REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical-Electronic Engineering

RELIABILITY ANALYSIS BASED ON OPTIMAL LOCATION PARTITION OF COMMUNICATION NETWORK

Master Thesis

Rafal Hameed Hadi AL-ELAYAWI

Supervisor

Assoc. Prof. Dr. Indrit MYDERRIZI

Istanbul – 2022

THESIS INTRODUCTION FORM

Name and Surname	**:**	Rafal Hameed Hadi
Language of the Thesis	**:**	English
Name of the Thesis	**:**	Reliability Analysis Based on Optimal Location Partition of Communication Network
Institute	**:**	Istanbul Gelişim University Graduate Education Institute
Department	**:**	Electrical-Electronic Engineering
Thesis Type	**:**	Post Graduate
Date of the Thesis	**:**	09.02.2022
Page Number	**:**	76
Turkish Abstract.AAA	**:**	Bir ağı kümelemek, bireylerin ağı birkaç alt ağa ayırmasını sağlar. K-ortalama kümeleme, basitliği ile bilinen en iyi bilinen kümeleme tekniklerinden biridir. Temel olarak K-ortalama, kümeleri oluşturmak için düğümlerin coğrafi konumlarına ve aralarındaki Öklid mesafesine bağlıdır. Bu çalışmada, kümeleme sonuçlarının doğruluğunu artırmak için sadece coğrafi konum ve mesafeye değil, aynı zamanda düğümler arasındaki bağlantıya da bağlı olarak K-ortalama kümeleme algoritması için bir geliştirme olarak yeni bir algoritma önerilmiştir. Ağ güvenilirliği, bir ağın ve tüm alt ağlarının, bir kaynak ve hedef (hedef) arasında belirtilen bir süre boyunca tanımlanan koşullar altında gerçekleştirmeyi amaçladığı görevi mükemmel bir şekilde tamamlama olasılığıdır. Ağ hatalarını hesaplamak ve kontrol etmek ve özellikle ağ çok sayıda düğüm içerdiğinde başarısızlık sonuçlarını

tahmin etmek zordur. Ağın hesaplama karmaşıklığını azaltmak için ağı, kullanımı kolay daha küçük alt ağlara ayırmak için kümeleme gibi bir basitleştirme yöntemi önerilmiştir.

Thesis Supervisors **:** Assoc. Prof. Dr. INDRIT MYDERRIZI

Index Terms **:** Clustering, Partitioning, Reliability, K-mean.

Distribution List **:**
1. To the Institute of Graduate Studies of Istanbul Gelisim University
2. To the National Thesis Center of YÖK (Higher Education Council)

Rafal Hameed Hadi AL-ELAYAWI

REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical-Electronic Engineering

RELIABILITY ANALYSIS BASED ON OPTIMAL LOCATION PARTITION OF COMMUNICATION NETWORK

Master Thesis

Rafal Hameed Hadi AL-ELAYAWI

Supervisor

Assoc. Prof. Dr. Indrit MYDERRIZI

Istanbul – 2022

DECLARATION

I hereby declare that in the preparation of this thesis, scientific ethical rules have been followed, the works of other persons have been referenced in accordance with the scientific norms if used, there is no falsification in the used data, any part of the thesis has not been submitted to this university or any other university as another thesis.

Rafal Hameed Hadi AL-ELAYAWI

…/…/2022

TO ISTANBUL GELISIM UNIVERSITY

THE DIRECTORATE OF THE INSTITUTE OF GRADUATE STUDIES

The thesis study of Rafal Hameed Hadi titled as RELIABILITY ANALYSIS BASED ON OPTIMAL LOCATION PARTITION OF COMMUNICATION NETWORK has been accepted as MASTER THESIS in the department of ELECTRICAL-ELECTRONIC ENGINEERING by out jury.

Member	*Signature* *Assoc. Prof. Dr. Indrıt MYDERRİZİ* (Supervisor)
Member	*Signature* *Asst. Prof. Dr. Ahmed A. SOLYMAN*
Member	*Signature* *Asst. Prof. Dr. Mussaria K. MAHMOOD*

APPROVAL

I approve that the signatures above signatures belong to the aforementioned faculty members.

... / ... / 2022

Signature

*Prof. Dr. İzzet G*ÜMÜŞ

Director of the Institute

SUMMARY

Clustering a network enables the individuals to separate the network in to number of subnetworks. K-mean clustering is one of the most well-known clustering techniques, known for its simplicity. Basically the K-mean depends on the nodes' geographical locations and the Euclidian distance between them to create the clusters. In this work, a new algorithm has been proposed as a development for the K-mean clustering algorithm in order to increase the clustering results accuracy, by depending not only on the geographical location and distance but also on the connectivity between nodes. Network reliability is the probability that a network and all its subnetworks will perfectly complete the task that intend to perform under the conditions defined for a period of time specified between a source and destination (target). It is difficult to calculate and control networks faults and predict failure consequences especially when the network contains massive number of nodes. In order to reduce the network computational complexity, a simplification method has been proposed like clustering to separate the network into smaller subnetworks that are easy to handle.

Keywords : Network Clustering, Reliability, partitioning, clustering techniques, connectivity based clustering, network modeling, tie-set.

ÖZET

Bir ağı kümelemek, bireylerin ağı birkaç alt ağa ayırmasını sağlar. K-ortalama kümeleme, basitliği ile bilinen en iyi bilinen kümeleme tekniklerinden biridir. Temel olarak K-ortalama, kümeleri oluşturmak için düğümlerin coğrafi konumlarına ve aralarındaki Öklid mesafesine bağlıdır. Bu çalışmada, kümeleme sonuçlarının doğruluğunu artırmak için sadece coğrafi konum ve mesafeye değil, aynı zamanda düğümler arasındaki bağlantıya da bağlı olarak K-ortalama kümeleme algoritması için bir geliştirme olarak yeni bir algoritma önerilmiştir. Ağ güvenilirliği, bir ağın ve tüm alt ağlarının, bir kaynak ve hedef (hedef) arasında belirtilen bir süre boyunca tanımlanan koşullar altında gerçekleştirmeyi amaçladığı görevi mükemmel bir şekilde tamamlama olasılığıdır. Ağ hatalarını hesaplamak ve kontrol etmek ve özellikle ağ çok sayıda düğüm içerdiğinde başarısızlık sonuçlarını tahmin etmek zordur. Ağın hesaplama karmaşıklığını azaltmak için ağı, kullanımı kolay daha küçük alt ağlara ayırmak için kümeleme gibi bir basitleştirme yöntemi önerilmiştir.

Anahtar Kelimeler : Ağ Kümeleme, Güvenilirlik, bölümleme, kümeleme teknikleri, bağlantı tabanlı kümeleme, ağ modelleme, bağ kümesi.

TABLE OF CONTENTS

CHAPTER THREE

METHODOLOGY AND IMPLEMENTATION

CHAPTER FOUR

IMPROVED ALGORITHM APPLICATION

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORK

ABBREVIATIONS

PM	**:**	Probability Matrix
DTC	**:**	Distance To Centroid matrix
IPA	**:**	Intermediary Partition
OEP	**:**	One End Partition
CP	**:**	Complete Partition

LIST OF TABLES

LIST OF FIGURES

LIST OF ANNEXES

PREFACE

During the preparation and writing process of this thesis, i would like to thank my esteemed professor Assoc. Prof. Dr. Indrit MYDERRİZİ.

Finally, I want thanking my family for their supporting.

INTRODUCTION

Clustering (partitioning) techniques is one kind of unsupervised learning technique, it is widely used in many applications including DNA analysis in biotechnology, security, analyzing music and feature, biochemistry. Clustering a network is creating several subnetworks while maintaining the full performance of the system. The clustering algorithms are as varied as the clustering criteria's. Mainly, the algorithms are categorized in to Partitioning clustering, hierarchical clustering and density based clustering. One of the most used and well-known algorithms is the K-mean clustering algorithm which is one of the partitioning clusterings. The K-mean clustering depends on the geographical location to specify number of initial points then, calculating the Euclidian distance between the initial points and the rest of the nodes to arrange the clusters that those nodes belong to.

High attention in modern society is given to network performance and network reliability, yet, evaluation of the reliability for a network is a hard and complicated procedure especially for networks with large scales, at this point the importance of clustering arise besides the concept of reliability. Reliability measurements quantify system behaviour related to its failure. Numerous methods are proposed to evaluate the reliability of a system. Some of the commonly used analytical methods are the State-space method using the Markov process, Network reduction method, Cut-set/Tie-set method, and Graph transformation.

The proposed enhancement in this study is to improve the accuracy of the classical K-mean clustering algorithm by adding connectivity as an additional constraint next to geographic location to generate clusters. After the network is clustered, a reliability evaluation method based on the Tie-set algorithm is applied to evaluate the reliability of the network.

CHAPTER ONE

PURPOSE OF THE THESIS

Wide attention from different research communities attracted to the area of reliability evaluation and network partitioning including clustering and graph partitioning as a core concept.

1.1.Literature Survey

Much related work has been done on many issues related to improved K-mean algorithms some to increase the accuracy and others to decrease the evaluation time and other criteria, for example:

Surbhi Bhatia (2014), the researcher proposed an improvement in initial centre locating by using a Genetic Algorithm instead of random locating in the classical method to have a fast convergence to the local optimum with complexity decreasing and accuracy increasing. This algorithm helps in smoothing the research progress by providing stable clusters even in large data sets. The Genetic Algorithm follows the heuristic approach to find the optimal solution following the evaluation of the Darwinian Theory which concentrates on the "survival of the strongest" concept.

Caiquan Xiong, Zhen Hua, Ke Lv and Xuan Li (2016), improvement is proposed for the classical K-mean clustering by combining two methods "distance optimization" and "density" to find the best initial cluster centre. The improvement affects the clustering stability and accuracy. At first, the densities of all data points are calculated. The data object which has the farthest distance or largest density is used as an initial cluster centre. If there are isolated data points to be removed from the data collection to avoid choosing those data points as the initial cluster centre. Secondly choosing the next cluster centre point from the other high densities data points that is the furthest from the previous initial cluster centre, and so on until all K initial cluster centres are found. The proposed algorithm decreases the cluster sensitivity of the K-mean clustering algorithm to the initial cluster centre.

Shyr-Shen Yu, Shao-Wei Chu, Chuin-Mu Wang, Yung-Kuan Chan and Ting-Cheng Chang (2018), in this study two approaches have been proposed, the tri-level K-mean algorithm and Bi-layer K-mean algorithm. The tri-level K-mean algorithm constraints on the clusters with a big number of nodes that need to be simplified into

smaller clusters and this is done with four stages. The first stage is data normalization and the other three stages are three levels of clustering. This approach provides a higher clustering accuracy and has a higher noise tolerance. The Bi-layer K-mean algorithm employs the same data normalization as the Tri-level approach but with further separation using the classical K-mean algorithm. This approach provides more effective classification and higher noise resistance. A genetic-based algorithm is used to determine the values of the most suitable parameters that are employed in both tri-level and bi-level approaches.

Cheng-Hao Deng and Wan-Lei Zhao (2018), the work presents a method to overcome the scalability problem of the K-mean algorithm using the K-nearest neighbour graph for the large data sets. The solution proposed is achieved with the use of an approximate K-NN graph in the iterations of the K-mean clustering algorithm. A single node needs to be compared with the clusters that the nearest neighbours establish. For this reason, the clustering complexity is irrelative to the number of clusters. If the number of input data and the number of clusters are very large, a speed-up of hundreds to thousands of times is achieved. Over the existing solutions, the proposed algorithm shows a great trade-off between clustering efficiency and quality.

Mohamed LEHSAINI and Meriem Bouchra BENMAHDI (2018), and improved K-mean clustering algorithm is proposed to be used by the wireless sensor network to have balanced clusters. Specifying limit for the maximum number of nodes in each cluster to distribute the load equally among the clusters. The limitation of nodes number improves the network lifetime and decreases the power consumption. After the first clustering process with classical K-mean clustering, the average number of nodes within the clusters need to be computed. If the number of nodes within the cluster exceeds the limit, the additional nodes must join the clusters that have not reached the maximum number of nodes.

Guang-Yu Zhanga, Chang-Dong Wanga, Dong Huang, Wei-Shi Zhenga and Yu-Ren Zhou (2018), a multi-view clustering called Two-level Weighted Collaborative K-mean is proposed which gives the ability to analyse data from multiple views. An objective function has been designed for clustering in multi-view, exploiting specific information in each view while using the complementariness and consistency advantages of different views in a cooperating manner. The importance of the data points in each view gets affected by the weights that are allocated for the views and

the characteristics in those views. In order to accomplish the clustering results, an iterative optimization method has been used to find the optimum of the objective function.

Yang Chen, PengFei Hu and Weilan Wang (2018), another improvement for the classical k-mean clustering algorithm cluster centre selection depending on the Mean Shift algorithm has been proposed. In general, the mean shift algorithm is based on the density, as the maximum probability density has to be considered in the distribution of clusters centres. The improved algorithm specifies the initial clusters' centres according to the spatial distribution of data, the points farthest from the high-density region are assigned as clusters' centres. The proposed algorithm provides more stable results, and it has been slightly improving the accuracy of the k-mean clustering algorithm. Calculation of two Parameters is required. The first parameter is the Min Distance to determine if the data point needs to be offset. And the second parameter is the kernel-bandwidth of the kernel function to compute the dot product of two vectors. Those two parameters have a great effect on the clustering results.

Chengchang Zhang, Yuwen Gong and Huayu Zhang (2019), present an improvement for the classical k-mean clustering algorithm initial cluster centre selection by adding the most powerful algorithm of sample density which improves the accuracy and reduces the instability of the K-mean clustering algorithm. The sample element density has to be calculated to appoint the element with the highest density as the first cluster centre. Also, the weights of the remaining elements have to be calculated to specify the element with maximum weight as the second cluster centre and so on for the rest of the clusters' centres. MeanDis condition has been used to arrange the elements to their clusters.

Jinyan Liu, Dongqing Wang, Zhuo M and Boyang Xu (2019), one of the limitations in the classical K-mean clustering algorithm is the difficulty of solving nonlinear classification problems. The kernel function is used to transform the nonlinear problems into linear problems as a base step to the classical k-mean clustering algorithm. By using the improvement, the accuracy of the classical K-mean clustering algorithm increased by 6.25%.

