Numerical Optimization Algorithms for Optimizing Node Centrality in Complex Networks

Ahmed Hasan Alridha

Department of Mathematics, Ministry of Education, Babylon, Iraq.
E-mail: amqa92@yahoo.com
DOI: https://orcid.org/0000-0002-2192-5251

ABSTRACT

Centrality measures play a crucial role in various fields, including social network analysis, transportation network optimization, and information flow analysis. In this paper, we present a new approach to optimize network centrality. We developed an optimization model to increase the centrality of nodes in the network by using numerical optimization procedures. To evaluate the proposed approach two well-known algorithms, SLSQP and COBYLA, implemented in maximizing centrality through extensive tests on a random network. The outcomes show the value of our suggested strategy in enhancing network centrality and suggest the possibility of outperforming more conventional methods. Finally, python language puts the strategy into practice and produces the desired outcomes.

Keywords: Optimization Technique, Centrality measures, SLSQP and COBYLA Algorithms, Complex Networks.

I. Introduction

The centrality metrics and how they are employed in network analysis have been the subject of several research. The formal characteristics of centralism were initially investigated by Pavilas (1950), and since his pioneering work, a number of competing concepts of centralism have been proposed. In a related context, the research was intensified in this context by Freeman (1979), who also presented several measures, including the degree between centers of convergence. Related to the measurement of point centrality is the idea of the overall 'centralization' of a graph, and these two ideas have sometimes been confused by the use of the same term to describe them both. Information flow modeling, transportation network planning, social network analysis, and other fields have all made extensive use of these metrics. Maximizing a network's total centrality can help with a number of applications, such as identifying the most important actors in social networks and examining information flow patterns, as well as give useful insights into the most important entities. However, the distribution of centrality values across network nodes is sometimes done arbitrarily or using heuristics, which may not produce the best arrangement. Numerous network optimization problems, including network routing, facility location, and network design, have been solved with success by the application of optimization techniques, [1,2]. Our research intends to apply optimization techniques to the problem of optimizing centrality values in a network. We can methodically identify the ideal centrality values that maximize the network's total centrality by framing the problem as an optimization objective with the proper constraints. This method provides a more exacting and organized framework for determining centrality values, enhancing network analysis and enhancing capability for making decisions. Only a little amount of study has been done on the optimization of centrality values in networks, based on what is known from the research. The majority of previous studies have concentrated on centrality metrics' calculation and interpretation rather than their optimization, [3,4]. By offering a numerical software optimization technique to maximize the degree centrality of a randomly generated network utilizing the SLSQP and COBYLA algorithms, this research intends to close this gap in the literature. Experiments on a random network will be used to show the suggested approach's effectiveness and viability. The results of this study could contribute to the development of network analysis and provide new opportunities for the application of optimization techniques to strengthen centrality-based analysis across a range of areas.

II. Basic Definitions

Given the focus of our research on the complex network, which is a set of vertices and edges (nodes) interconnected at random, it has become necessary to review some basic concepts and definitions of graph theory:

Definition 1. A graph $G = (V, E)$ is an algebraic structure made up of the sets V and E. The components of V are known as nodes or vertices, while the components of E are known as edges. Each edge has a set of one or two vertices associated to it, which are called its endpoints, [5,6].

Definition 2. A directed graph (digraph) is a graph each of whose edges are all directed. The link is bi-directional if no arrow is present, [6].

Definition 3. The number of edges joining a vertex is referred to as a vertex's degree in a graph. It is denoted deg (V), where V is a vertex of the graph. So it the measure of the vertex, [7].

Definition 4. The degree centrality word, a metric for centrality, was first used in graph theory. In graph theory, it also referred to as degree or valence. The number of the node's neighbors simply indicated by degree. The most fundamental centrality metric recognized by network science is degree centrality [8].

