community structure in social and biological networks*, Proceedings of the national academy of sciences, 99 (2002), pp. 7821–7826.

[9] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters*, ACM transactions on Knowledge Discovery from Data (TKDD), 1 (2007), pp. 2–es.

[10] J. Leskovec, K. J. Lang, and M. Mahoney, *Empirical comparison of algorithms for network community detection*, in Proceedings of the 19th international conference on World wide web, 2010, pp. 631–640.

[11] H. S. Nalawala, J. Shah, S. Agrawal, and P. Oza, *A comprehensive study of "etcd"—an open-source distributed key-value store with relevant distributed databases*, in Emerging Technologies for Computing, Communication and Smart Cities: Proceedings of ETCCS 2021, Springer, 2022, pp. 481–489.

[12] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), p. 036104.

[13] G. K. Orman and V. Labatut, *A comparison of community detection algorithms on artificial networks*, in International conference on discovery science, Springer, 2009, pp. 242–256.

[14] G. K. Orman, V. Labatut, and H. Cherifi, *Qualitative comparison of community detection algorithms*, in International conference on digital information and communication technology and its applications, Springer, 2011, pp. 265–279.

[15] P. Oza, P. Sharma, and S. Patel, *Machine learning applications for computer-aided medical diagnostics*, in Proceedings of Second International Conference on Computing, Communications, and Cyber-Security: IC4S 2020, Springer, 2021, pp. 377–392.

[16] ———, *A drive through computer-aided diagnosis of breast cancer: a comprehensive study of clinical and technical aspects*, Recent Innovations in Computing: Proceedings of ICRIC 2021, Volume 1, (2022), pp. 233–249.

[17] A. Patel, P. Oza, and S. Agrawal, *Sentiment analysis of customer feedback and reviews for airline services using language representation model*, Procedia Computer Science, 218 (2023), pp. 2459–2467.

[18] N. Patel, P. Oza, and S. Agrawal, *Homomorphic cryptography and its applications in various domains*, in International Conference on Innovative Computing and Communications, S. Bhattacharyya, A. E. Hassanien, D. Gupta, A. Khanna, and I. Pan, eds., Singapore, 2019, Springer Singapore, pp. 269–278.

[19] P. Pons and M. Latapy, *Computing communities in large networks using random walks*, in International symposium on computer and information sciences, Springer, 2005, pp. 284–293.

[20] U. N. Raghavan, R. Albert, and S. Kumara, *Near linear time algorithm to detect community structures in large-scale networks*, Physical review E, 76 (2007), p. 036106.

[21] J. Reichardt and S. Bornholdt, *Statistical mechanics of community detection*, Phys. Rev. E, 74 (2006), p. 016110.

[22] M. Rosvall and C. T. Bergstrom, *Maps of information flow reveal community structure in complex networks*, arXiv preprint physics.soc-ph/0707.0609, (2007).

[23] V. A. Traag, L. Waltman, and N. J. Van Eck, *From louvain to leiden: guaranteeing well-connected communities*, Scientific reports, 9 (2019), pp. 1–12.

[24] N. Xiao, *Community detection and key nodes of complex technology exchange network*, in 2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT), 2016, pp. 101–104.

[25] W. W. Zachary, *An information flow model for conflict and fission in small groups*, Journal of anthropological research, 33 (1977), pp. 452–473.

Edited by: Katarzyna Wasielewska
Received: Feb 16, 2023
Accepted: Jul 9, 2023