IJICTR
International Journal of Information & Communication Technology Research

Volume 11- Number 4 – Autumn 2019 (8 -20)

Development and Improvement of Network Reduction Algorithms for Multilayer Networks

Mahya Moheb
Department of Network Science and Technology
Faculty of New Sciences and Technologies
University of Tehran
Tehran, Iran
mahya.moheb@gmail.com

Sayyed Hamid Reza Ahmadi
Department of Network Science and Technology
Faculty of New Sciences and Technologies
University of Tehran
Tehran, Iran
hrahmadi@ut.ac.ir

Morteza Ebrahimi*
Department of Network Science and Technology
Faculty of New Sciences and Technologies
University of Tehran
Tehran, Iran
mo.ebrahimi@ut.ac.ir

Received: 3 April 2019 - Accepted: 3 August 2019

Abstract—**Given the complexity of today's networks, performing data analysis requires reducing the network's size into smaller manageable useful sizes. To the best of our knowledge, in the domain of multilayer networks, reducing the size of such networks while simultaneously preserving the features and the nature of the network has not been done before. This paper, for the first time, combines three separate single-layer network simplification methods to make a new method for reducing the size of multilayer networks in a way that preserves the fundamental features of the network. The three simplification algorithms are Path Simplification, Degree-based Node Selection, and Hair Reduction algorithms. A hybrid approach is used for combining these algorithms with modifications to support multilayer features. To reduce the multilayer network, these algorithms are applied to the network sequentially. Our proposed method is tested on four real-world datasets. Results of the comparison among the reduced and the original networks, show that the reduced networks maintain the main features while their analysis complexity is less than the original ones.**

Keywords-Network Systems; Shortest Directions; Network Reduction; Layered Networks; Social Networks.

I. INTRODUCTION

A. Definition

For the past decade, network analysis has been a useful tool for describing different systems, understanding their structure and analyzing their properties. However, the evolution of the web and the ability to store large amounts of data have increased the size of networked systems and as a result, increased their complexity. Network analysis and imaging algorithms seem impractical to deal with very large systems. Therefore, various methods have been proposed to simplify complex single networks. Simplification is a process that reduces the size of a network by reducing the number of nodes and edges. These methods are derived from graph theory and were originally developed for efficient graph storage and

* Corresponding Author

maintaining. As networks become more complex, simplification techniques help to clarify the network and perform more efficient analysis. In addition, it allows us to evaluate the difference between complete and reduced systems. Recently, network simplification has been widely considered from a large variety of perspectives. Some studies on simplifying specific networks are including simplifying social networks based on sustainability, scale-independent sampling or directional networks, estimating different features on social networks, sampling peer-to-peer networks using random steps. Streaming is concentrated by removing unnecessary links. Other studies in singular networks have attempted to observe network changes and features influenced by simplification such as clustering coefficient, degree distribution, community structure, spectral properties, or network connectivity [1] [2]. One of the simplest names for reducing network in scientific articles, especially in directional networks, is network simplification. Most of the networks that have important streaming use this term, but there is a concept that is very close to network reduction. Simplification is a process in which the size of a network is reduced by abating the number of nodes and edges. Calculation of flow in a network is one of the fundamental issues in graph theory and it has well-known applications in the real world. Its purpose is to find a stream of specific features that travel from the source node to the destination node using network edges with specific constraints such as limited capacity. Current issues are of great importance in the field of computer networks, hybrid optimization and transport. Many other key theory issues, such as the smallest two-way matching, can be formulated in terms of current flow. Due to the wide application of flow problems, it is important that the algorithms run smoothly. Because current networks can have very large sizes and the complexity of their algorithms is high, it is very clear that it will be decreased by removing the edges and nodes which have no effect on the flow paths. In general, there are six ways to simplify a single layer network:

1) A randomized method in which randomly uniform nodes are selected for a simplified network (RN).

2) A randomized method in which randomly uniform edges are selected for a simplified network (RL).

3) A randomized method based on the node's degree, where the probability of the selection of each node is based on the degree of that node (RD).

4) The first level search method from a randomly selected node is to search for the first level, and nodes are added to the simplified network (BF).

5) Balanced community method in which the nodes are merged according to the community's detection, so that it is put together for all nodes within a community (BP).

6) The cluster growth method in which a node is randomly selected as the kernel, and the nodes with less than the specified value are all added together as a node in the simplified network (CG) [1].

Despite the efforts described above, several open questions about simplifying complex networks remain, one of which is how we can apply network reduction in multilayer networks? In this paper, we try to answer this question by proposing a combined approach to reduce a multilayer network. We also evaluate the effectiveness of the proposed simplification process in four real-world networks of various sizes.

B. *Innovation*

During the past two decades, network science has provided many insights in natural, social, biological, and technological systems. However, real systems are often interconnected, with many interdependencies that are not properly captured by single-layer networks [3]. Multilayer and multiplex networks, which take into account different kinds of relations among the same set of nodes at the same time, are currently a hot research topic in network science. The main idea behind the investigation of high-dimensional network representations is that retaining full information about the structure of a system under study is often fundamental to fully understand its behavior. Indeed, multilayer networks have helped unraveling interesting structural properties in transportation systems and neuroscience, and have revealed qualitatively new emerging phenomena, including abrupt cascading failures, super diffusion, explosive synchronization, and hyper fast spreading [4]. In this paper, we have an idea that transforms a multilayer network into a singular one while preserving all of the links from all layers. We use a form of tagging to eliminate the links between nodes and make them independent from each other. To do this, we tag every link depending on its layer number. For example, if we have a network with 3 layers, after network aggregation and transforming it into a single layer, in the worst case, we will have 3 links (equal to the number of layers) between every two nodes. One of the links has tag 1 that means it was for layer 1, another has tag 2 meaning it was for layer 2, and the other has tag 3 meaning it was for layer 3. Using this tagging, we lose nothing from the original network. After that, when we study most of the previous works in single layer and dependent on their performance and how they preserve main network features, we choose the best three methods for network reduction in single layer networks and combine them together in two forms, sequentially and simultaneously, and then apply them to our flat multilayer networks.

II. Overview of Previous Works

So far, various networking technologies have been used to solve various problems. In the category of social networks and issues surrounding it, there are some things that we will briefly discuss.

A. *Community Detection in a Complex or Weightless Network with the Help of Network Reduction*

Detecting hidden communities is important for the analysis of complex networks. However, many algorithms have been designed for single layer networks (SLNs) while just a few approaches have been designed for multiplex networks (MNs) [5]. The presence of a contractual structure in complex networks of the real world is the focus of the attention of

researchers in many studies to find out the relationship between network topology and its performance. The best approach to identifying a modular structure is the modular quality optimizer function. This function is hardly computable polynomial computing. Detecting the community on large networks in ways that are discovered but unavailable or if it's possible but are not optimal. A definite way to reduce the size of the complex networks is to maintain their modularity [5]. It is not possible to apply exploratory algorithms to obtain optimal results, and in practice, computational complexity is extremely demanding. The question posed here is how much of the modularity of the network has been maintained? According to the network and more on the issue where the network is looking, the basis of the grouping will be different. So because the purpose of detecting the community in a reduced network is to replace the main network with the reduced one because of its ease of work and the less complexity of the calculation, therefore modularity preservation is a very significant issue during the downsizing method[6] [7].