Rajdeep Baruri, Anindya Das, Anannya Ghosh, Arindam Mandal, Ranjan Banerjee and Tapas Halder (2019), shows an improvement in reducing the execution

time of the K-mean clustering algorithm at the cluster centroid reassignment step instead of using the traditional Euclidian distance to reassign the centroids of the clusters. A distance comparing approach has been used, compares the distances between the nodes and the centroids before and after the cluster centre changes. If the distance after the centre changes is less or equal to the distance before the centre changes that means, there is no need to recalculate the distance.

Su-yu Huang and Bo Zhang (2019), an improvement in the initial cluster centre selection and clusters number specification is proposed based on the Hadoop platform algorithm. Hadoop algorithm uses the canopy algorithm based on the cosine similarity. The canopy algorithm is a sup clustering algorithm used in the early stages of clustering to find the most suitable number of clusters. It uses cosine similarity to calculate the difference between clusters. The proposed algorithm speeds up the clustering process and has a better effect on the clustering result of large sets.

V.Divya, R.Deepika, C.Yamini and P.Sobiyaa, (2019), one of the classical K-mean clustering algorithm limitations is the instability of the clustering results for the same data set due to the initial centre random selection. To avoid such a limitation an initialization algorithm is proposed using the Cuckoo search algorithm to find the initial cluster centre. The Cuckoo algorithm is inspired by the Cuckoo birds' unique behaviour of committing brood parasitism. Those birds are laying their eggs in the nest of the host birds according to the similarity between their eggs and the host's eggs, to increase the chance of their survival and productivity. The best nest from the host nests is assumed to be the cluster centroid. The results show that the improved algorithm gives more accurate results compared with classical K-mean clustering results.

Naji Albakay, Michael Hempel, Mahmoud Alahmad and Hamid Sharif (2019), initial cluster centre selection improvement is proposed in this work. The traditional K-mean clustering algorithm chooses the initial centres randomly then iteratively updates take place until the optimum centres are found. While the modified algorithm uses predefined clusters centres. The proposed algorithm is used to find the frequency offset of the received signals in wireless communication by clustering the signal. The centroids assignment depends on the modulation format which also affects the number of clusters. The improved algorithm overcomes the errors of classification that take place because of the random selection of the centres, so as it reduces the number of iterations and improves the efficiency.

Dong G, Jin Y, Wang S, Li W, Tao Z, Guo S (2019), DB-K-means algorithm is proposed to achieve the benefits of both the DBSCSN clustering and K-mean clustering and combine them. The algorithm improves the random initial cluster centre selection in the K-mean clustering algorithm by using the density function. The points with the highest density function are selected as the centres of the initial clusters. The combination of the two algorithms eliminate the effect of noise points and increases efficiency.

Siwei Wang, En Zhu, Jintao Hu, Miaomiao Li, Kaikai Zhao, Ning Hu and Xinwang Liu (2019), two novel approaches are adopted in this work, average multiple kernel K-mean with late fusion and adaptive multiple kernel K-mean with late fusion. The kernel method is used to convert the nonlinear problem into linear problems, as the k-mean clustering algorithm is effective only with linear problems. Theoretically, the proposed algorithm combines the clustering process and the optimization into a single optimization problem which improves the convergence and reduces the computational complexity.

Kristina P. Sinaga and Miin-Shen Yang (2020), presents a free-of-initialization k-mean algorithm called the U-k-mean algorithm that uses the entropy concept to calculate the number of clusters. The proposed algorithm uses several clusters equal to the number of points. During the iterations of the U-k-mean algorithm, the extra clusters are discarded and the optimal number of clusters is found according to the structure of data.

Youjin Rong and Yi'an Liu (2020), provide a hybrid clustering algorithm SC-KH (Staged text Clustering algorithm based on K-means and Hierarchical clustering algorithms). The algorithm is aimed to obtain a more accurate clustering technique by overcoming the insufficiency of k-means low accuracy and the high time complexity of the hierarchical clustering. Two stages have been used to provide the clusters, the splitting stage using k-mean clustering and the merging stage using the hierarchical clustering. The Canopy calculation is appropriate to specify the number of clusters in the early levels of the K-mean clustering algorithm to provide the K value (number of clusters) and the initial centroids. The K value is equal to the canopy number and each centre point selected by the canopy algorithm is used as an initial centre point for the K-mean algorithm.

Arghyadeep Sen, Prof. Manjusha Pandey and Prof. Krishna Chakravarty (2020), in this particular research improvement of two classical k-mean shortcomings are proposed. The initial centre random selection is enhanced using the genetic algorithm and overcoming the predefined number of clusters using the elbow method. The nodes are modelled as n chromosomes, for each of those chromosomes a predefined fitness function has to be applied. The chromosomes with the highest fitness function are selected as initial clusters centres for each cluster specified by the elbow method. These improvements increase the accuracy of the classical k-mean clustering.

Yang Liu, Shuaifeng Ma and Xinxin Du (2020), Kwdm algorithm is proposed which overcomes random selection of the clusters' centres set in the classical k-mean clustering algorithm. The algorithm adds weight to the Euclidian distance to optimize the initial cluster selection which improves the accuracy of the results. Also, the Kohonen network clustering process is applied to select the number of clusters using a self-organizing map.

Wenqiang Guo, Zixuan Huang, Yongyan Hou, Qinkun Xiao, Jia Jia and Lingling Mao (2020), the improved algorithm overcomes the initial cluster centre selection shortcoming by adding weight to the data points. According to the proposed algorithm, the highest weight is given to the nodes near the cluster centres, and the weight of nodes decreases when moving away from the cluster centre. The error function has to be calculated to determine the convergence of the clustering process. This proposed clustering algorithm improves the effect of clustering by weakening the randomness of the initial cluster centre, providing a better clustering accuracy compared with the classical K-mean clustering algorithm.

1.2.Problem Statement

The K-mean clustering algorithm enables individuals to cluster networks into several small subnetworks easily and quickly using only the distance measurements. Although it is simple and easy to implement it also has relatively low accuracy. Thus, this makes us concentrate on developing an algorithm to improve the accuracy of the classical k-mean clustering algorithm by using another constraint besides the distance measurements of the geographical location. Using another constraint provides

alternate clustering results, which seems to be very useful for many applications like network security and reliability.

Reliability evaluation is a very complicated process especially for large networks that contain a massive number of nodes. From this point of view, the importance of clustering arises besides the reliability evaluation. Clustering simplifies large networks into subnetworks whose reliability is easy to find.

1.3.Aim and Objectives

The present work aims to develop an improved algorithm of the K-mean algorithm based on the nodes geographical locations and network connectivity. The algorithm includes two constraints, the distance measurements while clustering using the classical K-mean algorithm, and the connectivity consideration using the network topology matrix to provide more accurate clustering results when used for wired or wireless networks. Also, the minimum number of nodes within the clusters is specified. A series of checks are made for all nodes to ensure that the nodes are connected in the cluster to which the node belongs. The objectives of this work are formulated as:

- Adding the connectivity constraint using the network topology matrix.
- Partitioning any random network into clusters.
- Evaluating the reliability of any complicated network.

1.4.Thesis Organization

The thesis is composed of five chapters.

- **Chapter One** includes general information, related work, the problem statement, and the objective of the thesis.

- **Chapter Two** provides a theoretical background of network, the clustering and its well-known types, the reliability and the mobile communication network design.

- **Chapter Three** presents the methodology adopted to develop the improved K-mean algorithm based on the original classical K-mean algorithm.

- **Chapter Four,** the simulation results of randomly generated networks are presented where the validation of the developed algorithm is clarified, and the reliabilities of those networks are evaluated.

- **Chapter Five** concludes the work and presents some ideas for future work.

CHAPTER TWO
THEORETICAL BACKGROUND

The field of network partitioning, including clustering as a core concept and graphical partitioning, has received wide attention from different research communities. In this chapter, a detailed survey of the fundamentals of this research is presented. In section 2.1. the graph theory in network modelling is explained. While in section 2.2. the well-known clustering techniques are briefly explained. Then in section 2.3. the network reliability and the main methods used to evaluate the reliability are explained.

2.1 Network Modeling

A network is a set of entities called nodes that are connected by lines. In mathematics, networks are mainly modeled as graphs (G) by using graph theory, even though the graph representation of the network is quite simple to create, it also may be very complicated and hard to be analyzed. Today, many real-life schemes can be realized as a network, for example, social networks, electrical networks, metabolic networks in computational biology (Chormunge and Jena, 2014), geographical location systems (Eslamnezhad and Varjani, 2014), internet, communication networks, World Wide Web, artificial intelligence (Nath, Lee, Chowdhury, and Chang, 2010).

2.1.1 Graph Theory

A graph is a network corresponding to a set of interrelated elements called nodes or vertices. Each pair of associated nodes is called an edge or link. Each edge joins two of the nodes. A graph (G), as shown in Figure 1, can be referred to as a pair of vertices (V) and edges (E),$G = (V, E)$.

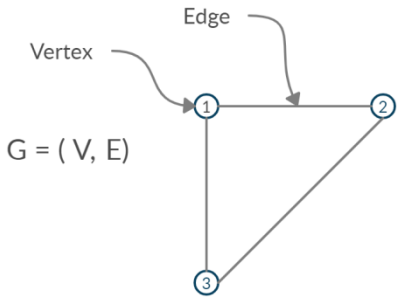

Figure 1. Network graph.

Depending on the connectivity lines (edges), graphs can be classified into different types, such as:

- *Connected Graph*

All the nodes in such a graph are connected. There are one or more edges connecting the node to the other nodes within the network. As a result, the graph does not contain any separated nodes (unconnected nodes). Figure 2 shows a graph with 3 nodes and 3 edges, all nodes connected.

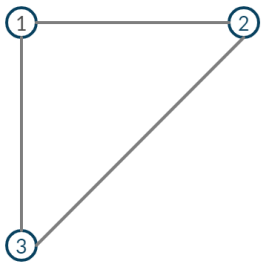

Figure 2. Simple connected graph.

- *Unconnected Graph*

In this type of graph, there is at least one node that is not connected to the other nodes within the network (separated node). A single network may have more than one unconnected node. The graph shown in Figure 3 where node 4 is an unconnected node (separated) illustrates an unconnected graph.

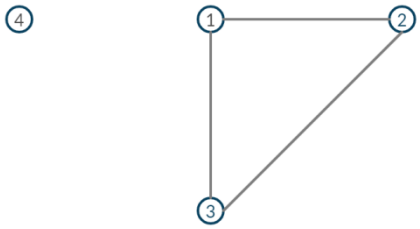

Figure 3. Unconnected graph.

• *Undirected Graph*

It is a simple network graph that contains only the nodes and their edges, like the graph shown in Figure 2.

• *Directed Graph*

The directed network graph, as shown in Figure 4, consists of edges (E) presented as arrowed lines or arcs to indicate the direction of information flow.

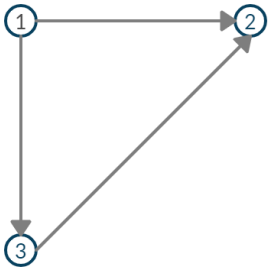

Figure 4. Directed graph.

The undirected graph can be transformed into a directed graph easily by replacing the undirected edge with two opposite directed edges (arcs) as shown in Figure 5, however, those two arcs are dependent, and so the failure in one of the directed arcs typically implies the failure in the opposite direction as well.

Figure 5. Transformation of Undirected graph to a directed graph.

• *Weighted Graph*

The weight is associated with the edges of the network as shown in Figure 6. The weight of the link is the measure of the capacity of the line, the energy, the probability, the length, the cost... etc. depending on the application of use. A graph can be both directed and weighted.

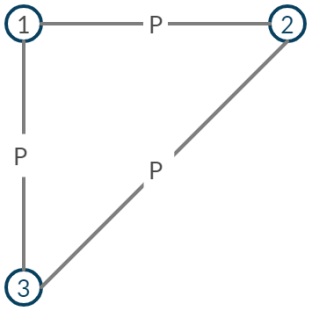

Figure 6. Weighted Graph.

2.1.2 Data Structure (Network Topology)

Network topology is the method used to show the arrangement of the elements of the network, it also can be used to define or describe some of the network properties like probability, connectivity …, etc.

• *Probability Matrix (PM)*

A fter representing the network as a graph $G = (V, E)$, all edges (P_e) and nodes (P_n) should have probabilities in order to work perfectly. The network is formed by probability matrix (PM). The probability matrix is a square symmetric matrix of order $N_V \times N_V$ (N_v is the total number of nodes). Each element in the matrix $a_{i,j}$ represents the probability of the link (edge) that connect node i and node j, If $a_{i,j} = 0$ means there is no connection between node i and node j. The diagonal elements of the matrix $a_{i,j}|i = j$ represent nodes probability which usually is equal to 1, as nodes are assumed to be perfect and 100% probable since the network contains redundant components. Figure 7 shows a simple network

graph of 5 nodes and 6 edges; equation 1 shows the probability matrix of the network topology.

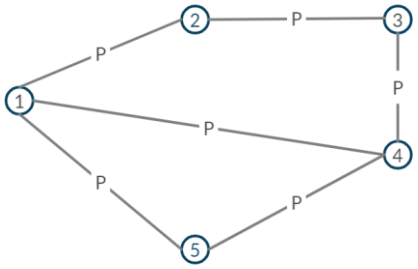

Figure 7. 5-nodes network topology

$$PM = \begin{bmatrix} 1 & p & 0 & p & p \\ p & 1 & p & 0 & 0 \\ 0 & p & 1 & p & 0 \\ p & 0 & p & 1 & p \\ p & 0 & 0 & p & 1 \end{bmatrix} \ldots\ldots (1)$$

• *Adjacency Matrix (AM)*

The adjacency matrix is a square matrix of order $N_V \times N_V$(N_v is the number of nodes (vertices)). The elements of the matrix indicate the connectivity of the nodes in the network. If element $a_{i,j} = 1$, means there is a connection between node i and node j, if $a_{i,j} = 0$, means there is no connection between node i and node j. Equation 2 shows the adjacency matrix of the network in Figure 7.

$$AM = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \ldots\ldots (2)$$

• *Cost Matrix*

The cost matrix is the function related to the length of each edge (link). The cost matrix is a symmetric square matrix, each element $a_{i,j}$ in the matrix shows the distance between node i and node j. This type of network topology modelling is more related to the geographical location of the nodes, thus, the nodes' locations have to be predefined.