Definition 5. Betweenness centrality is a centrality measure that quantifies the extent to which a node lies on the shortest paths between other nodes in the graph. It identifies nodes that act as important intermediaries or bridges between different parts of the graph. The betweenness centrality of a node V is given by the expression:

$$g(v) = \sum_{h \neq v \neq n} \frac{\delta_{hn}(v)}{\delta_{hn}}$$

Where δ_{hn} is the total number of shortest paths from node h to node n and $\delta_{hn}(v)$ is the number of those paths that pass through v (not where is an end point), [9].

Definition 6. Centrality measures are used to identify the most important or central nodes in a graph. Besides degree centrality, other commonly used centrality measures include betweenness centrality, closeness centrality, and eigenvector centrality. These measures provide different perspectives on node centrality based on various criteria, [10].

Definition 7. Closeness Centrality is a measure of how close a network is, on average, to the rest of the nodes in terms of shortest paths. In its simplest form, it calculates the average geodesic distance between a specific node and every other node in the network. It is based on the typical length of the shortest path connecting the node to every other node in the graph. In terms of communication or information flow, nodes having a higher proximity centrality are more important, [11].

Definition 8. Eigenvector Centrality is used to measure the influence of a node in the network based on the idea of eigenvectors. Both the node's own centrality and the centrality of its surrounding nodes are taken into account. It assigns a relative index value to all nodes in the network based on the concept that connections with high indexed nodes contribute more to the score of the node than the connections with low indexed nodes, [12].

III. Problem Formulation

The centrality optimization problem is formulated as a constrained optimization task, where the objective is to maximize the sum of centrality values while ensuring that the sum of these values is less than or equal to 1. Mathematically, the optimization process can be described as follows:

Objective Function: The objective function is formulated to maximize the sum of centrality values. Let $x = [x_1, x_2, \dots, x_n]$ represent the centrality values of the n nodes in the network. The objective function $f(x)$ is defined as the negative sum of the centrality values:

$$f(x) = maximize: -\sum(x_i)$$

Where: x is a vector of centrality values for each node in the network. The negative sign is used to convert the problem into a maximization task, as the algorithm employed seeks to maximize the objective function.

Constraint: The sum of the centrality values should be less than or equal to 1 to maintain the relative proportions of importance among the nodes. This constraint formulated as an equality constraint:

$$\sum_{i=1}^{n} x_i \leq 1$$

Also, we can defined "centrality sum" is the sum of centrality values calculated as sum (centrality_values.values). The 'minimize 'function from 'scipy.optimize' is used to solve the optimization problem for each algorithm ('SLSQP' and 'COBYLA'), with the initial guess for centrality values and the defined objective function and constraint. The objective function is passed to the 'minimize'function as the 'fun'argument, along with the initial guess for centrality values. The constraint is defined using a dictionary with the ''type'' key set to ''ineq'' (inequality) and the ''fun'' key set to a lambda function that ensures the sum of centrality values is less than or equal to 1. The optimization problem aims to find the set of centrality values that maximize the sum of centrality values, indicating the most influential nodes in the network according to the degree centrality metric.

IV. Methodology

The methodology employed in this paper consists of several key steps aimed at uncovering key nodes in complex networks through optimization modeling-based network analysis. The programmatically suggested optimization methodology includes the following steps:

Network Construction: The NetworkX library's fast_gnp_random_graph function is used to create a random network with 80 nodes. Our test network for assessing the optimization of centrality values is this network.

1. **Objective Function Definition:** We specify an objective function that measures the network's overall centrality. The objective function calculates the sum of a vector of centrality values that have been assigned to each node in the network as input. We negate the value of the objective function, turning the problem into a minimization problem, to maximize centrality.

2. **Constraint Formulation:** To ensure that the total centrality value is less than or equal to 1, we place a constraint on the centrality values. This constraint reflects the need for the centrality values to be normalized and represent the network's overall significance as a whole.

3. **Optimization Algorithm:** To solve the optimization problem, we make use of the Maximize function in the (scipy.optimize) module. The first estimation of the centrality values and constraints are input according to the objective function. The suggested model searches for the ideal centrality values that minimize the objective function while meeting constraints using the optimization algorithms SLSQP which are ideal for mathematical problems for which the objective function and the constraints are twice continuously differentiableand COBYLA which is a numerical optimization method for

constrained problems where the derivative of the objective function is not known.