B. *The Shortest Route in the Webserver Network by Simplifying the Network*

The idea of network simplification in these networks is a path-based approach in which the edges are eliminated by maintaining the quality of the best path between all pairs of nodes (but not necessarily all the best paths). For all graph models, randomly, the flow networks of this idea can be implemented. Connections between nodes are measured using the best path between them, and the edges are removed from the graph in a way that does not create problem for the best path. Reduced algorithms are classified with the idea of removing extra edges from two perspectives. In the first view, based on the search type this view is divided into two categories; the first category is the best overall route in which additional edges find the best paths between the endpoints of those edges are identified. The second category is a triangular search that searches for paths that contain two edges rather than identifying additional edges, so it cannot search for all edges excessive network eternal but is faster than the first one. In the second view, the classification is performed based on the type of edge deletion. Here the algorithms are divided into two categories. The first-order deletion is a repetitive one that works dynamically, a change in the graph occurs quickly and may affect to add or remove series of the edges. The result of this operation is a completely pruned graph but the result is not necessarily unique and is depending on which edges are processed. The second type of deletion is static, which takes the main graph and removes the extra edges one by one. By combining the categories of these two views, we have four types of algorithms: 1) static and general 2) repetitive and general 3) static and triangular 4) repetitive and triangular [8].

III. ACHIEVEMENTS

A. *Development of Reduction Methods for Monolithic Networks to Multilayer Networks*

Handling redundant and irrelevant features in high-dimension datasets has caused a long-term challenge for network anomaly detection. Eliminating such features with spectral information not only speeds up the classification process but also helps classifiers make accurate decisions during attack recognition time, especially when coping with large-scale and heterogeneous data [9]. As stated in this paper, our effort is to make it easier to communicate with the real world, and we know that in the real world, many of the systems we interact with them, are networked. But networking is a complex network involving different objects and different communications. Therefore, the analysis of these types of networks can be valuable because of the high volume and complexity of them, we have adopted the approach of reducing and simplifying the network, keeping in mind the features and structure of the main network and the reduced network being a similar but small sample of the main network. So we redefined three of the most efficient methods of network shrinkage and network simplification in single-layer networks and applied changes to them. Here are some of the achievements we were looking for in this paper:

• Modeling a multilayer network as a single-layer network by preserving all the edges and nodes and properties of each layer in this way that which edge and node belongs to which layer.

• Modeling the best three single layer reduction algorithms for application in multilayer networks, those algorithms are path simplification algorithm, degree-based node random selection algorithm and hair-reduction algorithm.

• Applying algorithms in sequential combination on four real-world datasets and reducing the number of nodes and edges of the network by maintaining the network structure.

• Evaluate the reduced network and the main network then compare the features and structure of the two networks, find the shortest path and other optimization features in them.

The main focus of this research is on solving the best and most efficient application problems in multilayer networks based on network reduction methods. Methods based on network shrinkage, network simplification, network aggregation, graph reduction, graph compression and summarization have been used in the past for both monolithic networks and a few number of multilayer networks, some of which are more efficient depending on the type of network and don't considered the multilayer nature of the network.

B. *Why those three algorithms are the best in single layer network*

We chose three algorithms that named path simplification algorithm, degree-based node random selection algorithm and hair-reduction algorithm. Here we want to say why we chose them from all of the methods were in single layer for network reduction. For the first algorithm, Path Simplification, we should say that we had three methods to simplify weighted graphs by pruning least important edges from them, the first one named Naive approach (NA) that was the simplest one. It first sorts edges by their weights in an ascending

order. Then, it iteratively checks the edge from the top of the sorted list, and prunes the one whose removal will not lead to disconnected components. The second one was the Brute Force approach (BF) which prunes edges in a greedy fashion. In each iteration, it picks the edge whose removal best keeps the connectivity and the third one was Path Simplification approach (PS) this method finds, for each edge, the best possible alternative path globally. It then prunes in each loop the edge with the largest lower bound of connectivity kept. So each of them prunes edges in a somewhat different way. The results of comparing between these three methods showed that each method had different ratio of connectivity kept from the other one. The best method was Path Simplification, it kept high connectivity for dense graphs up to approximately 90%. So we chose this method for participation in our combination method in multilayer. For the second algorithm, degree-based node random selection, we should say that in Simplification methods we had 6 methods that we said about them in the part A of introduction. The results of performance comparison of them revealed that there were several distinctions in the behavior of the simplification methods. RD and BF proved the best for preserving the local properties of networks, whereas for global properties, RN outperforms the other methods. However, RL and merging methods show the worst performance. These findings are consistent with the results of the study reported in [1], where RD had a better performance than RN and RL. So we chose RD for participation in our combination method in multilayer. For the third algorithm, hair-reduction, among all the methods that existed for size reduction in complex networks, hair-reduction algorithm preserved modularity. The direct consequence of its application is an improvement in computational cost, and then accuracy, of any heuristics designed to optimize modularity. We think that the idea of the exact reduction could be extended to other specific motifs in the network, although its analytical treatment can be more difficult. The reduced network is also an appealing concept to renormalize dynamical processes in complex networks (in the sense of real space renormalization).With this reduction it is plausible to perform a coarse graining of the dynamic interactions between the formed groups. So we chose hair-reduction for participation in our combination method in multilayer. All of these solutions are aimed at achieving efficiency, accuracy, faster speed or lower computational and space cost. It has also attempted to remove fewer important features from the network and to provide a more appropriate and less accurate estimate of the actual response. Each of these solutions is described in detail below. There are very few ways to reduce multilayer networks and there is rarely works to do, so there must be many ways to go about this. There are several methods used in monolayer networks, some of which are mentioned in the overview of past research.

C. *Reduce Each Layer Individually*

One general idea is that each of these methods is to reduce monolayer networks, so their application to each layer of a multilayer network should naturally lead to the reduction of that layer, so by reducing the layers of a multilayer network, we expect that the entire network to shrink. The idea is to reduce the number of single layers in one or more efficient ways and this will reduce the whole multilayer network. It is certain that this idea will be accompanied by many problems, such as the coordination of the layers with each other after applying a reducer method on them.

D. *Combining reducer Methods and their Combined Development on Multilayer Networks*

This is one of the most common ideas in research. Since each method has many strengths and weaknesses, the idea of combining several approaches together to cover for each other's weaknesses and strengthen to achieve better outcomes has always been a focus of research. But the idea in general is accompanied by complexities that if not do it careful, will not only result in a more credible response than in the past, but may also result in much poorer performance and a more inefficient response. Because sometimes methods are combined do not match their presuppositions or the nature of their work is inconsistent. For example, combining a method that performs well at speed with a method that is less error-prone than others does not necessarily result in a high-speed or low-error method. In addition, one study has shown that although the two methods of rank-based random selection and first-level search are both among the most effective methods of network simplification, combining these two methods in the hope of achieving better results with failure has resulted in a worse and more ineffective outcome. But for example, if we group edges in networks and omit linear edges, then grouping nodes or any other reduction or simplification method would seem reasonable and expect the result to be ineffective. So this method of synthesis is also an idea that, given its details and delicacies, hopes to achieve a better result. Of all the algorithms outlined, the three most efficient algorithms have been selected to apply individually and consecutively to multilayer networks, so that changes should be made to better utilize multilayer networks. These three algorithms are: path simplification algorithm, degree-based random node selection algorithm, and hair reduction algorithm. The path simplification algorithm focuses only on the pruning of the edges, while the second and third algorithms work on deleting and integrating the nodes. In the present study, these three methods have been applied with remarkable modifications to the four real multilayer datasets in combination. The following describes these methods and their modifications in detail.

IV. ALGORITHMS

A. *Path Simplification Algorithm*

There are some ways to simplify the weighted theoretical networks, which in this research is the most efficient of them, called the path simplification method for multilayer networks. Path simplification methods for networks that weigh the edges, are focused on finding the edges and nodes that do not affect the flow of the route from source to destination [10] [11]. This research is based on two hypotheses that the connection between nodes is measured by the best path between them (and the graph connection using the mean connection between all pairs of nodes). The grid-scale algorithms generate so-called routing networks by

removing the edges when there is a better path from the maximum q to the edge, where q is a parameter. Related neighborhoods [12], only connect the nodes that are close together. They are often made by a distance matrix, but also used to simplify the graph. In fact, they only use triangular relationships. The method that most closely relates to the approach we used here is the path-based simplification method that eliminates the edges that have no effect on the best path between pairs of nodes [13].