2.1.3 Graph Connectivity

Basically for N-node network it is possible to have number of links connected to the node equal to $(N-1)$, so the possible number of links in one layer is equal to : $N \times (N-1)$ in case of flat network. The flat network is a two dimensional network that does not contain any parallel edges connects the nodes. In case there are parallel edges connecting the nodes, the topology matrix of the network becomes a three-dimensional matrix (N×N×D) (cubic matrix) shown in Figure 8, where D is the number of layers and each layer contains one of the parallel links.

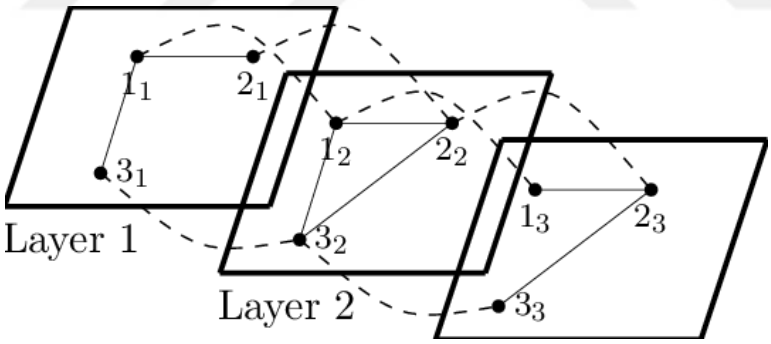

Figure 8. Multilayer matrix

2.2 Clustering

Clustering is one of the growing areas of research, it refers to the process of grouping together objects with similar properties while separating objects with different properties. The purpose of clustering is to have high-quality groups with low inter-cluster similarity and high intra-cluster similarity.

2.2.1 Fundamentals of clustering

There is a big variety of clustering methodologies that Includes but are not limited to, partition-based clustering, fuzzy clustering, Hierarchical clustering, density-based clustering, and Model-based clustering (D. Sisodia, L. Singh, and S. Sisodia, 2012). Some of the main and basic clustering algorithms are briefly explained below. These basic algorithms nowadays are being used for further advanced and hybrid clustering techniques development.

2.2.2 Clustering Types

1. Partition-Based Clustering (K-mean clustering algorithm)

- *Process of the algorithm*

The K-mean clustering algorithm is a typical clustering algorithm used for partitioning a large set of data or large networks. It is one of the simplest non-supervised learning algorithms applied as solution for the problems with well-known clusters (Sun Jigui, Liu Jie, and Zhao Lianyu, 2008). K-mean clustering algorithm is a partitioning based algorithm, the basic idea of this algorithm is to classify the input data into K clusters by iterative and local minimum convergence (Divya, Deepika, Yamini, and Sobiyaa, 2019).

It consists of two phases. The first phase is specifying K centres randomly, where K is a constant number fixed in advance. The second phase is the inclusion of each data point to the cluster with the closest centre (Fahim A M, Salem A M, and Torkey F A, 2006). It is generally accepted that the Euclidean distance measures the distance between data points to the centres of each cluster. The first step of the k-mean algorithm is the early grouping which is done if all nodes (data points) are included in clusters. For these groups, the average is iteratively recalculated until the criterion function is minimal. Equation 3 represent the criterion function; where E is the objective function, K is the number of clusters and n is the number of iterations.

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} \left| x_j - x_i \right|^2 \quad \ldots\ldots\ldots\ldots (1)$$

- *Advantages*

  - Easy implementation since it is very simple clustering algorithm.
  - The algorithm has only few calculations, computing and comparing distance among data points. Because of that it can be faster in computing than the other clustering techniques, with a time complexity of O (n), n is the number of the data samples.
  - It can scale up to enormous data set.
  - Furthermore, it can likewise effectively adjust to new data sample.

- *Disadvantages*

  - The results of the clustering vary according to the initial values, also the random selection of the centroids. Therefore, the outcomes vary from one execution to another, lacking inconsistency.
  - The K-mean algorithm doesn't have the ability of identifying the outliers. The clustering process get effected by the data set outliers or noise, as the outliers might drag by a cluster or a cluster may be created by the outliers themselves (Sabiha Firdaus and Md. Ashraf Uddin, 2015).

2. Density Based clustering (DBSCAN algorithm)

The (DBSCAN) Density Based Special Clustering of Applications with Noise is the most known algorithm of density based algorithms. Mainly two parameters have to be predefined to perform the clustering. The first parameter is the neighbourhood maximum radius from the point of observation (ε). Two nodes are assumed to be neighbours if the distance that separate them is equal or less than ε. The second parameter is the minimum number of nodes threshold (MinPts) to define the clusters, specifying an appropriate threshold value is very important to increase the accuracy of the clustering (Rodriguez and Laio, 2014).

In DBSCAN there are three kinds of points. The first point is the core point that has at least several nodes equal to MinPts included within its neighbourhood radius ε. The second point is the border point that has several points that are less than MinPts points included in its neighbourhood radius ε, and it is reachable by the core point. The third point is Noise point the outlier nodes that are not included in any cluster. This algorithm is able to find arbitrary-shaped clusters.

The DBSCAN works only with highly separated clusters, otherwise, clusters may be erroneously merged.

A. Advantages

- The DBSCAN algorithm works well with the data sets consisting of low density versus high density clusters and can define them separately.
- This algorithm handles the data set outliers and it is resistant to noise.
- Unlike K-mean clustering, this algorithm does not require a prior definition for number of clusters, although, the estimating of ε and MinPts becomes challenging.

B. *Disadvantages*

- Although this algorithm has the ability of separating the high and low density clusters, it does not perform well with clusters of changing or similar densities (Sabiha Firdaus and Md. Ashraf Uddin, 2015).

3. connectivity- Based clustering (Hierarchical algorithm)

This clustering is a method that creates clusters in the form of a hierarchy or a tree. The hierarchy clusters are known as dendrograms. Two methods are proposed to perform the clustering, the divisive method, and the agglomerative method (Nisha and Jai Kaur, 2015).

The divisive or top-down, this method divides the initially given large data into smaller groups known as clusters, until reaching a predefined threshold. The agglomerative or bottom-up, this method merges the initially given clusters where the clusters that merged are very similar to each other. This merging continues until the large clusters form. This method is the opposite of the divisive method (Aastha and Rajneet, 2013).

A. Advantages

- It is not necessarily required to predefine the number of clusters.
- Easy to implement, like K-mean clustering.
- It creates the structure of the cluster tree (dendrogram) hierarchically, which helps in determining the number of clusters.

B. *Disadvantages*

- It has the highest relatively time complexity comparing with other clustering algorithms of O (n^2logn), where n is the number of data points.
- It may be sensitive to noises and outliers.
- Backtracking is not stimulated in this algorithm, so it cannot be modified when the cluster is created.

2.3 Reliability

The system reliability idea is very simple, it is the measure of how good is the system and how often it goes down (Shooman, pp.283-288). The complexity of the system causes real problems in reliability evaluation. An efficient technique is used to evaluate the reliability of complex networks with a massive number of nodes and links, by dividing the network into several sub-networks (partitions, clusters) using partitioning techniques. The measure of reliability of the component is its probability, and the last vary between (0 – 1), the zero means that the component is unreliable while the one means that the component is fully reliable.

The nodes are assumed to be fully reliable, (nodes probability = 1), while the probability of the edges varies. The unreliable (failing) nodes or edges can be transformed into non-failing nodes or edges. Usually, this transformation is used with unreliable nodes as long as the network is assumed to have perfect nodes and unperfected links. If a graph has a failing node, it is represented by two perfect nodes and one unreliable link, whose probability is equal to the probability of the unreliable node.

If the calculations of reliability are done for the communication between pair of nodes one of them is the source node and the other is the target node, the reliability is called two-terminal reliability. If it measures the communication among all nodes, then it's called all-terminal reliability (Shooman, pp.302-308). Many methods are proposed to estimate the all-terminal reliability, some of the methods are briefly explained below.

a. Event – Space Enumeration

The simplest means of evaluating the all-terminal network reliability is the event-space enumeration. This method enumerates all possible combinations where the edges can be good or bad treating these combinations as events. Bad events are those events where all terminals are not communicating correctly, while good events are those events where all the nodes are communicating correctly.

The reliability of the network is the sum of the events that create path between source and destination points. For all terminal reliability all the nodes have to be taken as source and all nodes have to be taken as destination. This method usually is used for small network, but not practical for big network because it requires large number of mathematical operations.

b. Cut-Set and Tie-Set Methods

Cut-set and tie-set are two methods of calculating the all-terminal reliability. The tie sets are the connection paths between single nodes to all other nodes in the network, while cut sets are the interrupted path between single nodes to all other nodes in network.

Finding the results using these methods can be exact or approximate. In the exact calculations, network model programming is used to find the final result. While the approximation is used in the case of hand calculations, by eliminating the similar cut-sets during the calculating process.

c. Graph Transformations

The graph transformation is the most commonly used method to evaluate the all-terminal reliability. This method uses combinations of series, parallel and Y-D transformations to reduce the number of nodes in the network into only two nodes (source and target). The reliability of the edge (source- target) is the all-terminal reliability of the network.

2.4 Mobile Communication Network

a. Mobile communication network design

The design of the mobile communication network commences with reorganizing the area into basic geographical units called "cells". These cells are symmetrical in shape. A cellular network is used to provide both radio coverage and capacity for the cells that contain the mobile phone operators. Large geographic areas are split into smaller cells to avoid line-of-sight signal loss and to support a large number of active phones in that area achieving the optimal coverage.

Basically, the network starts with the Mobile Station (MS), which represents the mobile phone of the user. In every cell, there is several Mobile Stations (MSs) and one transmitting/receiving equipment called Base Transceiver Station (BTS) in the centre of the cell. The air or radio interface (UM interface) is used to exchange signals between the MS and the BTS.

In each Radio Station System (RSS) there is a number of cells that contains number of BTS's all controlled by a single Base Station Controller (BSC). The A-bis is the internal interface that links BSC to the BTS's. The whole network is managed by the (MSC) Mobile Switching Centre, A-interface provides the communication between MSC and the RSS (Davies, 2016). The block diagram in Figure 9 shows a single operator hardware requirement, while Figure 10 shows the GSM network.

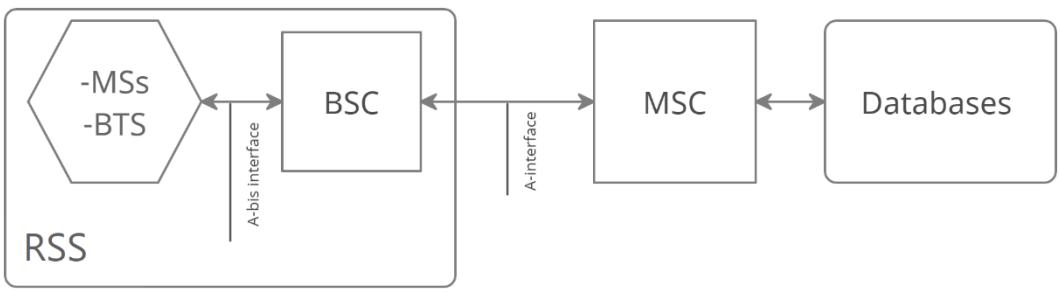

Figure 9. Infrastructure of a single operator
in a mobile communication network

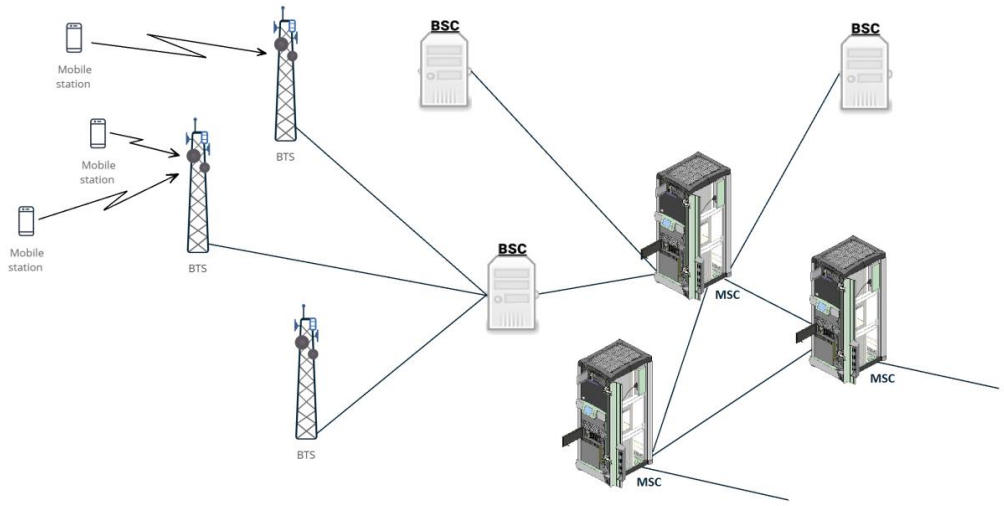

Figure 10. GSM network.

b. Graph theory for wireless communication system

Graph theory can easily model a mobile communication network as long as the mapping between the elements of the network and the graph is concerned.

- Nodes represent the mobile station and the base stations of the communication network.
- Edges are the side connections between the MS's (Device-to-Device) and between the MS to the BS.

CHAPTER THREE
METHODOLOGY AND IMPLEMENTATION

The improvement proposed in this work is adding a constraint for the classical K-mean clustering algorithm to have a more accurate clustering technique for a wide range of applications. As the applications various the added constraints various as well, like connectivity constraint in case of wired networks, security constraint in case of using codes to connect to a specific cluster, availability constraint, reliability constraint, and many others.

The second part is reliability evaluation for the network after applying the improved clustering algorithm to create clusters.

3.1 Comparison of Clustering Algorithms

A comparison has been made among the three clustering techniques mentioned previously according to their advantages and their time complexity to choose the most suitable technique that suits the proposed improvement.

Table 1. Time complexity of clustering techniques.

Clustering type	Algorithm	Time Complexity
Partitioning	K-mean	O(n)
Hierarchical	Agglomerative and divisive	$O(n^2 \log n)$
Density-Based	DBSCAN	$O(n^2)$

It can be seen from table 1. that the Hierarchical clustering algorithm has the highest time complexity and because there is no backtracking for the clusters which means once the node is included in a cluster it cannot be moved around, the hierarchical clustering is not suitable for this work.