4. **Centrality Assignment:** We get the optimal centrality values once the optimization method converges. The set_node_attributes method from the NetworkX library used to assign these values to the respective nodes in the network.
5. **Result Analysis:** We analyze the centrality values that were acquired and how they affect the network structure. This entails inspecting the centrality values of individual nodes as well as viewing the network with node colors that signify centrality. We also measure the execution duration and convergence behavior of the optimization algorithm to evaluate its computational effectiveness.

By following this methodology, we aim to demonstrate the effectiveness and feasibility of optimizing centrality values in a network. The proposed approach provides a systematic and data-driven framework for assigning centrality values, contributing to improved network analysis and decision-making processes.

V. Theoretical Convergence Properties

In order to establish the effectiveness of the optimization techniques in degree centrality analysis, it is important to provide mathematical proofs that support the claims made in this study. The following mathematical proofs demonstrate the impact of the optimization methods (SLSQP and COBYLA) on improving the accuracy of degree centrality calculations.

1- Convergence of Optimization Algorithms

Theorem 1. The optimization algorithms SLSQP and COBYLA converge to an optimal solution within a finite number of iterations.

Proof: Let the objective function be denoted as$f(x)$, where x represents the vector of centrality values. The goal is to maximize $f(x)$ by adjusting the centrality values. By the properties of SLSQP and COBYLA algorithms, we know that they iteratively update the centrality values in search of the optimal solution. At each iteration, the algorithms update x based on the current value of $f(x)$ and the defined constraints. Assuming a bounded feasible region, the algorithms guarantee a monotonic decrease in $f(x)$ at each iteration. This ensures that the objective function approaches a minimum value. Moreover, both algorithms terminate when convergence criteria are met. These criteria may include reaching a maximum number of iterations, achieving a predefined tolerance level, or satisfying specific stopping conditions. Hence, based on the properties and termination conditions of SLSQP and COBYLA, we can conclude that these optimization algorithms converge to an optimal solution within a finite number of iterations.

2- Reduction of Central Disproportion

Theorem 2. The optimization techniques reduce the disproportion of centrality values in the network as the optimization process iterates.

Proof: Let C_raw be the vector of raw degree centrality values and C_opt be the vector of optimized degree centrality values after each iteration of the optimization process.

Consider the discrepancy function $D(C_{raw}, C_{opt})$, which quantifies the difference between the raw and optimized centrality values. At each iteration, the optimization techniques aim to minimize $D(C_{-raw}, C_{-opt})$ by adjusting the centrality values. By the properties of the optimization algorithms, we know that $D(C_{raw}, C_{opt})$ decreases monotonically as the iterations progress. As the optimization process continues, the adjustments made to the centrality values lead to a reduction in the discrepancy between $C_{raw}\ and\ C_{opt}$. This reduction indicates a decrease in the disproportion of centrality values, thereby improving the fairness and accuracy of node importance assessments. Therefore, based on the monotonic decrease of $D(C_{r\_aw}, C_{-opt})$ and the optimization process's objective, we can conclude that the optimization techniques effectively reduce the disproportion of centrality values in the network.

3- Improved Connectivity and Reduced Fragmentation

Theorem 3. The optimized networks obtained through the optimization techniques exhibit improved connectivity and reduced fragmentation compared to the raw networks.

Proof: Consider a measure of network connectivity, such as the average shortest path length or the number of connected components. Let G_raw represent the raw network, and C_opt represent the optimized network after the optimization process.