Algorithm: Path Simplification

Input: *A weighted graph* $G = (V, E), q$ *and* γ
Output: *Subgraph* $H \subset G$
$F \leftarrow E$
$n \leftarrow \gamma(|E| - (|V| - 1))$
{*Iteratively prune the edge with the largest* κ *value.*}
$M \leftarrow \emptyset$
for $r = 1$ *to* n ***do***
  $\kappa\_largest \leftarrow -\infty$
  $e\_largest \leftarrow null$
  for $e = \{u, v\}$ *in* F *and* $e\_ \in M$ ***do***
    Find path S *such that* $q(S) = C(u, v; F \setminus \{e\})$
    if $q(S) \geq q(\{e\})$ ***then***
      $\kappa \leftarrow 1$
      $F \leftarrow F \setminus \{e\}$
      break
    else if $0 < q(S) < q(\{e\})$ ***then***
      $\kappa \leftarrow \frac{q(S)}{q(\{e\})}$
    else
      $\kappa \leftarrow -\infty$
      $M \leftarrow M + e$
    end if
    if $\kappa > \kappa\_largest$ ***then***
      $\kappa\_largest \leftarrow \kappa$
      $e\_largest \leftarrow e$
    end if
  end for
  $F \leftarrow F \setminus \{e\ largest\}$
end for
Return $H = (V, F)$

This algorithm defined a graph connectivity function based on the best paths between all pairs of nodes which given the number of edges to be pruned, the problem is then to select a subset of edges that best maintains the overall graph connectivity. The path quality function has a natural recursive property. A rough semantic analysis of the removed edges indicates that few important edges were removed, and that the proposed approach could be a valuable tool in aiding users to view or explore weighted graphs. A simple way to determine how strong the two nodes are connected is to calculate the quality of the best path between the two nodes, so this algorithm defined the connection between the two nodes as follows:

$$C(u, v; E) = \begin{cases} max_{P \subseteq E: u \xrightarrow{P} v} q(P) & , \quad if\ p\ exist \\ -\infty & , \quad otherwise \end{cases}$$

This approach calculated the value of k for each edge. Finded the best alternative route for each ridge. And each time it repeated, it removed the edge loop that has the largest minimum for k to maintain the connection. It immediately removed edges with k equal to 1.

$$k(E, e) = \begin{cases} -\infty & , \quad C(u, v; E \setminus \{e\} = -\infty \\ \frac{C(u, v; E \setminus \{e\})}{q(\{e\})} & , \quad C(u, v; E \setminus \{e\} < q(\{e\}) \\ 1 & , \quad C(u, v; E \setminus \{e\} \geq q(\{e\}) \end{cases}$$

The M list hold items whose removal will cause the network connection to be lost. The temporal complexity was the innermost loop for finding the best path between two nodes, which was equivalent to $O((|E| + |V|)\log|V|)$. This loop was executed n times for $O(|E|)$ edge. Therefore, the temporal complexity of this algorithm was $O(n|E|((|E| + |V|)log|V|))$.

B. *Applying Path Simplification Algorithm on Multilayered Networks*

To apply this algorithm to multilayer networks, significant changes had to be made. The beginning was that we had to put all the layers of the network in one layer, but in the previous works that was done in multi-layer networks this integration of the layers was accompanied by the merging of the same edges between the two nodes and considering the weight of that edge as the average weight of the edges between the two nodes. This causes it to move away from the main network and its features. In this research, multiple graphs have been used to solve this issue. Multiple graphs make it possible to define multiple edges between two nodes and used a key assignment to each edge to distinguish between edges. Here a multi-layer network is created, then the nodes and edges of each layer are added to it and all the edges and nodes of the main multi-layer network are finally present in the reduced network. The same edges between two nodes are distinguished by the value of the key, where the key value is the number of the layer to which the edge belongs. Therefore, in our algorithm, the edge between two nodes that belonged to layer 1 and the edge between the same two nodes that belonged to layer 2 are seen in a distinct network. In this algorithm, Dijkstra's algorithm is used to find the shortest paths, but changes have been made in it. The weight of an edge in one network can be flowing, in another one can be distance and in another can be probability. Depending on what the weight of the edges are in the network, the quality calculation function in this algorithm is different. In the Dijkstra function, this quality calculation is performed for each edge and returns to the reduction simplification algorithm, where these values are evaluated according to the reduction instructions and will be removed if the conditions are met. In the path simplification algorithm, the procedure is such that it receives one percentage of the user, which means what percentage of the edges should be eliminated, or in other words, what percentage of the network should be reduced. So it multiplies this percentage by the number of possible edges to delete and find the number of edges to be removed. Each time it wants to select an edge to remove from the network, it removes one edge from all the edges and calculates quality with the help of a modified Dijkstra algorithm. In calculating the quality, it takes into account the quality of the presence of that edge as q_e and on the other hand the absence of that edge and the quality of the alternative path instead of it, as q_p. These two qualities are the output of the modified Dijkstra algorithm. The ratio of the two is then evaluated in the path simplification algorithm, if $q_p > q_e$ then the edge will undoubtedly be removed. If $0 < q_p < q_e$ then all the edges are checked and finally the edge with the largest k is removed which $k = \frac{q_p}{q_e}$.

C. Applying Random-Based Random-Node Algorithm to Multi-Layered Networks

Network simplification methods may be divided into two categories. In the first category, there are methods in which the reduced network is a random sample of the main network (random selection of nodes, random selection of edges, bullet sampling, random walk sampling) and second grouping of edges and nodes based on characteristics such as their distance from one another (The growth of the cluster is either the characteristics and characteristics of the nodes and edges and the structure of the communities). To compare these methods, two similarity criteria are considered together, one based on general characteristics and the other based on local characteristics. To compare these methods, two similarity criteria are considered together, one based on general characteristics and the other based on local characteristics. The general similarity characteristics criterion determines how far the general characteristics of the network are close to the simplified network. This proximity rate is calculated by Spearman correlation coefficient ρ [14]. As we said before, RD algorithm has the higher efficiency. In this method is that all nodes are arranged in order of degree, and then depends on their degree they candidate to remove, in such a way that the node with low degree has little chance to remove. In total, this deletion continues to the specified extent, as in the previous algorithm, the user's percentage specifies that how many nodes should be remove, in the other words, what percent of the network should be reduced. It then selects the node to delete from a list sorted by degree of nodes. If the main network is connected, it checks during the deleting. If it doesn't disconnect the network the node and its associated edges will be removed.

D. Applying Hair Reduction Algorithm on Multilayered Networks

In this method, the grouping of nodes is a substitute for a group of nodes in the main network with a node in the reduced network. Each group of groups from N to the node in the complex network *G* in *N′* is displayed up to the part by an *R* function. The weight between the two groups of nodes with the names *r* and *s*, represented by w'_{rs}, is composed of all the weights of the nodes in the two groups. The main attribute of the reduced network is to maintain the property of modulation. The modulation property of each part of the network is reduced with its equivalent in the main network. It should be noted that the complexity of this algorithm is less than O (n) because nodes are removed over time.

Definition 1: (Hair points) The points in which the nodes are connected to the network with just one edge. So, at a point of the hair, the node can be grouped with its neighbor node *k* if:

$$w_{ii} \leq \frac{w_i^{out} w_i^{in}}{2w}$$

And an autonomous edge for node k with a given weight in the following formula [7]:

$$w'_{kk} = w_{ii} + 2\, w_{ik}$$

The multi-layer net *(V, E)* network is given as input to the hair-drop algorithm and becomes a multiple network, and the reduction operation is performed and the output of this algorithm is a multiplied *G*-network. This output, along with an input taken from the user, in the name that determines how many percent of the nodes is eliminated, is given as input to the randomly selected node degree-based algorithm, then after the network-decreasing operation, which actually reduces both of the algorithms in It is done as an input to the route simplification algorithm.

