Computing and Informatics, Vol. 32, 2013, 1170–1191

APPLYING CLUSTERING TECHNIQUES IN HYBRID NETWORK IN THE PRESENCE OF 2D AND 3D OBSTACLES

Lamiaa Fattouh Ibrahim

Faculty of Computing and Information Technology
King Abdulaziz University
&
Institute of Statistical Studies and Research
Cairo University, Cairo, Egypt
B.P. 42808 Zip Code 21551, Jeddah, Saudi Arabia
e-mail: `lfibrahim@kau.edu.sa`

Hesham Abou El Fetouh Salman

Faculty of Computing and Information Technology
King Abdulaziz University
B.P. 42808 Zip Code 21551, Jeddah, Saudi Arabia
e-mail: `Salman@kau.edu.sa`

Abstract. Clustering spatial data is a well-known problem that has been extensively studied. In the real world, there are many physical obstacles such as rivers, lakes, highways, and mountains, whose presence may substantially affect the clustering result. Although many methods have been proposed in previous works, very few have considered physical obstacles and interlinking bridges. Taking these constraints into account during the clustering process is costly, yet modeling the constraints is paramount for good performance. Owing to saturation in existing telephone networks and the ever increasing demand for wire and wireless services, telecommunication engineers are looking at technologies that can deliver sites and satisfy the demand and level of service constraints in an area with and without obstacles. In this paper, we study the problem of clustering in the presence of obstacles to solve the network planning problem. As such, we modified the Net-Plan algorithm and developed the COD-NETPLAN (Clustering with Obstructed

Distance – Network Planning) algorithm to solve the problem of 2D and 3D obstacles. We studied the problem of determining the location of the multi service access node in an area with many mountains and rivers. We used a reachability matrix to detect 2D obstacles, and line segment intersection together with geographical information system techniques for 3D obstacles. Experimental results and the subsequent analysis indicate that the COD-NETPLAN algorithm is both efficient and effective.

Keywords: DBSCAN algorithm, clustering algorithm, network planning, spatial clustering algorithm and obstacles

1 INTRODUCTION

Spatial data mining reveals patterns in spatial data and knowledge thereof [1]. Spatial data, in many cases, refers to GeoSpace-related data stored in geospatial data repositories like geographical information systems (GISs). The GeoSpace is a continuous 3D space [2], which is important in GISs for representing GeoSpace objects, mountains, and buildings in a 3D space. 3D visualization is helpful for designers because it offers a true to life design scheme. Having a true view of the experience, with the survey analyzed under current conditions in the design stage, plays a key role [3].

In the network planning process, one of the most difficult tasks faced by telecommunication companies is determining the best location and number of Multi Service Access Nodes (MSANs) specially when the city they plan suffer from obstacles. The process of network planning is divided into two sub-problems: determining the location of the switches or MSAN, and determining the layout of the network line paths from the switch to the subscribers while satisfying both cost optimization criteria and design constraints. Owing to the complexity of this process, artificial intelligence (AI) [4, 5] partitioning clustering techniques [6, 7, 8, 9, 10, 11, 12, 13, 14, 15] have been successfully deployed in a number of areas.

Clustering techniques are used by engineers to improve network planning by determining the location for the MSAN. Clustering is one of the most useful tasks in data mining and there are many algorithms that deal with the problem of clustering a large number of objects.

Different algorithms can be classified according to different aspects. These methods can be categorized as partitioning [16, 17, 18], hierarchical [16, 19, 20], density-based [21, 22, 23], grid-based [24, 25, 26], and model-based [27, 28] methods. The clustering task consists of dividing a set of objects into different groups according to certain measures of goodness, which differ according to the application. The application of clustering in spatial databases presents important characteristics. Spatial databases usually contain a very large number of points. Thus, algorithms for clustering in spatial databases do not assume that the entire database can be held in main memory. Therefore, in addition to good quality of clustering, scalability of the algorithm to the size of the database is of equal importance [29]. In spatial databases, objects are characterized by their position in the Euclidean space and, naturally, dissimilarity between two objects is defined by their Euclidean distance [30].

In many real applications the use of the direct Euclidean distance has a weakness [30] in that it ignores the presence of streets, paths, and obstacles that must be taken into consideration during clustering.

In this paper, a clustering-based, COD- NETPLAN algorithm solution is presented which uses the obstacles distance with a density-based clustering technique [31]. COD- NETPLAN algorithm solves the problem of network planning when the district is complained of the presence of obstacles such as mountains (3D), lakes, rivers and highways (2D) etc. The algorithm also solves the problem when the density of subscribers is not homogenously distributed, and there are dense regions in the spatial data space, separated by non-density subscribers regions. All these problems are solved taking into consideration the network constraints, best quality of service and with minimum cost.