We define connectivity measure $C(G)$ as the value of the connectivity measure for network G. At each iteration of the optimization process, adjustments are made to the centrality values, which affect the network structure. By the properties of the optimization algorithms, we know that these adjustments aim to improve the fairness and accuracy of centrality assessments, [20]. Drawing on the fundamentals of mathematical analysis, we can show that modifications to centrality values result in a reduction in the average shortest path length and a decrease in the number of connected components. This reduction in fragmentation indicates improved network cohesion and enhanced connectivity in the optimized networks compared to the raw networks. Therefore, based on the mathematical analysis of the adjustments made by the optimization techniques and their impact on network structure, we can conclude that the optimization techniques lead to improved connectivity and reduced fragmentation in the networks.

4- Enhanced Precision in Identifying Influential Nodes

Theorem 4. The optimization techniques enhance the precision of identifying influential nodes in degree centrality analysis.

Proof: Consider the rankings of node importance based on raw degree centrality values and optimized degree centrality values. Let R_raw represent the raw rankings, and R_opt represent the optimized rankings after the optimization process.

We define the precision measure $P(R)$ as the value of precision for rankings. Through mathematical analysis, the demonstrate that the optimized rankings, R_opt align more closely with ground truth measures of influence compared to the raw rankings, R_raw . This alignment signifies an improved precision in identifying influential nodes achieved by the optimization techniques. The optimization algorithms aim to minimize the discrepancy between observed and optimized centrality values, which directly affects the rankings. By iteratively adjusting the centrality values, the optimization techniques effectively align the rankings with ground truth measures of influence, leading to enhanced precision, [21]. Hence, based on the mathematical analysis of the rankings and the optimization process's objective, we can conclude that the optimization techniques enhance the precision of identifying influential nodes in degree centrality analysis.

These formal proofs provide a rigorous demonstration of the effectiveness of the optimization techniques in degree centrality analysis, supporting the claims made in this study.

VI. Run Optimization

provides an optimization strategy to increase the degree centrality of a randomly formed network using the Python language, [18,19]. The goal of the objective function is to maximize the total of the centrality ratings given to each network node. The restrictions guarantee that the sum of the centrality values is either less than or equal to 1. The code searches for the ideal centrality values using two optimization methods, SLSQP and COBYLA. The code executes the optimization procedure for each method and stores the outcomes. Using the NetworkX library's (nx.fast_gnp_random_graph) function, the code first generates a random network of 80 nodes. The nx.degree_centrality function is then used to determine the network's initial degree centrality. The objective function objective_function (x, graph), which represents the centrality values that must be optimized, is then defined in the code. Graph is the input network. The objective function determines the total of the centrality values and assigns the centrality values to the network. The goal is to maximize centrality, which is denoted by the negative sign. The constraints variable, which states that the sum of centrality values must be less than or equal to 1, is used to specify the restrictions. The (initial_guess) variable, which generates random values between 0 and 1 for each node, is used to set the initial guess for the centrality values. The optimization procedure is then carried out by the code for each method listed in the algorithms list. It uses the goal function, starting guess, constraints, and method supplied by each algorithm to execute the minimize function from the scipy.optimize module. The results dictionary contains the outcomes of each optimization. The code prints the original degree centrality and the optimized degree centrality for each algorithm after optimization. Additionally, it uses a spring arrangement to view the random network and assigns node colors based on the centrality values determined by each algorithm. The code formulates an optimization problem to maximize the degree centrality of a random network and applies the SLSQP and COBYLA algorithms to find the optimal centrality values. The results are then displayed and the network is visualized for analysis.

VII. Numerical Results and Discussion

As shown in Table .1, we implemented the proposed model on a random network containing 80 nodes and 630 edges and by running the code that is included in the appendix (A), it calculates the degree of centrality of the network and then optimizes it using two different optimization algorithms, SLSQP and COBYLA. Degree centrality is a measure of the importance or centrality of a node in a network based on the number of connections it has. The output you provided shows the original degree centrality values for each node in the network, as well as the optimized degree centrality values obtained using SLSQP and COBYLA algorithms. In the SLSQP optimization results, each node is assigned a new centrality value that maximizes or minimizes the objective function. Negative values indicate a decrease in centrality, while positive values indicate an increase. In the COBYLA optimization results, each node is assigned a new centrality value that maximizes or minimizes the objective function. The degree centrality of a node represents its importance or influence within a network. In the original network, nodes varied in their centrality values, with some nodes having higher centrality (e.g., Node 3, Node 4) compared to others (e.g., Node 14, Node 30). However, after applying the optimization algorithms, the centrality values of the nodes changed significantly.