E. Review on general procedure and modifications of our hybrid approach

The multilayer network turned to multi graph network using functions from Networkx library in Python programming and that multiplex network's edges have 4 components. The first and second components are the source and end nodes of the edge, the third component is the layer number to which the edge belongs, which is used as the edge key in the network and the last component is the edge's weight. Then hair reduction (HR) algorithm, degree-based random node selection algorithm (RD) and path simplification algorithm (PS) were applied to this multi graph network, respectively. The hair reduction algorithm consists of a loop that runs in order to the number of the nodes in the network and inside that loop, every node which has just one output edge or one input edge is selected for deletion and its edge is added to the neighboring node as a self-loop, this algorithm first applied to a copy network of the main one, if the connection of that copy network was maintained after reduction, then that node will be removed from the main network. The output reduced network of Hair Reduction algorithm entered as an input to the Degree-Based Node Random Selection algorithm. This algorithm take a value from user to determine what percentage of the nodes he wants to be removed or in the other words how percent should the network reduce. Then first of all it sorts the nodes by degree, then has a loop in it that runs to the number of nodes and in the loop, it starts deleting the node with the highest degree, but like the hair reduction algorithm, it first performs this deletion in the copied network, and if the network connection was maintained after deletion, it removes that node from the main network. Then the output reduced network of the Degree-Based Node Random Selection algorithm entered as an input to the path simplification algorithm. This algorithm also takes a value from the user at the beginning that determines how many edges should be removed. This algorithm consists of two main loops, the first loop is repeated to the number of edges specified by the user and in each of the execution of the first loop, one edge is removed, but the determination of which edge should be removed in the first loop is done in the second loop. The second loop is repeated to the number of all edges of the network, in each of the execution of this loop the desired edge is sent to the modified Dijkstra algorithm which is programmed by us and in the Dijkstra algorithm, investigate that if this edge will be removed, what the best and shortest path between the first and last nodes of this edge is. It then calculates the quality function in two cases: one is when the edge exists in the

network and the other is when the edge is removed and there is an alternative path instead of it. It then sends these two values as the output for the second loop of the path simplification algorithm. Then the loop compares the ratio between these two values and decides whether or not to remove the direct edge between the two nodes and examines the network connection that the deletion does not cause disconnection in network. In fig 1, the procedure of applying the proposed method in this paper is to illustrate in the form of flowchart. $net(v,e)$ is given as input to the Hair Reduction algorithm and converted to multiple network and the reduction operation is performed to it and the output of this algorithm which named G with an input that is taken from the user that determines what percentage of nodes to delete are given as an input to the Degree-Based Node Random Selection algorithm. After the reduction operation of these both algorithms resulting reduced network pass into last algorithm, Path Simplification, the operation of this algorithm applied to it and ultimately the final reduced network is obtained as the final output of our method.

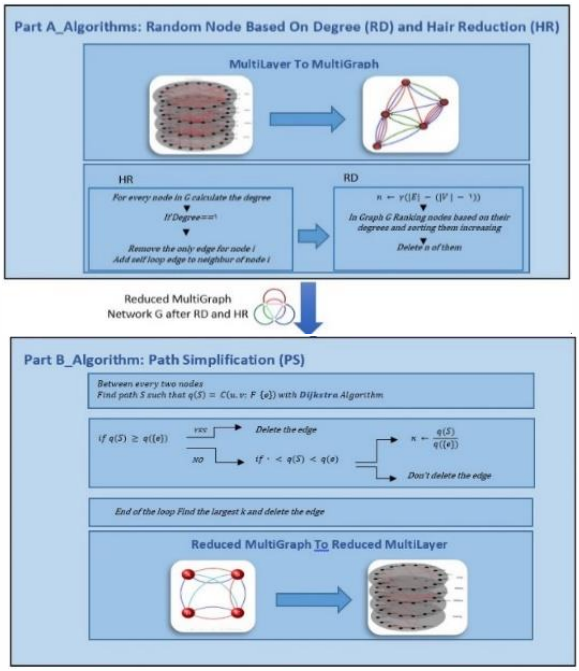


Figure 1. Flowchart Trend performance of Grade-Based Random Node Selection Reduction Algorithms, Hair Reduction and Path Simplification.

V. Experimental Steps

In this research, four sets of data have been used. All of them were voluminous multi-layered networks. These collections include the London Transportation Network, the Sampson Monks Network, the European Aviation Network and the Genetic and Protein Interaction Network. All of these algorithms programed with python language version 3.5.

A. *Why we chose these four datasets*

As we explained before our method must be applicable and conclusive for any type or at least most types of networks, including probabilistic graphs, flow graphs and distance graph and etc. So we try to choose one symbol of each type of network in our experiment so we have human relationship in Sampson network, transportation in London network and flowing of flights in European network which both of them are related about transportation in 2 ways flight and ground an investigation in this such of datasets is important and effective in the field of release which is the topic of the day for the transmission of diseases and positive and negative issues at the level of communities and countries. At the end we have relation between biology parameters in Genetic and Protein Interaction Network for example in a gene interaction network we have a set of genes (nodes) connected by edges representing functional relationships among these genes. These edges are named interactions, since the two given genes are thought to have either a physical interaction through their gene products, e.g., proteins, or one of the genes alters or affects the activity of other gene of interest, also we considered to choose networks in different sizes from big networks so we had Sampson network in the smallest one and the size get bigger in London network and bigger in European and the biggest in Genetic and Protein.

B. *Explain datasets*

Sampson Monks Network: Sampson (1969) recorded the social interactions among a group of monks while he was a resident as an experimenter at the cloister. During his stay, a political "crisis in the cloister" resulted in the expulsion of four monks so he started to investigate relations between the monks. The Sampson Monastery dataset consists of social relations among a set of 18 monk-novitiates preparing to enter a monastery. The data include a variety of relations, such as liking, esteem, influence and praising. So there is 18 nodes and 213 edges. The layers represent the social relationships among the 18 monks who were preparing to enter a temple. From this set of data we only consider positive relationships, which include four layers corresponding to relationships of liking, esteem, influence and praising. We have the simulation of this network in figure 2 part a.

London Transportation Network: Transport for London (TfL) is a local government body responsible for most aspects of the transport system in Greater London. Its role is to implement the transport strategy and to manage transport services across London. TFL have responsibility for London Underground, London Rail, and Surface Transport. Number of journeys on the public transport network by TFL reporting period, by type of transport. The public London transport data is broken down by underground, DLR, Overground. Docklands Light Railway is based on automatic passenger counts at stations. Overground is based on automatic on-carriage passenger counts. This dataset consists of three layers of underground, DLR, Overground. All of which are London's transportation routes. The network is unconnected, and has a total of 369 nodes and 441 edges, each containing 312, 83, 46 arrows in each layer, representing the density in the layers. The nodes in this network are stations, and the weights of the edges are the number of paths between

the stations [15]. We have the simulation of this network in figure 2 part b.

European Aviation Network: The European Aviation Network is the continent's fastest connectivity solution. Made possible by an alliance of European innovators, it's an unrivalled solution that provides passengers with a true-broadband service and helps airlines to benefit from new revenue streams, a range of operational savings and the fastest return on investment. This collection is a collection of thirty-seven unreformed, non-oriented layers, each layer belonging to an airline and includes all flights belonging to one company. The network has 450 knots and 3588 edges. The nodes represent the airports and the edges are the flights between them, thus the layers have the same nodes as the number of airports in Europe is fixed and constant. Because the network is non-exploded, therefore, the weight of all edges is equal to 1 [16]. We have 5 layers which are Ryanair airlines, Lufthansa airlines, Vueling airlines and British Airways and aggregate.