Difficult estimation of ε and MinPts in DBSCAN is a major drawback for this work, and furthermore, this algorithm cannot handle clusters of varying density despite time complexity.

Since it is the simplest and fastest algorithm to implement, it can be seen that the best algorithm for this work is the K-mean clustering algorithm.

3.2 K-mean clustering algorithm

- *The steps of the algorithm*

1. Cluster center initialization; randomly select a number of points equal to the number of clusters.
2. Assign all the nodes to the clusters according to the closest distance (Euclidian distance) between nodes and the cluster center assigned.
3. Find the central mass of each cluster by dividing the sum of all nodes' points by the number of nodes in the specified cluster, and update the cluster distance according to the nodes assigned to them.
4. Repeat step 2 and 3 until convergence.

K-mean algorithm

Input:

- N: number of nodes, K: number of clusters.

Output: K clusters.

1: Select K random points denoted as centroids (C_j).
2: **Repeat.**
3: Calculate distance between all nodes to all centroids.
4: Include N_i to the nearest cluster.
5: **For** j=1 to K **do**
6: Reassign the j'th centroid.
7: **End for**
8: **Until** convergence.
9: **Return.**
10: Create distance to centroids matrix (DTC).
11: Create Y matrix with K pages to save each cluster nodes.

This method produces exactly k different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The objective of K-Mean clustering is to minimize total intra-cluster variance, or, the squared error function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \qquad (2)$$

Where:

k= number of clusters, n= number of cases, c_j= centroid for cluster j.

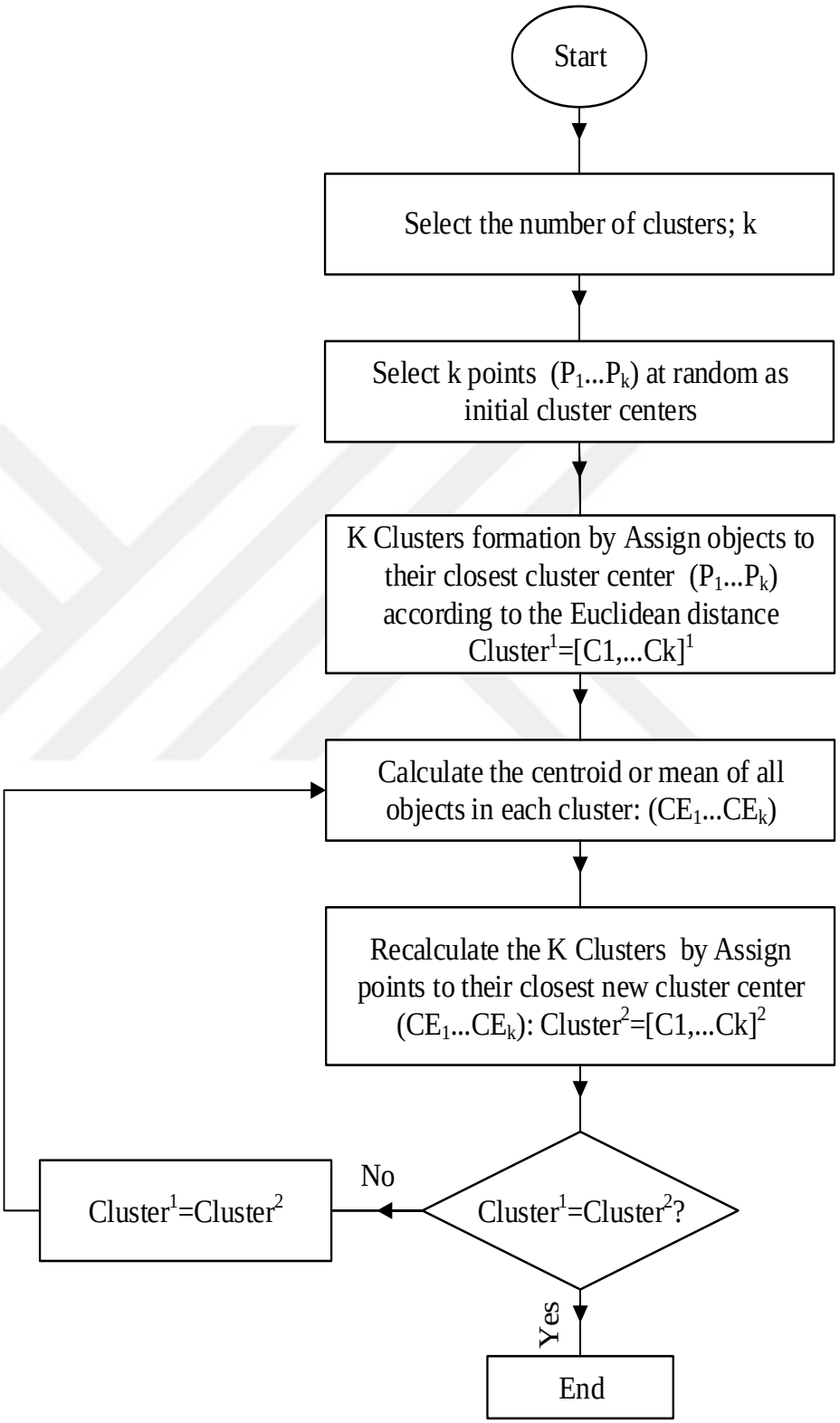

Figure 11. K-mean clustering algorithm flowchart.

3.3 Improved K-mean clustering algorithm

The standard K-mean clustering is working efficiently in the case of a wireless network because this algorithm mainly depends on the distances between different nodes. In the general case of wired or heterogeneous networks, a new constraint is introduced; network connectivity. The k-mean algorithm fails to provide a solution for this problem. The new improvement deals with this limitation in the k-mean algorithm by finding a solution for this drawback point.

This work provides an improved algorithm based on the k-mean algorithm that takes into consideration the connectivity of the network nodes. After the final definition of cluster centroid at the end of the application of k-mean, a new stage begins by checking the connectivity of all nodes, one by one.

- *The steps of the modified algorithm*

1. Define the matrix Distance to Centroid = DTC [N×K] where N is the network nodes number and K is the predefined cluster number. The elements of this matrix give the distances between nodes and centroids. For example, DTC [7, 3] represents the distance between node (7) and the centroid of the cluster number (3)).
2. Derive the Yr [N,N] matrix for each cluster (r) from M[n,m]- connectivity matrix-generated from the source and target vectors given at the beginning.
3. Create a matrix that shows the links in each cluster, YY matrix.
4. For all nodes one by one: CHECK the connectivity with nodes in the same cluster. If there is at least one connectivity, the node is marked as ACCEPTED-NODE in this cluster otherwise it is marked as REJECTED-NODE from this cluster.
5. Depending on the connectivity matrix of that cluster (j). If the summation (Rj) of the specified node raw elements in the Yj matrix is not equal to zero that means the node is in that cluster (j). If the summation is equal to zero, a move to the next cluster has to be taken by checking the summation of the raw of the same node in the next cluster and so on until the summation is not equal to zero.

$$R_j = \sum_{n=1}^{N} Y_j[i,n] \qquad (3)$$

6. For the exit nodes in (j), connectivity has to be checked. If the summation is greater than one (1) the node is ACCEPTED-NODE. If the summation is equal to one the node is REJECTED-NODE.
7. In some cases, the node is a single node in the cluster. A summation has to be checked for the diagonal elements of the cluster connectivity matrix Yk[N, N]. If the summation is not greater than two, in this case, the node is marked as REJECTED-NODE, and if there are only two nodes in the cluster the same procedure has to be applied remarking them as REJECTED-NODEs. Taking into consideration the minimum number of nodes in each cluster has to be more than two.

$$T_j = \sum_{n=1}^{N} Y_j[n,n] \qquad (4)$$

8. For ACCEPTED-NODE the clustering process is finished.
9. If a node is marked as REJECTED-NODE, a very high value is given for the DTC [i, j] belongs to the cluster j and node i that is currently being checked, and that node has to be eliminated from the Yj matrix of the cluster j. Then the next candidate cluster from the matrix DTC that has the second minimum distance from the same line in the matrix must be checked.

$$min = \min_{n=1\ to\ K} DTC[i,n] \qquad (5)$$

10. Again the process of CHECK is repeated for this node but with nodes in the second candidate cluster. This will continue until all nodes are marked as ACCEPTED-NODE.
11. In the case of more than one REJECTED-NODE (two nodes connected to each other but belong to another cluster). The YY matrix has to be checked, if the two nodes are only connected to each other, the two nodes have to be denoted as REJECTED-NODES and the DTC for both nodes has to be checked in order to find the nearest cluster the two nodes belong to.
12. Figure 13. shows the flow chart of the process.

Modified K-mean algorithm

Input:

- N: number of nodes, K: number of clusters.
- $X = \{(x_i, y_i): 1 \leq i \leq N\}$ nodes locations matrix.
- Source (s) and target (t) vectors, and the corresponding weight of each link.

Output: K clusters.

1. K clusters created by K-mean clustering.
2. Crate network topology matrix M.
3. **For** i=1 to K **do**
4. **For** j=1 to number of node in i cluster **do**
5. Check M for single nodes connectivity.
6. **If** connected **then**
7. stay.
8. **End if**
9. **If** single node not connected **then**
10. check DTC for nearest cluster.
11. move node to nearest connected cluster.
12. update CLUSTER[i].
13. **End if.**
14. **End for.**
15. **End for.**
16. create YY matrix. YY: source, destination and weight of links for each cluster.
17. **For** i=1 to K **do**
18. **For** j=1 to number of node in cluster i **do**
19. Check YY matrix for all nodes connectivity.
20. **If** all nodes connected **then**
21. Stay.
22. **End if**
23. **If** two nodes are not connected to all, or only two nodes in cluster **then**
24. Check DTC of both nodes for nearest cluster.
25. Move the nodes to nearest connected cluster.
26. update CLUSTER[i].
27. **End if.**
28. **End for.**
29. **End for.**

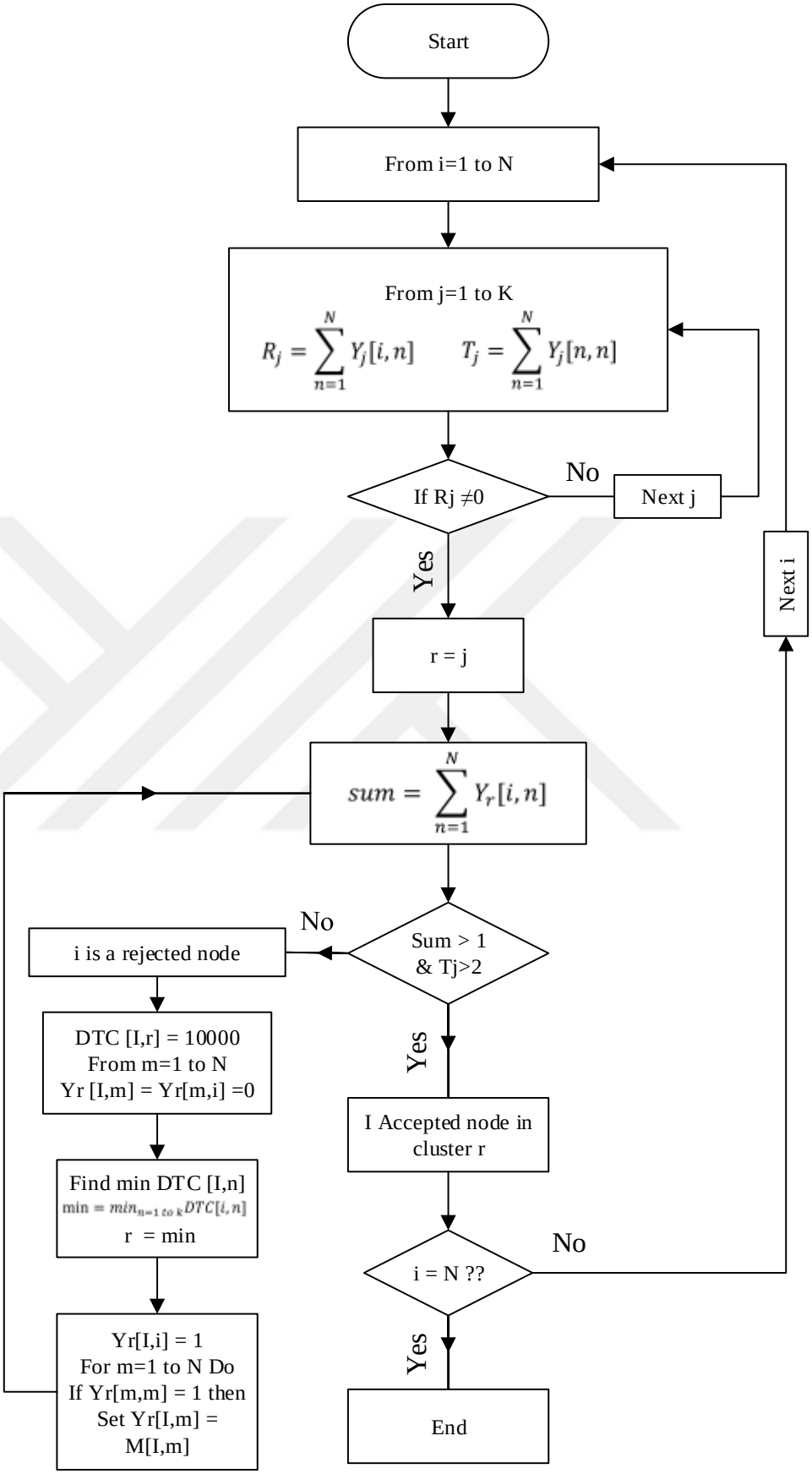

Figure 12. Improved K-mean clustering algorithm flowchart.

3.4 Improved Algorithm Implementation

From the steps of the algorithm three different cases can be seen:

Case1: single node belongs to another cluster

Figure 13. shows a network with 11 nodes. The optimum clusters for this network have to be found, specifying that the number of clusters is 2.