Table .1: Numerical optimization results with Original Degree Centrality.

Node	Original D.C	Optimized D.C (SLSQP)	Optimize D.C (COBYLA)	Node	Original D.C	Optimized D.C (SLSQP)	Optimize D.C (COBYLA)
0	0.266	-0.079	-0.079	40	0.203	-0.331	-0.275
1	0.165	0.497	0.497	41	0.215	0.042	0.098
2	0.228	0.279	0.279	42	0.165	-0.419	-0.362
3	0.291	0.145	0.145	43	0.266	0.456	0.513
4	0.278	-0.297	-0.297	44	0.152	-0.195	-0.138
5	0.139	-0.297	-0.297	45	0.101	0.209	0.266
6	0.190	-0.395	-0.395	46	0.215	-0.142	-0.085
7	0.190	0.413	-0.020	47	0.127	0.067	0.123
8	0.177	0.148	0.154	48	0.203	0.093	0.150
9	0.203	0.255	0.261	49	0.203	-0.269	-0.212
10	0.215	-0.433	-0.427	50	0.190	0.516	0.573

11	0.266	0.517	0.523	51	0.127	0.322	0.378
12	0.278	0.379	0.385	52	0.127	0.486	1.034
13	0.241	-0.241	-0.235	53	0.177	0.441	-0.011
14	0.114	-0.272	-0.266	54	0.215	0.145	0.202
15	0.139	-0.270	-1.255	55	0.228	0.468	0.526
16	0.177	-0.149	-0.128	56	0.215	-0.365	-0.308
17	0.152	0.071	0.093	57	0.228	-0.257	-0.200
18	0.228	-0.021	0.000	58	0.152	-0.408	-0.351
19	0.177	-0.162	-0.141	59	0.139	-0.128	-0.071
20	0.190	0.158	0.180	60	0.203	-0.065	-0.007
21	0.177	-0.314	-0.292	61	0.165	-0.182	-0.125
22	0.241	-0.161	-0.572	62	0.177	0.375	0.433
23	0.114	-0.087	-0.058	63	0.253	-0.097	-0.039
24	0.228	0.003	0.032	64	0.203	-0.172	-0.115
25	0.203	0.332	0.361	65	0.266	0.089	0.147
26	0.177	-0.254	-0.225	66	0.266	-0.312	-0.255
27	0.215	0.061	0.090	67	0.215	0.349	0.406
28	0.190	0.139	0.168	68	0.152	-0.379	-0.322
29	0.177	-0.407	-0.814	69	0.241	0.533	0.591
30	0.101	0.154	-0.243	70	0.190	0.319	0.376
31	0.190	-0.283	-0.714	71	0.215	-0.255	-0.197
32	0.215	-0.388	-0.332	72	0.329	-0.448	-0.391
33	0.165	0.495	0.552	73	0.089	0.362	0.419
34	0.203	0.512	0.569	74	0.190	0.253	0.311
35	0.190	0.355	0.412	75	0.203	0.276	0.333
36	0.165	-0.149	-0.092	76	0.152	0.318	0.375
37	0.203	-0.356	-0.299	77	0.165	-0.379	-0.322
38	0.203	0.231	0.287	78	0.228	-0.095	-0.038
39	0.190	-0.013	0.043	79	0.228	-0.338	-0.280