Genetic and Protein Interaction Network: Genetic interactions capture functional relationships between genes using phenotypic readouts, while protein-protein interactions identify physical connections between gene products. These complementary, and largely non-overlapping, networks provide a global view of the functional architecture of a cell, revealing general organizing principles, many of which appear to be evolutionarily conserved. Different types of inter-organic communications are considered in the biological reservoir collection series. A generic set of collections that maintains communication between proteins and genetics in humans and organized models. This collection now includes 720000 interactions. Which is supervised by a high-volume collections or personal studies of 41,000 articles. The Genetic and Protein Interaction Data Sheet that we used in this study includes 6 layers, each layer identifying the type of communication between nodes that are genetic and proteins, and the number of nodes is 3866 and the number of edges is 7874, which is a large network proportion [17] [18]. We have 6 layers in this networks which they are direct_interaction, physical_association, additive_genetic_interaction_defined_by_inequality, suppressive_genetic_interaction_defined_by_inequality, association colocalization. The general information of the 4 datasets used in the research, are shown in Table 1.

C. Evaluation Measures

In this section we will evaluate the proposed solution. The evaluation of this research is divided into two broad categories. The first category is to evaluate and compare the network with the main network in terms of the main characteristics of the network, such as network strength and the centrality of the nodes and the edges and the centrality of the proximity also the importance of the layers using the cluster state before and after reduction. The second category compares and evaluates the shortest paths found in the network and the importance of the layers and the number of switches between layers when finding all the shortest paths in the network, which is compared with other studies in order to find the shortest path in multi-layer networks. Considering that in the face of large networks, this research has used statistical methods such as uniform sampling, which definitely uses a method weaker and far from the actual network, with the principles of maintaining the nature and similarity to the main network of reduction operations [19] [20] but our method do it better and maintain the multilayers features.

TABLE I. THE GENERAL INFORMATION USED IS BASED ON THE NUMBER OF ACTIVE NODES IN EACH LAYER, THE NUMBER OF EDGES, THE NETWORK DENSITY AND THE SUM OF THE WEIGHTS IN EACH LAYER.

Datasets	Layers	Num. Of Node	Num. Of Edge	Network Density	Sum Of Weight
Sampson	Liking	18	41	0.267	86
	Esteem	18	63	0.411	107
	Influence	18	59	0.385	134
	Praising	18	50	0.326	97
London Transport	Underground	271	312	0.0085	374
	Overground	83	83	0.024	63
	DLR	45	46	0.046	46
Genetic and Protein	direct_ interaction	3111	5334	0.0011	5334
	physical_ association	239	255	0.0089	255
	additive_genetic_interaction _ defined_by_ inequality	1046	2112	0.0038	2112
	suppressive_ genetic_ interaction_ defined_by_ inequality	120	159	0.022	159
	Association	11	6	0.109	6
	Colocalization	14	8	0.0879	8
European Aviation	Lufthansa	106	244	0.0438	244
	Ryanair	128	601	0.0739	601
	Easyjet	99	307	0.063	307
	British_Airways	65	66	0.0317	66
	Turkish_Airlines	86	118	0.032	118

Definition 2: (Metrics for centrality in multilayer networks) Centrality metrics in multilayer networks use the concept of random pass. Therefore, we first define the concept of random pass in multilayer networks. In the multilayer networks without self-loops, we consider random transitions as discrete-time and finite positions. Assume a multilayer network $M = (V_M, E_M, V, L)$. We show node i in layer α with layer node (i,α) and edge weight between (i,α) and (j,β) with $w_{ij}(\alpha,\beta)$. We define the amount of node strength with $s_{i\alpha}$ for (i,α) as equal to the sum of the weights of all the edges, including the interlayer edges and the interlayer edges that intersect with (i,α). Also $n_{i\alpha}$ is the weight of the node (i,α) and $T_{j\beta}^{i\alpha} \in [0,1]$ tends to pass from (i,α) to (j,β). So for multilayer networks with nodes and weighted edges, random walk is defined as follows:

$$T_{j\beta}^{i\alpha} = \frac{(n_{i\alpha} + n_{j\beta}) w_{ij}(\alpha,\beta)}{\max\{\sum_{(k,\gamma)\in V_M}(n_{i\alpha} + n_{k\gamma}) w_{ik}(\alpha,\gamma), \epsilon\}} \quad ,$$

- Find the center of random pass

This centrality measures the probability of finding a random pass in a node position. The higher the probability is, the more important that node is. A classic random walk in a multilayer network is one that can pass from one node to the neighbors of that node via interlayer and interlayer edges. We assume that $T_{j\beta}^{i\alpha} \in [0,1]$ is the transition tendency and $P_{i\alpha}(t) \in [0,1]$ is the transition probability. Therefore, the centrality of the probability of finding a random passage in position (i, α) is defined as follows:

$$\pi_{i\alpha} = \lim_{t\to\infty} p_{i\alpha}(t) \ ,$$

With the community $\pi_{i\alpha}$ for node i for all layers of α, we obtain the community of centrality of the probability of finding a random transition for a node i.

- The center of the nearby random pass

Centrality of proximity measures the average distance between two nodes. If

$$p_{|d|}{}_{j\beta}^{o\sigma}(t) = (T_{|d|}^{t})_{j\beta}^{o\sigma} \ .$$

The probability of seeing node (j, β) after t is a time step, while passing through *(o, σ)* starts and stops when node d is observed if h is the first time passing through d. Then the probability that the transition before time t reaches d is as follows:

$$(q_{|d|})^{o\sigma}(h \geq t) = u^{o\sigma} - (T_{|d|}^{t})_{j\beta}^{o\sigma} u^{j\beta}$$

So the probability that the time of the first crossing of d is exactly at time t is equal to:

$$(q_{|d|})^{o\sigma}(h = t) = (q_{|d|})^{o\sigma}(h \geq t) - (q_{|d|})^{o\sigma}(h \geq t - 1) = [(T_{|d|}^{t-1}) - (T_{|d|}^{t})]_{j\beta}^{o\sigma} u^{j\beta} \ .$$

Therefore, the average of the first transit time starting at (o, σ) and ending at node d is equal to:

$$(H_{|d|})^{o\sigma} = \sum_{t=0}^{\infty} t(q_{|d|})^{o\sigma}(h = t) = \left[(\delta - T_{|d|})^{-1}\right]_{j\beta}^{o\sigma} u^{j\beta} \ .$$

$$\delta_{j\beta}^{o\sigma} = \begin{cases} 1 & , \quad if \quad j = o = \sigma = \beta \\ 0 & , \quad otherwise \end{cases}$$

The aggregation of all possible node starts gives us the average time of the first pass [21]:

$$h_{|d|} = \frac{1}{NK} u_{o\sigma} (H_{|d|})^{o\sigma} + \frac{1}{N} \pi_{|d|}^{-1}$$

Definition 3: (Kendall correlation coefficient) In statistics, the Kendall's correlation coefficient, which is known as Kendall's τ, is represented by a Greek τ, a non-parametric statistic used to measure the statistical correlation between two random variables. The τ test is also used to measure the degree of continuity between the two variables.

$$\tau = \frac{(number\ of\ concordant\ pairs) - (number\ of\ discordant\ pairs)}{\frac{1}{2}\, n(n-1)}$$

Kendall is always between -1 and +1, which corresponds to the matching and incompatible values. Kendall is expected to be close to zero for two independent variables. So if the agreement between the two rankings is perfect (i.e., the two rankings are the same) the coefficient has value 1. If the disagreement between the two rankings is perfect (i.e., one ranking is the reverse of the other) the coefficient has value −1. If X and Y are independent, then we would expect the coefficient to be approximately zero. The Kendall rank coefficient is often used as a test statistic in a statistical hypothesis test to establish whether two variables may be regarded as statistically dependent. This test is non-parametric, as it does not rely on any assumptions on the distributions of X or Y or the distribution of (X,Y).