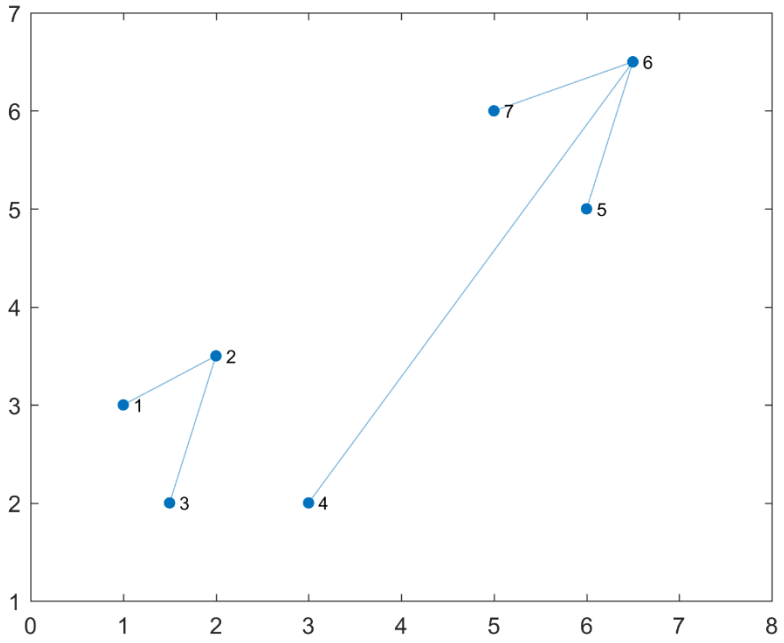

Figure 13. network 1

The required entries are the number of nodes (N) =7, the number of clusters (K) = 2, the nodes' geographical locations as (x,y) coordinates, see table 2. Also, the vectors of source (s = {1, 2, 4, 5, 7}), target (t = {2, 3, 6, 6, 6}), and weight (w= {1, 1, 1, 1, 1}) have to be entered in order to generate the connectivity matrix (M) automatically.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Table 2. Nodes location of network 1.

Node	1	2	3	4	5	6	7
X	1	2	1.5	3	6	6.5	5
Y	3	3.5	2	2	5	6.5	6

Step 1:

Regular K-mean clustering is the first step applied on the network, specifying centres for each cluster (2 clusters means 2 centres) named as centroids. The Euclidean distance has been used for 5 replicates (the number of replicates is specified in advance), also, a number of iterations are applied in each replicate to find the best distance.

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

Where: p and q are points in the network.

Replicate 1, 2 iterations, total sum of distances = 8.

Replicate 2, 1 iterations, total sum of distances = 8.

Replicate 3, 1 iterations, total sum of distances = 8.

Replicate 4, 1 iterations, total sum of distances = 8.

Replicate 5, 2 iterations, total sum of distances = 8.

Best total sum of distances = 8

Centroid 1: X = 6, Y = 6

Centroid 2: X = 1.75, Y = 2.5

The K-mean clustering process is finished and two clusters are generated as a result of this step. In this work, a multidimensional matrix (Y) is created. The matrix Y consists of a number of pages equal to the number of clusters, and each page contains the connectivity matrix for one of the clusters

The result of this step shows that node 4 belong to cluster 1 according to its geographical location, yet the node is not connected to any other nodes in that cluster. In the other hand node 4 is connected to node 6 that belongs to cluster 2. Node 4 must belong to the cluster to which it is connected to at least one node, and this is where the modification works, as seen in step two.

Step 2:

This step is to apply the modified part of the algorithm:

1. Define DTC, the distance between nodes to the centroids of each cluster as shown in table 3. The Euclidean distance is used to calculate the elements of the DTC matrix.

Table 3. Networks 1 distance to centroids

DTC	1	2	3	4	5	6	7
K=1	5.8310	4.7170	6.0208	5.0000	1.0000	0.7071	1.0000
K=2	0.9014	1.0308	0.5590	1.3463	4.9308	6.2099	4.7762

2. Drive the $Yk[7,7]$ for each cluster resulted from the K-mean clustering :

$$Y_1[7,7] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Y_2[7,7] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

3. Create the YY matrix (multidimensional, has pages equal to the number of clusters) that contains the links and their weights for each cluster.

$$YY_1 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$YY_2 = \begin{bmatrix} 5 & 6 & 1 \\ 7 & 6 & 1 \\ 4 & 6 & 1 \end{bmatrix}$$

4. Connectivity Check procedure
   a. For node 1, the existence of this node in the clusters is checked by finding the summation of the first raw for each cluster connectivity matrix $Y_{1\ and\ 2}$ [7,7].

$$R_{1\,to\,2} = \sum_{n=1}^{7} Y_{1\,to\,2}[1,n]$$

$$R_1 = 1+1+0+0+0+0+0 = 2 > 0$$

$R_1 \neq 0$ … node1 is belonging to this cluster (existed)

The summation of the diagonal elements of cluster 1 (T1) must be calculated to see if the node is isolated.

$$T_1 = \sum_{n=1}^{7} Y_1[n,n]$$

$$T_1 = 1+1+1+1+0+0+0 = 4$$

$T_3 \neq 2$ … node is not isolated.

The summation R_1 is greater than two (R_1>2) which means the nodes are more than two and connected to other nodes in cluster 3 so the node is identified as ACCEPTED-NODE. For nodes 2,3,5,6 and 7 the exact same procedure has to be applied.

b. For node 4, the existence of the node in clusters has to be checked.

$$R_{1\,to\,2} = \sum_{n=1}^{7} Y_{1\,to\,2}[4,n]$$

$$R_1 = 0+0+0+1+0+0+0 = 1 > 0$$

$R_1 \neq 0$ … node is belonging to this cluster (existed)

But the summation R_1 is not greater than one which means the node is not connected in cluster 1 the node is identified as REJECTED-NODE.

Eliminate node 4 from $Y_1[7, 7]$, and give the DTC [4,1] = big number (999)

$$Y_1[7,7] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 4. Modified distance to centroids for network 1.

DTC	1	2	3	4	5	6	7
K=1	5.8310	4.7170	6.0208	999	1.0000	0.7071	1.0000
K=2	0.9014	1.0308	0.5590	1.3463	4.9308	6.2099	4.7762

Find the minimum number (*min*) of DTC [4, k],

$$min = \min_{n=1\ to\ 2} DTC[4,n]$$

min = 1.3463 … which it is cluster 2, accordingly the Y2 matrix has to be rearranged adding node 4 to cluster 2 as follows.

$$Y_2[7,7] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

connectivity is checked again, but with cluster 2,

$$R_2 = 0 + 0 + 0 + 1 + 0 + 1 + 0 = 2 > 1$$

Node 4 is connected in cluster 2 and now belong to that cluster denoted as ACCEPTED-NODE.

c. For node 11, the existence is checked for the node in the clusters.

$$R_{1\ to\ 4} = \sum_{n=1}^{11} Y_{1\ to\ 4}[11,n]$$

R1≠ 0 … node is belonging to this cluster (existed)

Isolation for this node has to be checked.

$$T_1 = \sum_{n=1}^{11} Y_1[n,n]$$

$T_3 = 1$ … node is isolated.

The node is isolated and ACCEPTED-NODE, check is done for this node.

5. All nodes are checked and all are ACCEPTED-NODEs.

Case2: two connected nodes belong to another cluster

If the same network of Figure 14 has one more node connected to node 4 as shown in Figure 14. In this case, there are two nodes connected to each other, isolated from the other nodes in the cluster. The two nodes have to be denoted as REJECTED-NODEs.

The YY matrix is used to check the connectivity, and the DTC matrix is used to specify the nearest cluster.

$$YY_1 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 4 & 8 & 0 \end{bmatrix}$$

$$YY_2 = \begin{bmatrix} 5 & 6 & 1 \\ 7 & 6 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

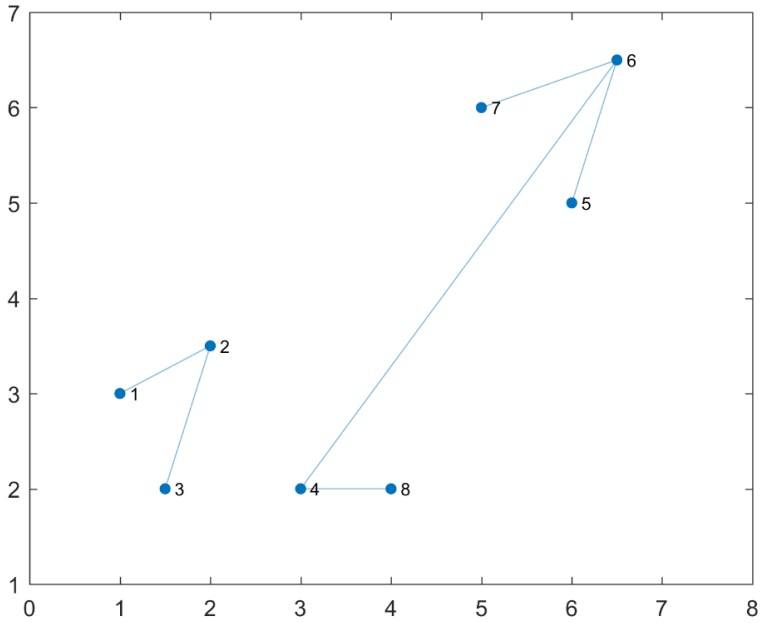

Figure 14. Network 2

It can be seen from YY1 that node 4 and 8 are connected only with each other, so the two nodes have to be moved to cluster 2, and the connectivity and isolation have to be checked.

3.5 Reliability evaluation based on Tie-set method

The definition of the Tie-set algorithm is the minimal path between the input and the output of the system without passing over a node twice. It is a well-known reliability evaluation algorithm. Generally, it is used to find the two-terminal reliability but also used to find the all-terminal of simple to medium networks.

A hybrid algorithm is proposed by K.Mahmood, Myderrizi based on the Tie-set method that is assumed to be more efficient than the classical reliability evaluation algorithms. Two steps take place in the proposed work (partitioning considerations and the Hybrid Algorithm).

The network must be clustered in several clusters (partitions) using the improved algorithm. The nodes in the clusters according to K.Mahmood, Myderrizi's work are classified into three types of important nodes (source node, destination node, contact nodes). The network clusters must be simplified to have clusters containing only the mentioned three types of nodes. The reliability inside the partition is calculated using an R-algorithm (the Tie-set algorithm has been used). The contact nodes are the points that connect two or more partitions.

To simplify any network topology, the three suggested rules need to be applied to partitions:

- If the partition has neither source nor destination nodes, the partition is called Intermediary Partition (IPA). The simplified form of this partition is gained by replacing all clusters nodes with only the contact nodes and their links. The reliability of the cluster can be calculated using R-algorithm.
- If the specified partition has either source or destination nodes, the partition is called One End Partition (OEP). In this case, the simplified partition can be obtained by replacing all the nodes with the contact and source/destination nodes and their links. R-algorithm can be used to calculate the reliability of the cluster.
- If the partition has both the source and destination nodes, the partition is called the Complete Partition (CP). The simplified partition contains the contact nodes, source node, and destination node. In this case, the reliability that can be computed by the R-algorithm is the reliability of the network regardless of other clusters.

After applying the previous three rules, a simplified network is generated. The same procedures have to be applied to the simplified network. Until the simplification results in a single cluster that contains source node, destination node, and a total number of nodes equal to or less than 10.

CHAPTER FOUR
IMPROVED ALGORITHM APPLICATION

4.1.General

This chapter presents the practical experiments of the adopted algorithm, and discusses the obtained results. The obtained results are compared with the classical K-mean clustering results. The network simulator is the best solution to test the algorithms or applications built to achieve a specific task. The aim of a simulator is to provide a virtual network environment alternative to the real network environment. In this work, MATLAB is used to test the proposed algorithm.

4.2.Experimental Results.

The evaluation of the proposed algorithm is carried out by applying it to number of networks that contain a different number of nodes and different connectivity's. Each example explains the different propriety of the improved algorithm. This section shows the verification of the effectiveness and feasibility of the proposed improved algorithm.

4.2.1. Example 1: Single node belongs to another cluster and isolated node.

Figure 15 shows a network of 11 nodes. According to the classical K-mean clustering algorithm the network has to be clustered into 4 clusters as shown in Figure 16. It can be seen that node 4 is in the second cluster C2, but according to the connectivity, it must belong to the third cluster C3 (a single node belongs to another cluster). And node 11 is a single node in the first cluster C1 (isolated node) connected only to node 8 in the second cluster C2.

Those nodes (4, 11) are denoted as REJECTED-NODEs. It is assumed that the minimum number of nodes in each cluster must be greater than 2. Empty cluster (C1) has to be eliminated, as a result only three clusters are created as shown in Figure 17.