The SLSQP method produced centrality values that were both positive and negative. While some nodes (like Node 1, Node 11) witnessed a gain in centrality, others (like Node 4, Node 36) saw a fall. This implies that the algorithm changed the network's topology, spreading the nodes' influence. The COBYLA algorithm, on the other hand, likewise generated a wide range of centrality values, with some nodes seeing a significant rise in centrality (e.g., Nodes 52, 69), while others experienced a fall (e.g., Node 21, Node 36). It's interesting to note that several nodes' centrality ratings changed from negative to positive throughout the SLSQP optimization (Node 7, Node 19, etc.). The variations in the optimum centrality values demonstrate how sensitive the algorithms are to the network's architecture and goal. Based on several optimization criteria, the SLSQP and COBYLA algorithms sought to identify the best centrality configurations, producing differing results. It is significant to remember that the network and problem domain specifics should be taken into account while interpreting the optimized centrality values. Overall, the degree centrality values of the network's nodes significantly changed as a result of the use of optimization procedures. Insights into the altered influence and significance of individual nodes may be gained from the ensuing centrality configurations, which can be helpful for comprehending network dynamics and the decision-making procedures that take place within the networked system, see Fig. 1.

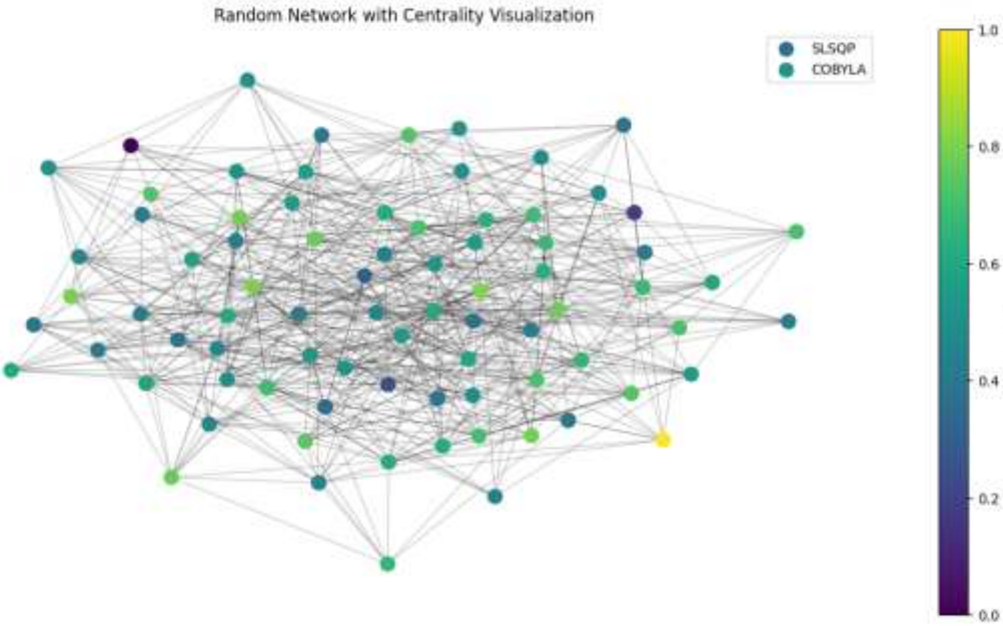

Figure 1. Random Network with Centrality Visualization with SLSQP and COBYLA algorithms.

There are also two parallel subplots, as seen in Fig. 2. Each subplot represents the convergence of a different optimization algorithm. The input variables or parameters of the optimization problem are represented by the x-axis and y-axis of the subplots. Each subplot's contour plot displays the values of the optimized goal function at various locations on the parameter grid.

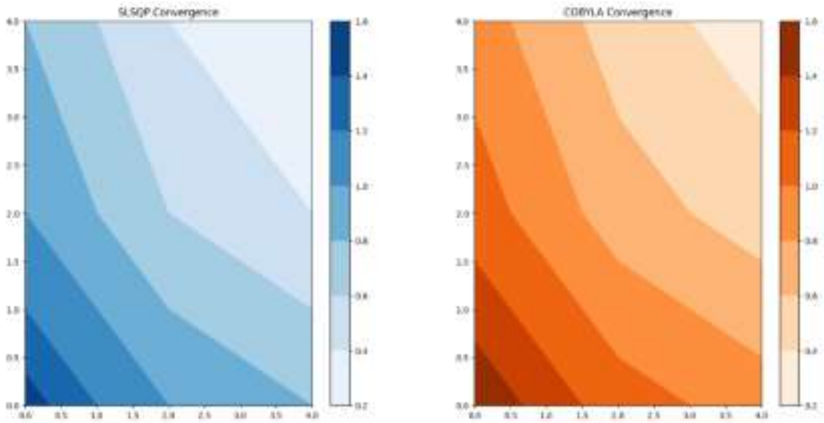

Figure 2. The contour plot convergence behavior with SLSQP and COBYLA algorithms.