Definition 4: (average_node_connectivity) For explain this title, first we define the concept connectivity in networks. In mathematics and computer science, connectivity is one of the basic concepts of graph theory: it asks for the minimum number of elements (nodes or edges) that need to be removed to separate the remaining nodes into isolated sub graphs. It is closely related to the theory of network flow problems. The connectivity of a graph is an important measure of its resilience as a network. So node connectivity is equal to the minimum number of nodes that must be removed to disconnect G or render it trivial. If source and target nodes are provided, this function returns the local node connectivity: the minimum number of nodes that must be removed to break all paths from source to target in G. Therefore the average connectivity $\bar{k}$ of a graph G is the average of local node connectivity over all pairs of nodes of G.

$$\bar{k} = \frac{\sum_{u,v} k_G(u,v)}{\binom{n}{2}}$$

D. Evaluation Process

In this section, we have seven parts of evaluation that shows us the benefit of our method. We used python programming language for virtualization and evaluation. In part one we have table 2 that shows structural features of reduced network which can compared with what we had in table 1. In part two we virtualize two networks, Sampson and London Transportation in multilayer mode with using of Pymnet library. Pymnet, is used to keep advantages of the properties of multilayer networks in the Python programming language. In part three we calculate multilayer centralities in four cases, Degree centrality, Closeness centrality, Intermediate centrality and Intermediate centrality of the edge, in our datasets. Then we compare these centralities before and after our reduction method with using of Kendall coefficient and evaluate how these values are close together or far from each other. We used function Kendall tau from scipy library in python. This function give 2 array and return the Kendall value of them, for example to calculate degree centrality of nodes, we call this function and give two arrays to it, one array of degree centrality of nodes before reduction and the other array of degree centrality of nodes after reduction finally it gives us the Kendall value of degree centrality. In part four we calculate and virtualize communities in the multilayer networks before and after reduction by using the functions of the community Louvain library in Python programming. In part five we calculate the network strength and robustness which is an important feature in the network and show how connect the network is. We know that with deleting nodes and edges in a network connectivity in that network may be in danger so

calculating the connectivity before and after reduction is important for evaluating how the reduction method is good. So we use average node connectivity for describing robustness in this part. In parts six and seven we did some evaluation with using of shortest path. For example in part six we calculate some shortest paths between some pairs of nodes then we calculate how many switches between the layers we have before and after reduction. Because the reduction method can increase in number of switches between layers in finding shortest path and a switch between layers had overhead cost, so whatever more switches we have, more overhead cost we will have, so a good reduction method is one that don't increase the switches too much. In the last part, seven, we shows the layer importance before and after reduction and as we said we use shortest path for it too. Whatever more edges of a shortest path be in a layer that layer is more important, so a good reduction method is one that don't change the importance of the layers too much.

VI. RESULTS

In this section, we want to show the results of the research on these four datasets using the criteria described in the previous section. All of these results d with python programming language version 3.5.

1. *Structural Features*

For all four datasets, structural features such as the number of nodes and edges, the modularity value, the execution time of the reduction algorithms and the execution time to find the shortest paths in the main network and the reduction network to compare them. As shown in the table 2, in the Simpson Monks Network, the results shows two nodes were deleted and almost 50% of the edges. In other networks, this decrease is evident. Modulation is a feature of network structure that demonstrates the power of the network to be divided into communities and higher modularity means it is the stronger connections between nodes within the community and weaker connections between intercommunity nodes. This metric has a value between 1 and -1. In all four datasets, the amount of modulation has increased after the reduction of the network means the number of connections inside a community has increased compared to the connections between communities. The reason for this is that all three algorithms used in our hybrid approach were in line with this maintenance and increase of modularity and keeping connectivity. Thus, the path simplification algorithm eliminates edges and nodes that maintain the network connection, so nodes and edges within a community are less likely to be removed because the community connection and ultimately the network connection are preserved. The Hair Reduction algorithm also removes isolated nodes in the network that are not in any community, so the probability of deleting them by this algorithm is also low. Therefore, nodes within communities that have a high degree of connection due to high communication are less likely to be removed by this algorithm. And in Random Degree Algorithm is the same we consider keeping connectivity before removing a node.

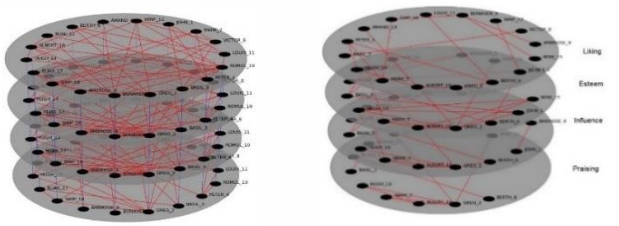
TABLE II. STRUCTURAL FEATURES OF THE REDUCED NETWORKS AND THE EXECUTION TIME OF OUR METHOD.

Datasets	Num. Of Nodes	Num. Of Edges	modularity	Execution Time Of Our method	Execution Time Of Finding SP
Sampson	18	213	- 0.19	7.26	0.0079
	16	83	- 0.06		0.0039
London Transport	369	441	0.674	14.84	1.094
	337	407	0.729		0.966
Genetic and Protein	3866	7874	0.58	1387.06	45.68
	1864	2164	0.76		41.62
European Aviation	417	3588	0.201	134.521	3.104
	332	1903	0.214		2.801

2. *Network View*

The first case in relation to the dataset is to display them as a multi-layered network before and after network-reduction operations, which provides an overview of the network status in general and in each layer, in terms of the number of nodes and edges, and density of the network and layers. The dataset is received to the Pymnet library as an input and its Multi-layer network is drawn using the functions defined in this library. Generally Pymnet, is used to keep advantages of the properties of multilayer networks in the Python programming language. Fig. 2, for example, shows the status of the Sampson Monks network before and after the reduction and a view of London Transportation Network.

a) Original network Reduced network



b)

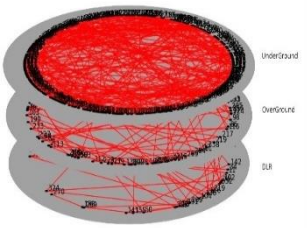


Figure 2. a) Network View of the Sampson dataset before and after reduction. b) Network View of the London Transportation Network befor reduction.

3. *Kendall Coefficients for Centrality*

In this section, for the datasets, four centrality centers based on centralities in the multilayers that we explain before in definition 2 are obtained in the field of degree centrality, closeness centrality, intermediate centrality and intermediate centrality of the edge before and after the reduction. Then, between these two groups, the Kendall correlation coefficient values are calculated, which, as we said in the concepts, has values between -1 and +1, as shown in Fig. 3.

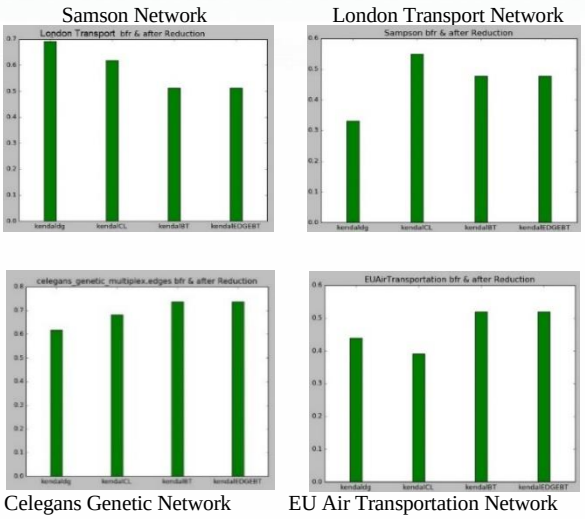


Figure 3. Kendall coefficient for centered degrees, proximity, interstitial, interstitial edges.