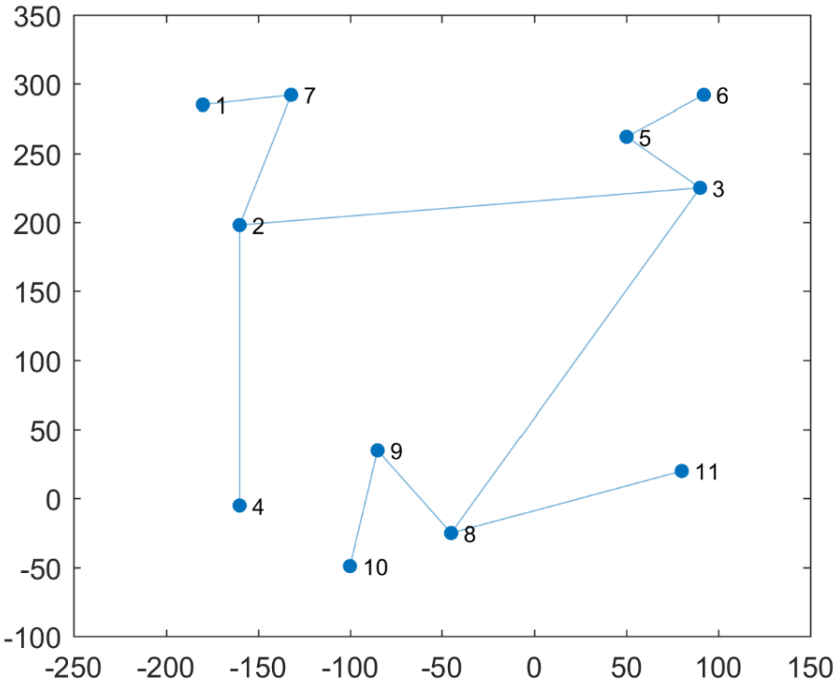

Figure 15. network model of example 1

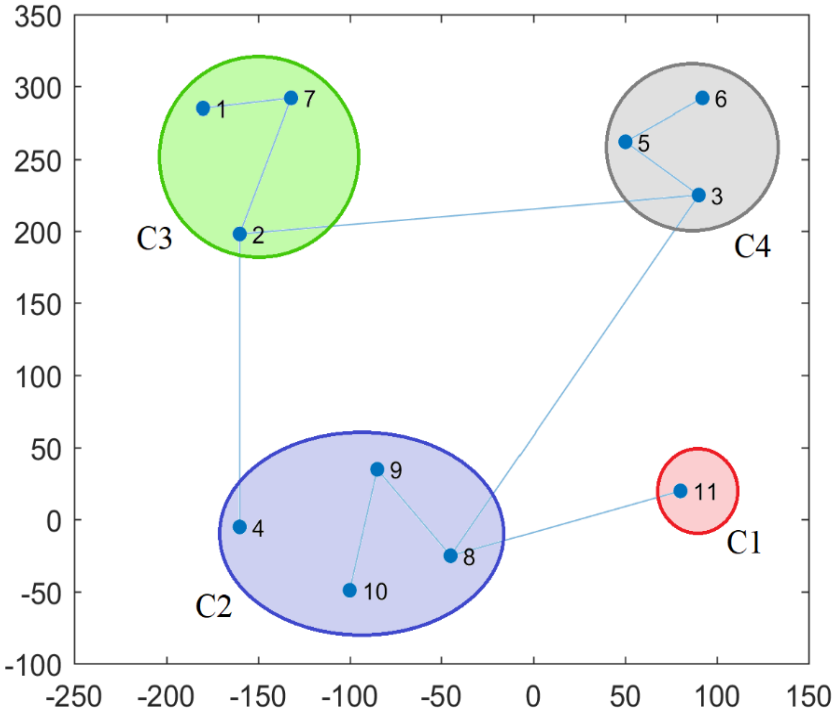

Figure 16. K-mean clustering result of example 1

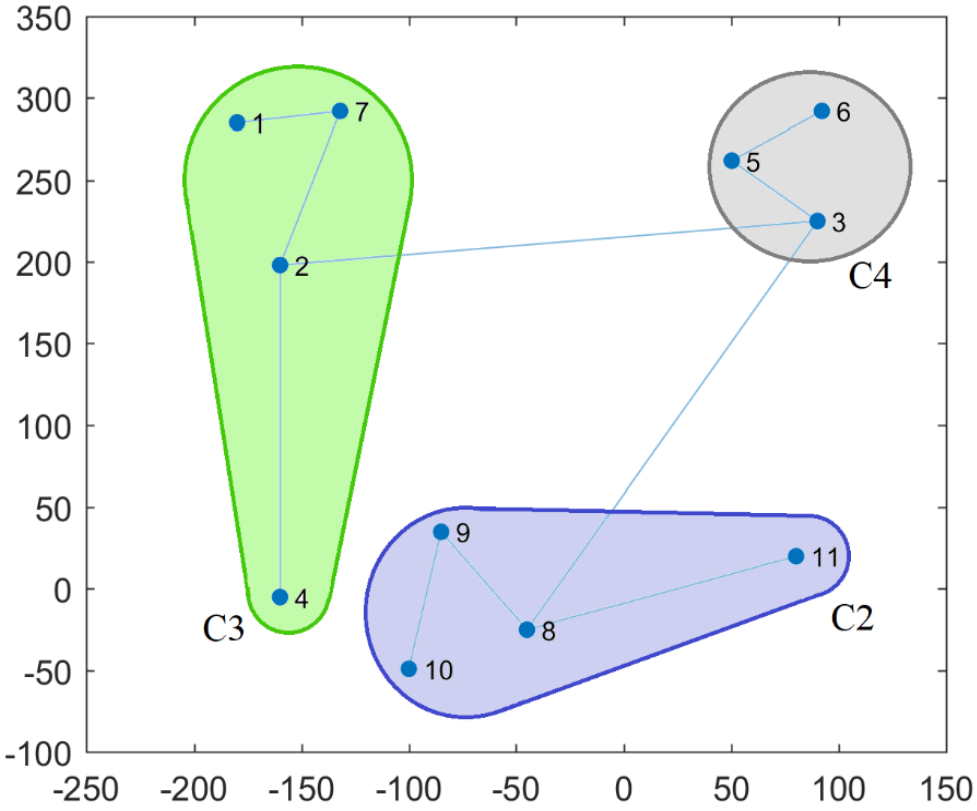

Figure 17. Improved K-mean clustering result of example 1.

For this example, network reliability has been evaluated between node 1 as the source and all the other nodes as destinations. The results are shown in table 5.

Table 5. Reliability of example 1.

Source-destination	Reliability evaluated
1-2	1
1-3	0.9990
1-4	0.9990
1-5	0.9990
1-6	0.9980
1-7	1
1-8	0.9990
1-9	0.9980
1-10	0.9970
1-11	0.9980

4.2.2. Example 2: Number of single nodes belong to another cluster and isolated node.

The more complex network with 15 nodes is shown in Figure 18. By using the classical K-mean clustering algorithm 5 clusters are created as shown in Figure 19.

It can be seen that nodes (4,5, and 14) are in different clusters but are connected only to nodes in the first cluster C1, also node 8 is connected to node 10 in the second cluster C2, and node 7 is connected to node 6 in the third cluster C3 (number of single nodes belong to another cluster).

While node 15 is a single node in cluster C4, resulting in an empty cluster has to be eliminated.

All those nodes are denoted as REJECTED-NODEs. Figure 20 shows the clusters obtained after applying the improved algorithm.

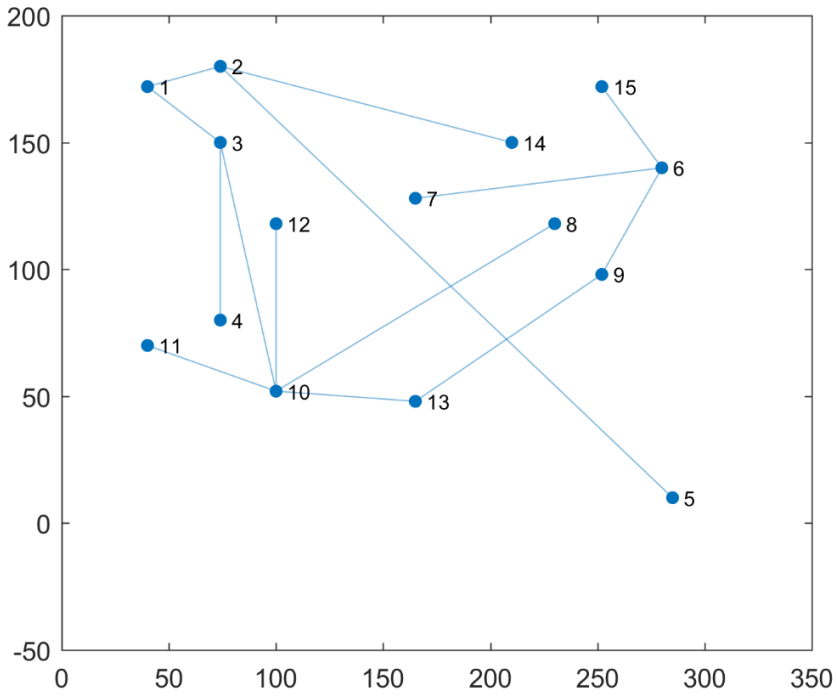

Figure 18. Network model of example 2.

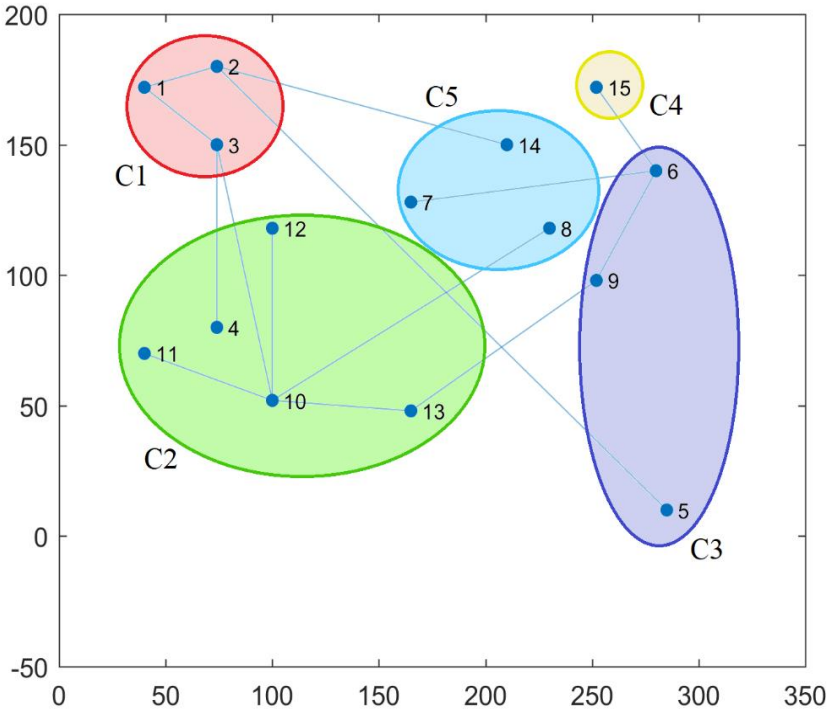

Figure 19. K-mean clustering result of example 2.

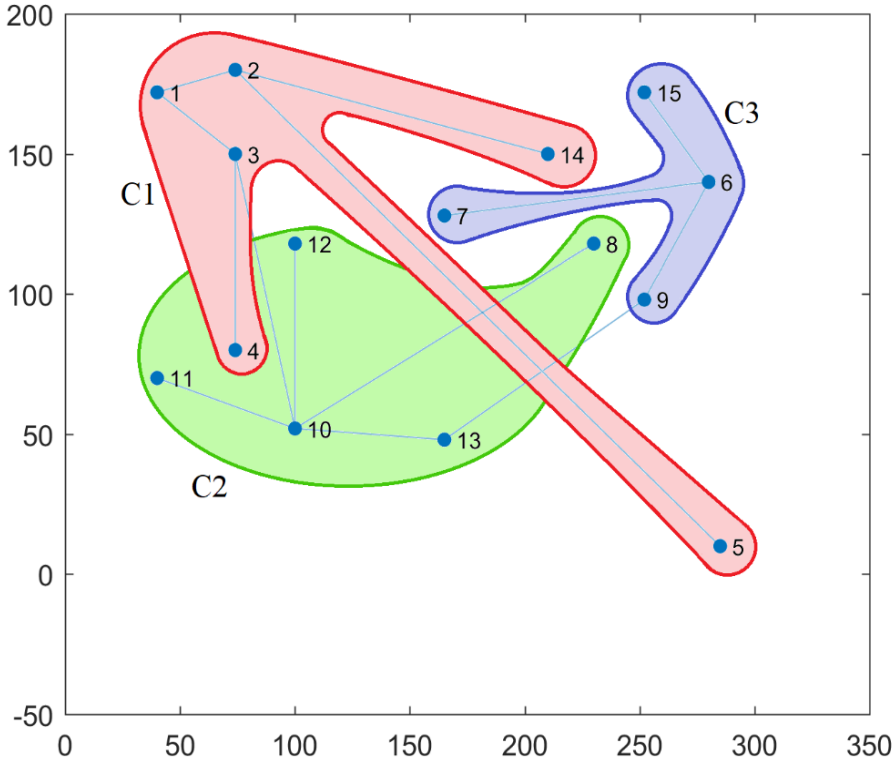

Figure 20. Improved K-mean clustering result of example 2.

For this example, the reliability has been evaluated between node 1 as the source and all the other nodes as destinations. The results are shown in table 6.

Table 6. Reliability of example 2.

Source-destination	Reliability evaluated
1-2	0.9729
1-3	0.9000
1-4	0.9639
1-5	0.9347
1-6	0.5905
1-7	0.5314
1-8	0.7290
1-9	0.6561
1-10	0.8100
1-11	0.7290
1-12	0.7290
1-13	0.7290
1-14	0.9347
1-15	0.5314

4.2.3. Example 3: Number of single nodes belong to another cluster, two connected nodes belong to another cluster, and isolated nodes.

The most complex network example with 23 nodes is shown in Figure 21. The result of the classical K-mean clustering algorithm is 5 clusters shown in Figure 22.

Nodes (7, 8, 10, and 22) are single nodes that belong to another cluster. And the connected two nodes 11, 12 from cluster C2 belong to the first cluster (two connected nodes belong to another cluster).

Nodes 20 and 21 in cluster C5 are connected, but it is assumed that the minimum number of nodes in each cluster must be greater than 2 (isolated nodes).

All those nodes are denoted as REJECTED-NODEs and they must be moved to the nearest cluster, and cluster C5 has to be eliminated. Figure 23 shows the result of the improved algorithm.

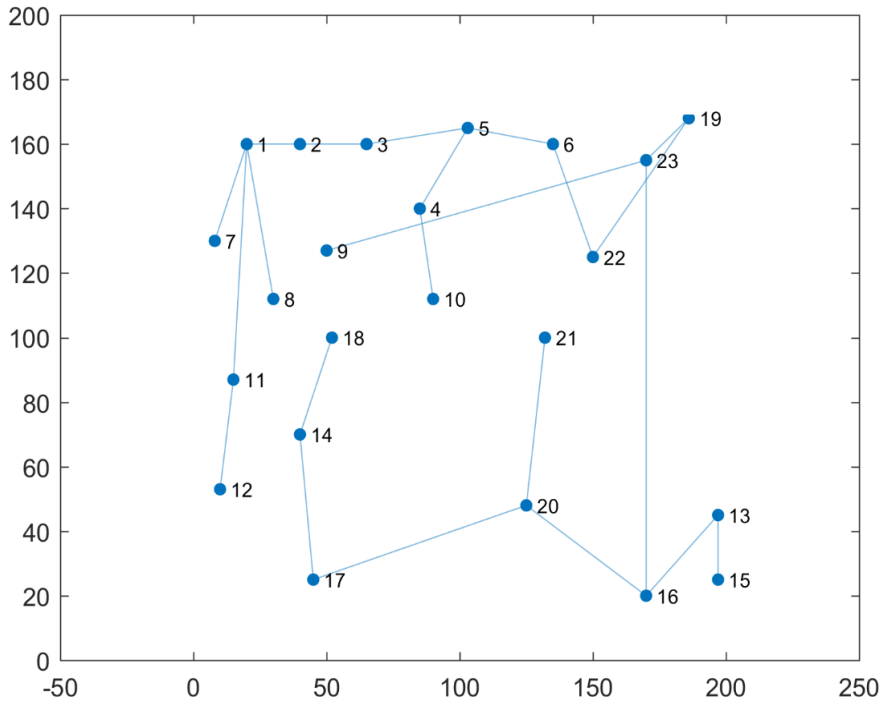

Figure 21. Network model of example 3.