In Section 2 the NetPlan clustering algorithm is reviewed, while in Section 3, the COD-NETPLAN algorithm is introduced. A case study is presented in Section 4. Section 5 discusses related work, with our conclusions presented in Section 6.

2 NETPLAN ALGORITHM

DBSCAN is a density-based algorithm [31], where density is defined as the number of points within a specified radius (Eps). DBSCAN defines three types of points. A point is a core point if it has more than a specified number of points (MinPts) within the Eps. These points are within a cluster (i.e., a density-based cluster). A border point has fewer than MinPts within the Eps, but is in the neighborhood of a core point. A noise point is any point that is not a core point or a border point. Figure 1 illustrates the three types of points, while Figure 2 gives the pseudo code for the original DBSCAN clustering algorithm.

Hierarchical algorithms can be categorized as agglomerative or divisive. Agglomerative implies that the clusters are created in a bottom-up fashion, while divisive algorithms work in a top-down fashion.

An agglomerative algorithm starts with each individual item in its own cluster and iteratively merges clusters until all items belong in one cluster [32]. Existing agglomerative algorithms differ with respect to how the clusters are merged at each level. Figure 3 gives the pseudo code for the agglomerative algorithm.

In divisive clustering, all items are initially placed in one cluster and clusters are repeatedly split into two until all items are in their own cluster.

The Network Planning system (NetPlan) [15] includes the following two steps:

Step 1: Apply the modified DBSCAN algorithm.

Step 2: Apply the agglomerative clustering algorithm to the resulting clusters.

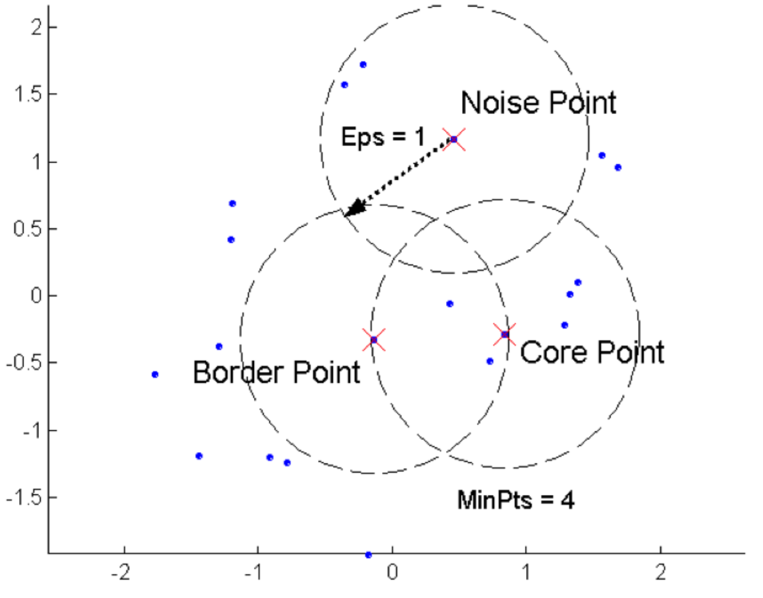

Figure 1. Types of points used in DBSCAN algorithm

2.1 Modified DBSCAN Algorithm

Two parameters must be determined before we can apply the DBSCAN algorithm, namely, MinPts and Eps. In network planning, the cable length must be at most 2.5 km for 0.4 cm diameter cable to achieve an acceptable level of service. So, we set the value of Eps to the value of the shortest path from the core (MSAN) to the furthest point (subscriber), which is 2.5 km. The original DBSCAN algorithm uses Euclidien distance (that is, the direct distance between the MSAN and the node). However, the direct Euclidean distance ignores the presence of streets and paths that must be taken into consideration during clustering. In NetPlan, a clustering based solution is presented that is dependent on the shortest physical routes available. To calculate the shortest path, we selected Dijkstra's algorithm.

When congestion occurs in the MSAN or the number of subscribers is less than 100, we use mobile towers as an auxiliary tool to serve this small number of subscribers. Therefore, the value of MinPts is set to 101.