By analyzing Fig. 3, we find that almost all of Kendall's values for all networks are higher than 0.5, which indicates the good proximity of the reduced network to the main one, which means that the reduced network maintains very well the core features of the main network, such as the proximity of a node from the others, the importance rate of an edge, the importance of a node and even a single node in the network. As mentioned in the previous sections, we have used three algorithms to reduce the main network, one of them is which simplifies the path, the other one focuses on the removal of unnecessary edges in the network and the last one which directly affects the degree of nodes in the network. However, the degree centrality in all four networks has a good Kendall coefficient after reduction. This is due to the fact that these algorithms use a quality function, based on the weight of the edge and the track weight, to replace the straight edge between the two nodes, and not the uniform sampling methods that randomly remove the nodes and the edges.

4. *Compare Communities*

The Communities in the networks are drawn by using the functions of the community Louvain library in Python programming. As can be seen in the Fig. 4, in the case of communities, what happened after the reduction is that the number of communities in London's transport network and the Sampson's Monks network are constant, but because the node-to-node paths were modified by the algorithm to simplify them, for example a knot that has been highlighted in the Blue Society before the reduction, had lower importance in that Blue Society after reduction. Similarly, because of node selection based on degree algorithms, the node-based algorithm and hair-based algorithm reduce the number of nodes, which also affects the nodes within the communities. But the remarkable thing is that after applying a sequential mix of these three downsizing algorithms, the reduced network still maintains its connectivity.

But in European aviation networks and the interaction of genetics and protein, as higher density networks, it is as if the population of nodes in denser communities has been distributed after the downsizing in the remaining communities, which has led to the increase in the number of communities after network cuts.

5. *Network Strength and Robustness*

Calculate the strength of the networks is done with using the average network connectivity parameter to obtain it. In this section Fig. 5 shows the strength of the networks before and after the decrease. As we know, reducing the network's volume has a direct impact on the strength and integrity of that network, hence a good reduction method is one that delete such proportion of volume until the network maintains its strength to an acceptable level. Strength, ability to withstand failures and disturbances are critical features of complex networks. The study of power in complex networks is important for many areas.

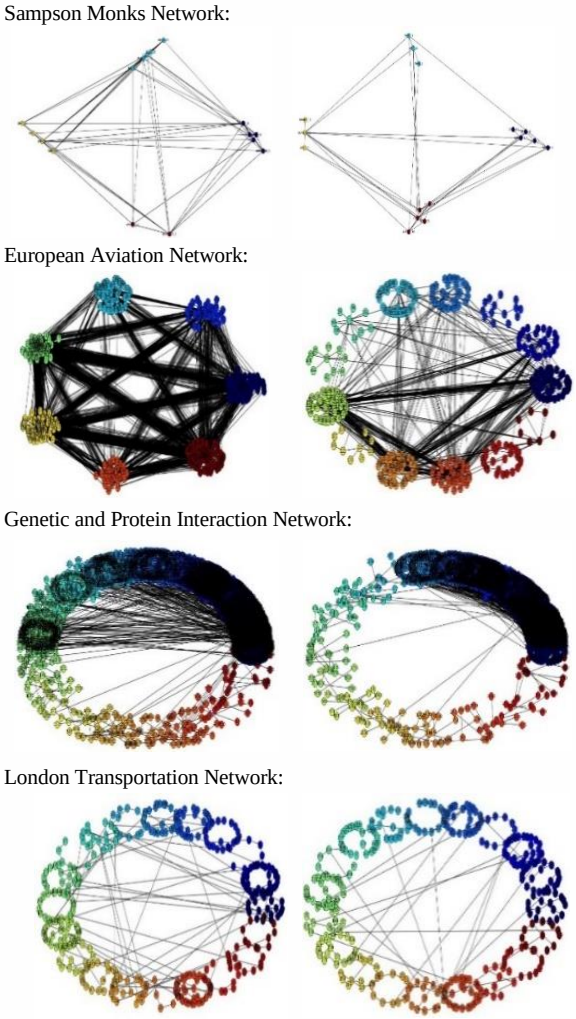


Figure 4. The images on the left are the main networks and the images on the right are the reduced networks.

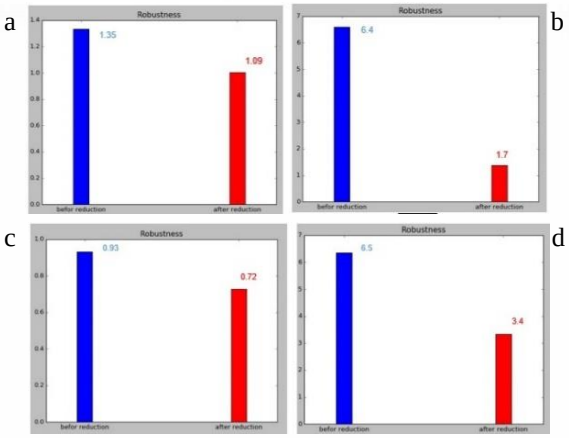


Figure 5. Robustness for a) European aviation network b) London Transportation Network c) Sampson Monks Network d) Genetic and Protein Interaction Network.

6. *Interlayer Switching Numbers*

In this section, all of the shortest paths are calculated in four datasets. To find the shortest path between the two nodes in the network, the switch between the layers is needed to find the edge that leads us to the destination node. So the average number of layer switch has been measured during that. The results shows that the number of switches between layers usually have been increased after the network reduction. Since the switch between the layers has an overhead cost for the network, so the smaller the number of switches are, the better and more efficient that reduction method is, as shown in Fig. 6. It has been noted that this increase has not exceeded of 0.2 in two Sampson networks and Genetic and Protein Interactions and in the London transportation network, even the number of switches has dropped to 0.2 after network cuts. But in the European aviation network, this amount has increased after a decrease of 0.6.

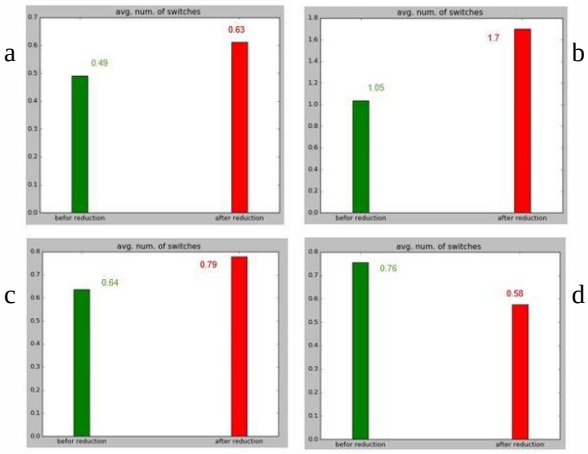


Figure 6. Average number of swithes for a) European aviation network b) London Transportation Network c) Sampson Monks Network d) Genetic and Protein Interaction Network.

7. *The Importance of Layers in Finding the Shortest Paths*

The importance of the layers in the networks is calculated using the fraction of the paths of the shortest paths are exist in that layer. In this section Fig. 7 shows the ranking of the importance of the layers which is based on the number of shortest paths in them. This ranking has been obtained before and after the reduction for each of the four sets of data. Clearly, the importance of a layer due to the network reduction operation is expected, but the problem that matters here is maintaining an approximate ranking of this before and after the reduction. For example, in London's transport network, the importance of layers is from the highest to the lowest, respectively: underground, extraterrestrial and terrestrial. Similarly, importance has been preserved even after the decline in this network, although the significance of each layer is lower than its predecessor, but it is important to maintain the order of the importance of the layers relative to each other in the network, as if the same network was just a bit smaller by the same nature.