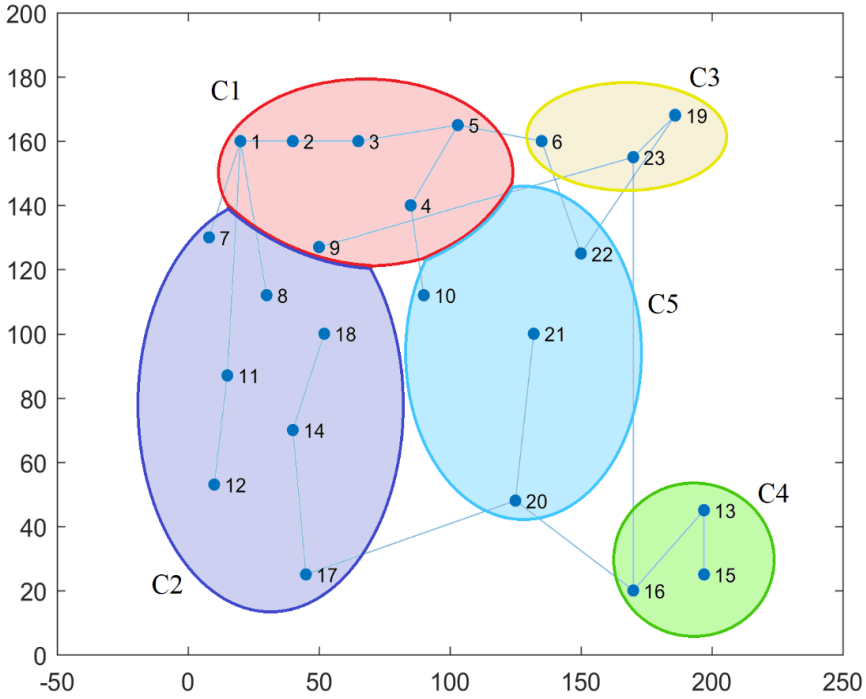

Figure 22. K-mean clustering result of example 3.

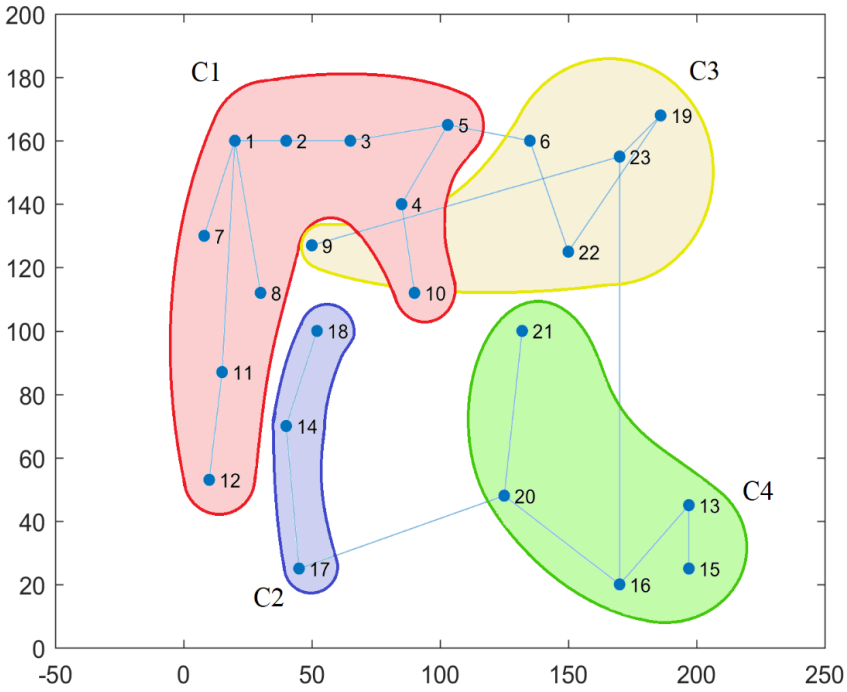

Figure 23. Improved K-mean clustering result of example 3.

Reliability has been evaluated between node 1 as the source and all the other nodes in the network as destinations.

Table 7. Reliability of example 3.

Source-destination	Reliability evaluated
1-2	0.9590
1-3	0.9347
1-4	0.8817
1-5	0.7290
1-6	0.6561
1-7	0.9478
1-8	0.9478
1-9	0.4305
1-10	0.8323
1-11	0.9478
1-12	0.8918
1-13	0.3874
1-14	0.3138
1-15	0.3487
1-16	0.4305
1-17	0.3487
1-18	0.2824
1-19	0.5314
1-20	0.3874
1-21	0.3487
1-22	0.5905
1-23	0.4783

4.3.Results comparison.

The execution time of the classical K-mean algorithm has been calculated besides the execution time of the improved algorithm, to be compared.

$$difference\ of\ executing\ time = \underset{execution\ time}{improved\ algorithm} \_ \underset{execution\ time}{K-mean\ algorithm} \quad (6)$$

$$example\ 1\ execution\ time = 7.971234 - 7.812809$$

$$= 0.158425$$

$$example\ 2\ execution\ time = 9.554213 - 9.358834$$

$$= 0.195379$$

$$example\ 3\ \ execution\ time = 14.079364 - 13.893036$$

$$= 0.186328$$

As can be seen from Table 8, the accuracy of the improved algorithm is much higher than the K-mean algorithm, and a very small difference is observed between execution times.

Table 8. Results comparison.

Execution time table	No. of nodes	Classical K-mean execution time (s)	Improved algorithm execution time (s)	relative time difference
Example 1	11	7.812809	7.971234	1.98 %
Example 2	15	9.358834	9.554213	2.045 %
Example 3	23	13.893036	14.079364	1.323412 %

CHAPTER FIVE
CONCLUSIONS AND FUTURE WORK

5.1. Conclusions

In this study, an efficient clustering algorithm is proposed based on the classical K-mean algorithm, which provides clusters depending on geographical location as well as connectivity. The clustering algorithm simplifies complex networks with large numbers of nodes and reduces the mathematical operations required for other computations. The examples in the previous chapter are provided to illustrate the different proprieties of the improved algorithm that overcame the classical K-mean clustering algorithm accuracy limitation, providing a more accurate algorithm. Also, the reliability has been evaluated using a hybrid algorithm that provide the result with a better computing time.

5.1 Future Works

The following recommendations can be considered for the improvement of the presented work.

1. Using the proposed procedure to improve any other clustering algorithm to increase the clustering accuracy.
2. In the field of mobile communication, applying the proposed algorithm on the cellular network to find the best available connected station in case of an error in one of the stations.
3. Adding a method to find the optimal clusters number depending on the inputs, like the Elbow method.

REFERENCES

Surbhi, B. (2014)." New Improved technique for initial cluster centers of K means clustering using Genetic Algorithm", IEEE International Conference for Convergence of Technology.

Caiquan, X., Zhen, H., Ke, L., Xuan, L. (2016). "An Improved K-means text clustering algorithm by Optimizing initial cluster centers", IEEE, 7th International Conference on Cloud Computing and Big Data.

Shyr-Shen, Y., Shao-Wei, C., Chuin-Mu, W., Yung-Kuan, C., Ting-Cheng, C. (2018). "Two improved k-means algorithms", Applied Soft Computing 68 747–755.

Cheng-Hao, D., Wan-Lei, Z. (2018). " Fast k-means based on k-NN Graph", in 34th International Conference on Data Engineering (ICDE), IEEE.

Lehsaini, M., Benmahdi, M. (2018). "An improved K-means Cluster-based Routing Scheme for Wireless Sensor Networks", in International Symposium on Programming and Systems (ISPS). IEEE.

Guang-Yu, Z., Chang-Dong, W., Dong, H., Wei-Shi, Z., Yu, R. (2018). "TW-Co-k-means: Two-level weighted collaborative k-means for multi-view clustering", in Appl soft compute 68:747-755.

Yang, C., PengFei, H., Weilan, W. (2019). "Improved K-means algorithm and its implementation based on Mean Shift", in 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE.

Chengchang, Z., Yuwen, G., Huayu, Z. (2019). "Research on Massive Data Analysis of Power Utilization Based on Improved K-means Algorithm and Cloud Computing", in International Conference on Intelligent Computing, Automation and Systems (ICICAS), IEEE.

Jinyan, L., Dongqing, W., Zhuo, M., Boyang, X. (2019). "Improved K-means clustering algorithm for screw locking classification", in IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).

Rajdeep, B., Anindya, D., Anannya, G., Arindam, M., Ranjan, B., Tapas, H. (2019). " An Empirical Evaluation of k-Means Clustering Technique and Comparison", IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India.

Su-yu, H., Bo, Z. (2019). "Research on improved k-means clustering algorithm based on Hadoop platform", in International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), IEEE.

Divya, V., Deepika, R., Yamini, C., Sobiyaa, P. (2019). “An Efficient K-Means Clustering Initialization Using Optimization Algorithm”, International Conference on Advances in Computing and Communication Engineering (ICACCE) , IEEE.

Albakay, N., Hempel, M., Alahmad, M., Hamid, M. (2019). “Novel Frequency Offset Estimation Scheme for Reliable Wireless Communication using Modified K-Means Clustering” in 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE.

Dong, G., Jin, Y., Wang, S., Li, W., Tao, Z., Guo, S. (2019). “DB-K means: an intrusion detection algorithm based on DBSCAN and K-means”, In 20th Asia-Pacific network operations and management symposium (APNOMS), IEEE.

Siwei, W., En, Z., Jintao, H., Miaomiao, L., Kaikai, Z., Ning, H., Xinwang, L., (2019). “Efficient Multiple Kernel K-means Clustering with Late Fusion”, IEEE ACESS, in press.

Kristina, P., Sinaga, Y. Miin-Shen, Y. (2020). “Unsupervised K-Means Clustering Algorithm”, IEEE ACCESS., in press.

Youjin, R., Yi’an, L. (2020). " Staged text clustering algorithm based on K-means and hierarchical agglomeration clustering", IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA).

Arghyadeep, S., Manjusha, P., Krishna, C. (2020). “Replacing Random Centroid Selection for Kmeans Clustering: A Proposed Algorithm for Improving Clustering Results”, in International Conference on Computer Science, Engineering and Applications (ICCSEA), IEEE.

Yang, L., Shuaifeng, M., Xinxin, D. (2020). “An improved k-means algorithm based on a new cluster center selection method”, IEEE ACCESS., in press.

Wenqiang, G., Zixuan, H., Yongyan, H., Qinkun, X., Jia, J., Lingling, M. (2020). “Environment Parameter Rating Evaluation for Smart Museum Based on Improved K-Means Clustering Algorithm”, in Chinese Control And Decision Conference (CCDC), IEEE.

Chormunge, S., Jena, S. (2014). “Evaluation of Clustering Algorithm for High Dimensional Data Based on Distance Functions,” in International Conference on Information and Communication Technology for Competitive Strategies.

Eslamnezhad, M., Varjani, A. Y. (2014). “Intrusion Detection Based on MinMax k-means Clustering,” in 7th International Symposium on Telecommunications, pp. 804-808.

Nath, R. P. D., Lee, H.-J., Chowdhury, N. K., Chang, J.-W. (2010). “Modified k-means Clustering for Travel Time Prediction based of Historical Traffic Data,” in International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 511 – 521.

Divya, V., Deepika, R., Yamini, C., Sobiyaa, P. (2019). “An Efficient K-Means Clustering Initialization Using Optimization Algorithm”, in International Conference on Advances in Computing and Communication Engineering (ICACCE), IEEE.

Sisodia, D., Singh, L., Sisodia, S. (2012). “Clustering Techniques: A Brief Survey of Different Clustering Algorithms”, in International Journal of Latest Trends in Engineering and Technology (IJLTET).

Sun, J., Liu, J., Zhao, L. (2008). “Clustering algorithms Research”,Journal of Software ,Vol 19,No 1, pp.48-61.

Fahim, AM., Salem, AM., Torkey, FA. (2006). “An efficient enhanced k-means clustering algorithm” Journal of Zhejiang University Science A, Vol.10, pp:1626-1633.

Sabiha, F., Ashraf Md. (2015). “A Survey on Clustering Algorithms and Complexity Analysis”, in IJCSI international journal of computer science, vol.12, no.2.

Rodriguez, A., Laio, A. (2014). “Clustering by fast search and find of density peaks”. Science, 344(6191):1492–1496.

Nisha, JK., Puneet, JK. (2015). "cluster qualitybbased performance evaluation of hierarchical clustering method", in IEEE 1st international cinference on next generation computing technologies.

Aastha, J., Rajneet, K. (2013). “Review: Comparative Study of Various Clustering Techniques in Data Mining”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, p.55-57.

Shooman, ML., “Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design”, pp.283-308],

Davies, R. (2016). “5G network technology Putting Europe at the leading edge,” EPRS (European Parliamentary Research Service).

K.Mahmood, M., Myderrizi, I. (2020). “Reliability Evaluation Using a Clustering Technique Based on Tie-set Method”, in 43rd International Conference on Telecommunications and Signal Processing (TSP), IEEE 2020.