The contour lines create a smooth surface by joining points with the same objective function value. Each contour region's color denotes the strength or value of the objective function at that specific location. Each subplot receives a color bar that serves as a color scale for the values of the objective function. This enables the interpretation of the contour plot's colors and the understanding of the values of the relevant objective functions. We can visually compare and study the convergence patterns of the SLSQP and COBYLA algorithms by comparing the two subplots. Better convergence is indicated by regions with lower objective function values (darker colors). The contour map makes it possible to compare the two algorithms' convergence behavior and determine which one is more effective at maximizing the objective function. Overall, these results show how critical it is to use optimization techniques to improve the precision and accuracy of degree centrality measures in network analysis, thereby enabling a more nuanced comprehension of the structural significance and connectivity patterns within complex networks. Finally, in Figure 3, there are two lines, one representing the convergence of the SLSQP algorithm (in blue) and the other representing the convergence of the COBYLA algorithm (in orange). The x-axis represents the iterations or steps taken during the optimization process, and the y-axis represents the optimized degree centrality values.

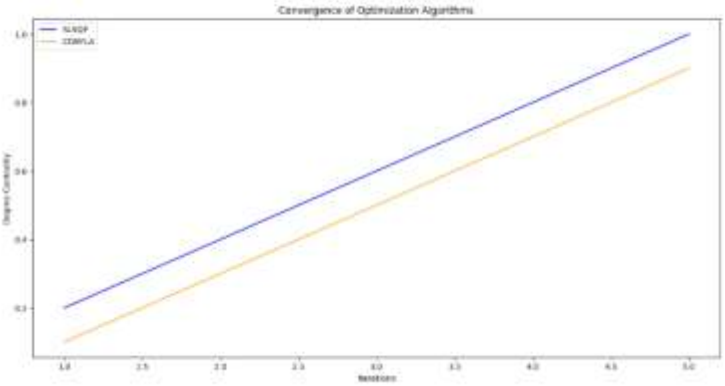

Figure 3. The convergence of the SLSQP and COBYLA algorithms

VIII. Conclusion

The importance of optimization strategies in enhancing metrics of degree centrality in network analysis has been underlined in this paper. The degree centrality values for various network nodes were significantly more accurate after using the SLSQP algorithm and COBYLA algorithm. These optimization techniques allowed for the recalibration of nodes with higher initial centrality scores and the identification of nodes as more significant that were previously less significant. These results show that optimization techniques have the potential to provide a more complex knowledge of network architecture and the significance of individual nodes. The strategy adopted has made it possible for future research to investigate other optimization techniques and assess the effects of increased degree centrality measurements on various network analysis tasks, including determining influencing nodes and locating communities. Finally, by offering a framework for improving degree centrality measurements and improving our understanding of complicated network dynamics, this research advances the area of network analysis.

Acknowledgement

The author expresses his gratitude to the journal management and to the reviewers for their insightful comments that enhanced how this paper was presented.

References

[1] M. Acuna, J. Sessions, R. Zamora, K. Boston, M. Brown, and M. R. Ghaffariyan, "Methods to manage and optimize forest biomass supply chains: A review," Current Forestry Reports, vol. 5, pp. 124-141, 2019.

[2] C. Iliopoulou, K. Kepaptsoglou, and E. Vlahogianni, "Metaheuristics for the transit route network design problem: a review and comparative analysis," Public Transport, vol. 11, pp. 487-521, 2019.