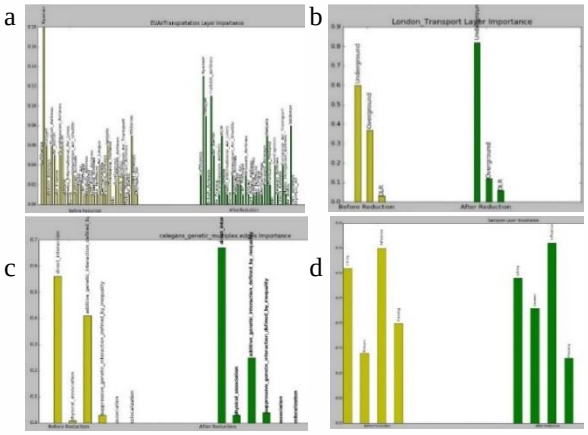


Figure 7. System Importance of layers in finding shortest paths a) European aviation net. b) London Transportation Net. c) Genetic and Protein Interaction Net. d) Sampson Monks Net..

VII. Conclusion and Future Works

In this research, we were able to reduce and simplify the multilayer networks and facilitate the study of network optimization problems such as the shortest paths. The largest networks in this study have been reduced to about 4000 nodes and 8000 edges. In the future, more emphasis will be placed on the features that affect the efficiency of the process of reducing and simplifying multilayer networks. In addition, other ways to compare the reduced network with the main network should be considered.

References

[1] N. Blagus, L. Subelj and M. Bajec, "Assessing the effectiveness of real-world network simplification," Elsevier, vol. 413, pp. 134-146, 2014.

[2] C. Hong, J. Zhang, X.-B. Cao and W.-B. Du, "Structural properties of the Chinese air transportation multilayer network," Chaos, Solitons and Fractals, vol. 86, pp. 28-34, 2016.

[3] A. Aleta and Y. Moreno, "Multilayer Networks in a Nutshell," Annual Review of Condensed Matter Physics, vol. 10, pp. 45-62, 2019.

[4] A. Santoro and V. Nicosia, "Algorithmic Complexity of Multiplex Networks," American Physical Society, vol. 10, no. 2, p. 021069, 2020.

[5] X. Wang and J. Liu, "A layer reduction based community detection algorithm on multiplex networks," Physica A: Statistical Mechanics and its Applications, vol. 471, pp. 244-252, 2017.

[6] H. G. Svendsen, "Grid model reduction for large scale renewable energy integration," Energy Procedia, vol. 80, pp. 349 -356, 2015.

[7] A. Arenas and J. Duch, "Size reduction of complex networks preserving modularity," New Journal of Physics, vol. 176, pp. 1367-2630, 2007.

[8] P. M. Pardalos, I. Granata, M. R. Guarracino, V. A. Kalyagin, L. Maddalena and I. Manipur, "Model simplification for supervised classification of metabolic networks," Annals of Mathematics and Artificial Intelligence , 2019.

[9] F. Salo, A. B. Nassif and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," Computer Networks, vol. 148, pp. 164-175, 2019.

[10] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno and M. A. Porter, "Multilayer networks," Journal of Complex Networks, vol. 2, pp. 203-271, 2014.

[11] J. Jiang, R. Zhang, L. Guo, W. Li and X. Cai, "Network Aggregation Process in Multilayer Air Transportation Networks," Chinese Physical Society and IOP Publishing Ltd, vol. 33, no. 10, pp. 1-5, 2016.

[12] T. Biedl, T. Vinar and B. Brejová, "Simplifying flow networks," Mathematical Foundations of Computer Science, vol. 1893, pp. 192-201, 2000.

[13] E. Misiołek and D. Chen, "Efficient Algorithms for Simplifying," Computing and Combinatorics, pp. 737-746, 2005.

[14] F. Khamayseh and N. Arman, "Comparing the approaches of graph_reduction and landmark shortest path," Journal of Theoretical and Applied Information Technology, vol. 83, no. 1, pp. 138-147, 2016.

[15] M. D. Domenico, A. Solé-Ribalta, S. Gómez and A. Arenas, "Navigability of interconnected networks under random failures," PNAS, vol. 111, no. 23, pp. 8351-8356, 2014.

[16] A. Cardillo, J. Gómez-Gardenes, M. Zanin, M. Romance, D. Papo, F. d. Pozo and S. Boccaletti, "Emergence of network features from multiplexity," Scientific Reports, vol. 3, 2013.

[17] M. D. Domenico, V. Nicosia, A. Arenas and V. Latora, "Structural reducibility of multilayer networks," Nature Communications, vol. 6, pp. 6864-6865, 2015.

[18] C. Stark, B.-J. Breitkreutz and M. Tyers, "Biogrid: a general repository for interaction datasets," Nucleic Acids Research, vol. 34, no. 1, pp. 535-539, 2006.

[19] Z.-M. Lu and X. Chen, "Measure of layer centrality in multilayer network," International Journal of Modern Physics C, vol. 29, pp. 185-186, 2018.

[20] H. Selim and J. Zhan, "Towards shortest path identification on large networks," Journal Of Big Data, vol. 3, 2016.

[21] D. Wang and X. Zou, "A new centrality measure of nodes in multilayer networks under the framework of tensor computation," Applied Mathematical Modelling, vol. 54, pp. 1016-1017, 2017.

[22] W. Schilders, "Introduction to Model Order Reduction," Model Order Reduction: Theory, Research Aspects and Applications, vol. 13, pp. 3-32, 2008.

[23] Y.-C. Liang and H. Lee, "Proper Orthogonal Decomposition and it's applications_Part 1: theory," Journal of Sound and Vibration, vol. 252, no. 3, pp. 527-544, 2002.

[24] D. Tuninetti, C. Fragouli, M. Cardone and Y. H. Ezzeldin, "Network Simplification in Half-Duplex: Building on Submodularity," IEEE Transactions on Information Theory, vol. 65, no. 10, pp. 6801-6818, 2019.

[25] M. L. Lamali, N. Fergani, J. Cohen and H. Pouyllau, "Path computation in multi-layer networks: Complexity and algorithms," Networking IEEE/ACM Transactions, vol. 26, pp. 2787-2800, 2018.

[26] F. Khamayseh and N. Arman, "Comparing the approaches of graph_reduction and landmark shortest path," Journal of Theoretical and Applied Information Technology, vol. 83, pp. 138-147, 2016.

[27] G. Li and A. Balakrishnan, "Models and algorithms for network reduction," European Journal of Operational Research, vol. 248, pp. 930-942, 2016.

Mahya Moheb received her B.Sc. degree in Software Engineering from Shahid Rajaei University, Tehran, Iran, in 2015, and her M.Sc. degree in Network Systems and Information Technology Engineering from Department of Network Science and Technology, Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran, in 2018. She is currently working in System Group Company as an IT engineer. Her research interests are in the area of Multilayer Network Reduction and Embedding.

Sayyed Hamid Reza Ahmadi received his B.Sc., M.Sc., and Ph.D. degrees in Electrical and Computer Engineering from University of Tehran, Tehran, Iran, in 1998, 2001, and 2011, respectively. Since 2012, he has been with University of Tehran, where he is an assistant professor with the Faculty of New Sciences and Technologies. His current research interests include Data and Network Security, Software-Defined Networks (SDN), Blockchain and DLT, Watermarking, and Video Compression.

M. Ebrahimi is an Assistant professor of Applied Mathematics at the Faculty of New Sciences and Technologies, University of Tehran, Iran. His research expertise is in mathematical modeling of engineering problems that have arisen in the fields of networked systems, information technology, decision sciences and health. He has been active in the combined numerical probabilistic algorithm based on Monte Carlo Methods for solving optimization problems. He has been active in research fields including Multi-Criteria Decision Making Methods and Mathematical Modeling for Diffusion Processes. He has published about 60 research papers. He has set up the Laboratory of Engineering and Analysis of Decision Support Systems (LEADS).