ANNEXES

ANNEXE A. K-mean clustering algorithm.

```
rng default; % For reproducibility
 X = [20 160,40 160, 65 160, 85 140, 103 165, 135 160,
8 130, 30 112, 50 127, 90 112, 15 87, 10 53, 197 45, 40
70, 197 25, 170 20, 45 25, 52 100, 186 168, 125 48, 132
100, 150 125, 170 155];
sp=5;
N=23;
s = [1 1 1 1 2 3 5 5 6 22 19 23 23 16 16 13 20 20 17 14
11 4];
t = [2 7 8 11 3 5 4 6 22 19 23 9 16 13 20 15 21 17 14
18 12 10];
w=[ 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9
0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9];
x= [X(:,1)];
y= [X(:,2)];
plot(graph(s,t),'XData',x,'YData',y);

A=zeros(N);
for i=1:N
    A(i,i)=1;
end
for i=1:length(s)
    A(s(i),t(i))=w(i);
    A(t(i),s(i))=w(i);
end
AR=N;
AC=AR;
G=sparse(s,t,w,AR,AC);
bg=biograph(G);
dolayout(bg);
%view(bg)
for i=1:AR
bg.nodes(i).Position = [X(i,1) X(i,2)];
dolayout(bg, 'Pathsonly', true);
end
%view(bg)
q = size(X);
%figure;
%plot(X(:,1),X(:,2),'.');
%title 'Randomly Generated Data';
opts=statset('Display','final');
[idx,C]=kmeans(X,sp,'Distance','cityblock','Replicates'
,5,'Options',opts);
figure;
```

```
plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12);
hold on;
plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12);
plot(X(idx==3,1),X(idx==3,2),'g.','MarkerSize',12);
plot(X(idx==4,1),X(idx==4,2),'y.','MarkerSize',12);
plot(X(idx==5,1),X(idx==5,2),'c.','MarkerSize',12);
plot(C(:,1),C(:,2),'Kx','MarkerSize',10,'LineWidth',1)
;
%legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Centroids', 'Location','NW');
title('Cluster Assignments and centroids');
hold off;

for i=1:size(C,1)
    display(['Centroid ', num2str(i), ': X1 = ',
num2str(C(i, 1)), '; X2 = ', num2str(C(i, 2))]);
end
```

ANNEXE B. DTC calculation program in MATLAB

```
for i=1:AR%%%%%%%
    for j=1:sp
        dt(i,j)=sqrt((X(i,1)-C(j,1))^2 +(X(i,2)-
C(j,2))^2);
    end
end
dct=[];
for i=1:AR
    for j=1:sp
        m=1;
        for k=1:sp
            if j~=k
                if dt(i,j)>dt(i,k)
                    m=m+1;
                end
            end
        end
        dct(i,j)=m;
    end
end
Dct=[];
for i=1:AR
    for j=1:sp
        Dct(i,dct(i,j))=j;
    end
end
```

ANNEXE C. Improved K-mean clustering.

```
%%%%%%%
Y=[];
k=ones(1,size(C,1));
for j=1:q(1)
Y(k(idx(j)),1:2,idx(j))=X(j,:);
Y(k(idx(j)),3,idx(j))=j;
k(idx(j))=k(idx(j))+1;
end
Y1=Y;
%%%%%%%
kk1=length(Y(:,3,1));
for k1=1:sp
    for i=1:kk1
        m=0;
        for j=1:kk1
            if Y(i,3,k1)~=0 && Y(j,3,k1)~=0
                if A(Y(i,3,k1),Y(j,3,k1))~=0
                    m=m+1;
                end
            end
        end
        if m==1
            mm1=0;
            for kkk=Dct(Y(i,3,k1),:)
                if kkk~=k1
                    mm=0;
                    for ii=1:kk1
                      if Y(i,3,k1)~=0&&Y(ii,3,kkk)~=0
                      if A(Y(i,3,k1),Y(ii,3,kkk))~=0
                            mm=mm+1;
                      end
                      end
                    end
                    if mm1==0
                        if mm>0
                            rr=0;
                            for jj=1:kk1
                                if rr==0
                                    if Y(jj,3,kkk)==0
                                        rr=rr+1;

Y(jj,:,kkk)=Y(i,:,k1);
                                        Y(i,:,k1)=0;
                                    end
                                end
                            end
```

```
if rr==0

Y(jj+1,:,kkk)=Y(i,:,k1);
                                        Y(i,:,k1)=0;
                                    end
                                    mm1=mm1+1;
                                end
                            end
                        end
                    end
                end
            end
        end
end
Y2=Y;
%%%%%%%%%
kk1=length(Y(:,3,1));
for kkk=1:sp
    m=0;
    for i=1:kk1
        if Y(i,3,kkk)~=0
            m=m+1;
            Y(m,:,kkk)=Y(i,:,kkk);
        end
    end
     if m~=0 && m<kk1
         Y(m+1:kk1,:,kkk)=0;
     end
end
%%%%%%
YY=[];
kk2=1;
kk=ones(1,size(C,1));
for i=1:sp
for j=1:length(Y(:,1,1))
    if Y(j,3,i)>0
        for k1=1:length(s)
            if Y(j,3,i)==s(k1)
                kkk=0;
               for jj=1:length(Y(:,1,1))
                   if t(k1)==Y(jj,3,i)
                YY(kk(i),1,i)=s(k1);
                YY(kk(i),2,i)=t(k1);
                YY(kk(i),3,i)=w(k1);
                kk(i)=kk(i)+1;
                kkk=kkk+1;
                   end
               end
```

```
             kkk=kkk+1;
                    if kkk==1
             YYY(kk2,1)=s(k1);
             YYY(kk2,2)=t(k1);
             YYY(kk2,3)=w(k1);
             kk2=kk2+1;
             end
         end
     end
 end
end
end
Y3=Y;
YY1=YY;
YYY1=YYY;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k1=1:sp
    kk1=length(YY(:,1,1));
    for i=1:kk1
        mm=0;
        for j=1:kk1
            if YY(j,1,k1)~=0
                mm=mm+1;
            end
        end
        if mm>0
            m=0;
            for j=1:kk1
                if (YY(i,1,k1)==YY(j,1,k1))||(YY(i,1,k1)==YY(j,2,k1))
                    m=m+1;
                end
                if (YY(i,2,k1)==YY(j,1,k1))||(YY(i,2,k1)==YY(j,2,k1))
                    m=m+1;
                end
            end
            if m==2
                mm1=0;
                for kkk=1:sp
                    if (YY(i,1,k1)~=0)&&(YY(i,2,k1)~=0)
                    kkk1=Dct(YY(i,1,k1),kkk);
                    kkk2=Dct(YY(i,2,k1),kkk);
                    kk2=length(Y(:,3,1));
                    if mm1==0
                     for j=1:kk2
                       if (Y(j,3,kkk1)~=0)&&(k1~=kkk1)
                         if A(Y(j,3,kkk1),YY(i,1,k1))~=0
                                mm1=mm1+1;
                         end
                        end
                     end
```

```
if (mm1>0) && (mm1<10000)
  rr=0;
   for jj=1:kk2
     if rr==0
       if Y(jj,3,kkk1)==0
          rr=rr+1;
            for ij=1:kk2
            if YY(i,1,k1)==Y(ij,3,k1)
                           ij1=ij;
              end
            end
            Y(jj,:,kkk1)=Y(ij1,:,k1);
            Y(ij1,:,k1)=0;
            for ij=1:kk2
           if zYY(i,2,k1)==Y(ij,3,k1)
                       ij2=ij;
             end
            end

      Y(jj+1,:,kkk1)=Y(ij2,:,k1);
              Y(ij2,:,k1)=0;
              YY(i,:,k1)=0;
              mm1=10000;
            end
       end
     end
     if rr==0
          for ij=1:kk2
             if YY(i,1,k1)==Y(ij,3,k1)
                    ij1=ij;
             end
          end
         Y(jj+1,:,kkk1)=Y(ij1,:,k1);
          Y(ij1,:,k1)=0;
          for ij=1:kk2
             if YY(i,2,k1)==Y(ij,3,k1)
                     ij2=ij;
             end
          end
          Y(jj+2,:,kkk1)=Y(ij2,:,k1);
          Y(ij2,:,k1)=0;
          YY(i,:,k1)=0;
          mm1=10000;
          end
        end
  end
     if mm1==0
    for j=1:kk2
           if (Y(j,3,kkk1)~=0) && (k1~=kkk1)
                if A(Y(j,3,kkk1),YY(i,1,k1))~=0
                   mm1=mm1+1;
```

```
                end
            end
        end
        if (mm1>0) && (mm1<10000)
            rr=0;
            for jj=1:kk2
                if rr==0
                    if Y(jj,3,kkk2)==0
                        rr=rr+1;
                       for ij=1:kk2
                        if YY(i,1,k1)==Y(ij,3,k1)
                                ij1=ij;
                        end
                       end
                      Y(jj,:,kkk2)=Y(ij1,:,k1);
                      Y(ij1,:,k1)=0;
                      for ij=1:kk2
                       if YY(i,2,k1)==Y(ij,3,k1)
                           ij2=ij;
                       end
                      end
                      Y(jj+1,:,kkk2)=Y(ij2,:,k1);
                      Y(ij2,:,k1)=0;
                      YY(i,:,k1)=0;
                      mm1=10000;
                   end
                end
            end
                  if rr==0
                     for ij=1:kk2
                         if YY(i,1,k1)==Y(ij,3,k1)
                             ij1=ij;
                         end
                      end
                     Y(jj+1,:,kkk2)=Y(ij1,:,k1);
                      Y(ij1,:,k1)=0;
                      for ij=1:kk2
                      if YY(i,2,k1)==Y(ij,3,k1)
                            ij2=ij;
                      end
                     end
                     Y(jj+2,:,kkk2)=Y(ij2,:,k1);
                     Y(ij2,:,k1)=0;
                     YY(i,:,k1)=0;
                   mm1=10000;
                end
            end
            end
                  end
```

```
                end
            end
        end
    end
end
%%%%%%%%%
kk1=length(Y(:,3,1));
for kkk=1:sp
    m=0;
    for i=1:kk1
        if Y(i,3,kkk)~=0
            m=m+1;
            Y(m,:,kkk)=Y(i,:,kkk);
        end
    end
     if m~=0 && m<kk1
         Y(m+1:kk1,:,kkk)=0;
     end
end
%%%%%%
YY=0;
kk2=1;
kk=ones(1,size(C,1));
for i=1:sp
for j=1:length(Y(:,1,1))
    if Y(j,3,i)>0
        for k1=1:length(s)
            if Y(j,3,i)==s(k1)
                kkk=0;
               for jj=1:length(Y(:,1,1))
                   if t(k1)==Y(jj,3,i)
                YY(kk(i),1,i)=s(k1);
                YY(kk(i),2,i)=t(k1);
                YY(kk(i),3,i)=w(k1);
                kk(i)=kk(i)+1;
                kkk=kkk+1;
                   end
               end
                kkk=kkk+1;
                       if kkk==1
                YYY(kk2,1)=s(k1);
                YYY(kk2,2)=t(k1);
                YYY(kk2,3)=w(k1);
                kk2=kk2+1;
                end
            end
        end
    end
end
end
```

ANNEXE D. Contact points matrix generation.

```
b=size(YY);
bb=b(3);

for i=1:bb;
    HH=YY(:,:,i);
    d=size(YYY1);
    dd=d(1);
    for j=1:dd;
      e=YYY1(j,:);
      ddd=b(1);
        for a=1:ddd;
          ee=HH(a,:);
          tf=isequal(e,ee);
          if tf>0
            YYY1(j,1)=0;
            YYY1(j,2)=0;
            YYY1(j,3)=0;
          else
              YYY1(j,:)=YYY1(j,:);
          end
        end
    end
end
```

ANNEXE E. Contact points, source and destination vector.

```
A=YYY1(:,1);
A=nonzeros(A);
B=YYY1(:,2);
B=nonzeros(B);
D=[A B];
D=D(:);
D=[ D ,S, DD];
%D=D';
HHH=YY;
HHH(:,1:2,:)=0;
u=0;
uu=0;
nn=0;
for nn=1:bb;
    HH=YY(:,:,nn);
    for n=1:length(D);
          u=D(n);
            for j=1:2;
                for i=1:length(YY);
                    uu=HH(i,j);
                    if u==uu
                       HHH(i,j,nn)=YY(i,j,nn);
                    end
                end
            end
    end
end
```

ANNEXE F. Tie-set algorithm.

```
for ii=1:length(YY(1,1,:))
AA(:,:,ii)=zeros(AR);
for i=1:length(YY(:,1,ii))
    if YY(i,1,ii)~=0
        AA(YY(i,1,ii),YY(i,2,ii),ii)=YY(i,3,ii);
        AA(YY(i,2,ii),YY(i,1,ii),ii)=YY(i,3,ii);
    end
end
end
AAA=zeros(AR);
for i=1:length(YYY1(:,1))
    if YYY1(i,1)~=0
        AAA(YYY1(i,1),YYY1(i,2))=YYY1(i,3);
        AAA(YYY1(i,2),YYY1(i,1))=YYY1(i,3);
    end
end
for j2=1:length(HHH(1,1,:))
    clear h
h=[];
for i=1:length(HHH(:,1,j2))
    if HHH(i,1,j2)~=0
        h=[h HHH(i,1,j2)];
    end
end
for i=1:length(HHH(:,1,j2))
    if HHH(i,2,j2)~=0
        h=[h HHH(i,2,j2)];
    end
end
if length(h)>1
    for i=1:length(h)-1
    for j=i+1:length(h)
        if h(j)==h(i)
            h(j)=0;
        end
    end
    end
end
hh=h;
k2=1;
for i=1:length(hh)
    if h(k2)==0;
        h(k2)=[];
        k2=k2-1;
    end
    k2=k2+1;
end
```

```
if length(h)>1
    h2=0;
    for j3=1:length(h)
        if h(j3)==S||h(j3)==DD
           h2=h2+1;
        end
    end
    if h2==0
    for i=1:length(h)-1
        for j=i+1:length(h)
            AAA(h(i),h(j))=Rela(AA(:,:,j2),h(i),h(j));
            AAA(h(j),h(i))=AAA(h(i),h(j));
        end
    end
    end
    if h2==1
        for i=1:length(h)
            if h(i)==S||h(i)==DD
                h3=h(i);
            end
        end
        for j=1:length(h)
            if h(j)~=h3
            AAA(h3,h(j))=Rela(AA(:,:,j2),h3,h(j));
            AAA(h(j),h3)=AAA(h3,h(j));
            end
        end
    end
        if h2==2
        for i=1:length(h)
            if h(i)==S
                h3=h(i);
            end
            if h(i)==DD
                h4=h(i);
            end
        end
        for j=1:length(h)
            if h(j)~=h3
            AAA(h3,h(j))=Rela(AA(:,:,j2),h3,h(j));
            AAA(h(j),h3)=AAA(h3,h(j));
            end
        end
        for j=1:length(h)
            if h(j)~=h4
            AAA(h4,h(j))=Rela(AA(:,:,j2),h4,h(j));
            AAA(h(j),h4)=AAA(h4,h(j));
            end
        end
    end

end
end
```