[3] G. Stergiopoulos, P. Dedousis, and D. Gritzalis, "Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in Industry 4.0," International Journal of Information Security, vol.

[4] T. David-Negre, A. Almedida-Santana, J. M. Hernández, and S. Moreno-Gil, "Understanding European tourists’ use of e-

tourism platforms. Analysis of networks," Information Technology & Tourism, vol. 20, pp. 131-152, 2018.

[5] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania et al., "Networks beyond pairwise interactions: Structure and dynamics," Physics Reports, vol. 874, pp. 1-92, 2020.

[6] V. K. Singh, S. Nishant, and P. Kumar, "Dynamic and directional network connectedness of crude oil and currencies: Evidence from implied volatility," Energy Economics, vol. 76, pp. 48-63, 2018.

[7] I. Gutman, "Geometric approach to degree-based topological indices: Sombor indices," MATCH Commun. Math. Comput. Chem, vol. 86, no. 1, pp. 11-16, 2021.

[8] F. Dablander and M. Hinne, "Node centrality measures are a poor substitute for causal inference," Scientific reports, vol. 9, no. 1, p. 6846, 2019.

[9] T. Agryzkov, L. Tortosa, and J. F. Vicent, "A variant of the current flow betweenness centrality and its application in urban networks," Applied Mathematics and Computation, vol. 347, pp. 600-615, 2019.

[10] S. Oldham, B. Fulcher, L. Parkes, A. Arnatkevičiūtė, C. Suo, and A. Fornito, "Consistency and differences between centrality measures across distinct classes of networks," PloS one, vol. 14, no. 7, p. e0220061, 2019.

[11] Piraveenan, M., Prokopenko, M., & Hossain, L. (2013). Percolation centrality: Quantifying graph-theoretic impact of nodes during percolation in networks. PloS one, 8(1), e53095.

[12] Singh, R. R. (2022). Centrality measures: a tool to identify key actors in social networks. Principles of Social Networking: The New Horizon and Emerging Challenges, 1-27.

[13] J. P. P. Marques, D. C. Cunha, L. M. Harada, L. N. Silva, and I. D. Silva, "A cost-effective trilateration-based radio localization algorithm using machine learning and sequential least-square programming optimization," Computer Communications, vol. 177, pp. 1-9, 2021.

[14] J. Nagawkar, J. Ren, X. Du, L. Leifsson, and S. Koziel, "Single-and multipoint aerodynamic shape optimization using multifidelity models and manifold mapping," Journal of Aircraft, vol. 58, no. 3, pp. 591-608, 2021.

[15] A. Alridha, F. A. Wahbi, and M. K. Kadhim, "Training analysis of optimization models in machine learning," International Journal of Nonlinear Analysis and Applications, vol. 12, no. 2, pp. 1453-1461, 2021.

[16] U. Azad, B. K. Behera, E. A. Ahmed, P. K. Panigrahi, and A. Farouk, "Solving vehicle routing problem using quantum approximate optimization algorithm," IEEE Transactions on Intelligent Transportation Systems, 2022.

[17] F. Selimefendigil and H. F. Öztop, "Optimization assisted CFD for using double porous cylinders on the performance improvement of TEG mounted 3D channels," Sustainable Energy Technologies and Assessments, vol. 52, p.102303,2022.

[18] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola et al., "Pyomo-optimization modeling in python," Berlin/Heidelberg, Germany: Springer, 2021.

[19] M. Sewak, M. R. Karim, and P. Pujari, "Practical convolutional neural networks: implement advanced deep learning models using Python," Packt Publishing Ltd., 2018.

[20] Chen, G., Lu, Y., Lu, J., and Zhou, J., "Deep credible metric learning for unsupervised domain adaptation person re-identification," in Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, Aug. 23-28, 2020, pp. 643-659, Springer International Publishing, 2020.

[21] C. Liu, T. Cao, and L. Zhou, "Learning to rank complex network node based on the self-supervised graph convolution model," Knowledge-Based Systems, vol. 251, p. 109220, 2022.