DBSCAN classifies nodes as follows:

- core points, which form a subset of the candidate MSAN location;
- noise points, which are served by a mobile tower, since all subscribers must be served in an actual plan. Note that a maximum of 100 subscribers can be served by a mobile tower as this is the limit for mobile towers;

```
DBSCAN Algorithm
Eliminate noise points
Perform clustering on the remaining points
Current-cluster-label = 0
For all core points do
  If the core point has no cluster label then
    Current-cluster-label = current-cluster-label + 1
    Label the current core point with cluster label current-cluster-label
  End if
  For all points in the Eps-neighborhood, except the point itself do
    If the point does not have a cluster label then
      Label the point with cluster label current-cluster-label
    End if
  End for
End for
```

Figure 2. DBSCAN clustering algorithm

```
Construct the finest partition (clusters).
Compute the distance matrix.
DO
  Find the two clusters with the closest distance.
  Merge these two clusters into one cluster if the distance condition is satisfied.
  Compute the distance between the new groups and obtain a reduced
    distance matrix.
UNTIL all possible clusters have been agglomerated.
```

Figure 3. Pseudo code for the agglomerative clustering algorithm

- border points, which belong to certain clusters.

NetPlan uses Dijkstra's algorithm to calculate the shortest path from one node to all MSANs (in order to calculate the closest suitable MSAN that can serve this node).

2.2 Agglomerative Clustering Technique

The agglomerative clustering technique is a hierarchical clustering technique. It starts with the points as individual clusters and at each step merges the closest pair of clusters depending on the notion of cluster proximity and the furthest node from the MSAN, which should be at most 2.5 km distant to achieve the required level of service. In NetPlan, after we have distributed the nodes into different clusters (with each cluster served by one MSAN), there may be two cores (MSANs) that are close in distance (less than 1.25 km). In this case, we need to consider decreasing the cost of constructing a new MSAN if one of these cores can carry the loads (subscribers) of both MSANs.

2.3 Implementing the NetPlan Algorithm

The user first inputs the locations of the candidate MSANs, which NetPlan uses as the candidate core points and then selects the one that satisfies the criteria for a core node. Next, NetPlan determines the boundaries of the cluster by calculating the shortest path from the node to each core and allocates the node to the core with the minimum shortest path. The next step is the agglomerative step in which two clusters are merged if the overall load is less than the maximum load of the MSAN, which is 1 536, the maximum shortest path for all nodes to the core is less than or equal to 2.5 km and, as a third condition, the distance between the two cores is less than half of 2.5 km. NetPlan introduces a cost function, which is the sum of the load of the node multiplied by the shortest path between the node and the core for all nodes. This cost function is used to select the best MSAN when two clusters are merged. Figure 4 gives the pseudo code for the NetPlan algorithm used.

```
For (i = 1 to candidate no.)
  For (j = 1 to number of nodes)
    Calculate the shortest path from MSAN(i) to node(j)
    If (shortest path < 2.5 km)
    Then current load(i) = current load(i) + load of node
  End For
    If (Current load(i) >= 101)
    Then add MSAN to cores
End for
For each node in the city select the best switch for it by calculating the shortest
    path between the nodes and each MSAN (path < 2.5)
  Calculate the load of each core
  For each pair of clusters // Agglomerative
    If the shortest path between the cores < 1.25
    Then
      If (sum of two clusters' load < 1 536)
      Then
      First, assume that the first core is the core for the two clusters and apply
        the previous algorithm for each node in the two clusters.
        Second, assume that the second core is the core for the two clusters and apply
        the previous algorithm for each node in the two clusters.
      If (the two cores are both suitable as a core for all nodes in the two clusters)
      Then
      Calculate the Cost Function = ∑ (load×short path distance from node to core)
      Choose the appropriate core depending on minimum Cost Function value.
      Else let the most suitable core be the core for the two clusters.
End For
```

Figure 4. NetPlan algorithm

3 COD-NETPLAN ALGORITHM

The existence of natural obstacles affects the distribution of the MSAN in various regions. The responsible operator is looking to rapidly provide thousands of new subscribers with a high-quality telephone service and provide the correct equipment at the correct place and time, with reasonable cost in order to satisfy expected demand and provide an acceptable level of service.

In a particular city with a certain number of subscribers, we need to determine the number of MSANs required and define their boundaries in order to satisfy good quality of service with minimum cost.

The problem statement can be defined as follows:

Input: A set P of data points $\{p_1, p_2, \ldots, p_n\}$ in a 2-D map, representing the intersection nodes, coordinates of each node, a street map, and distribution of the subscriber loads within the city. We also have the location of obstacles in this city in 2D and 3D. The available cable sizes, the cost per unit for each size, and the maximum distance of wire satisfying the required level of service are also known.

Objective: Partitioning the city into k clusters $\{C_1, C_2, \ldots, C_k\}$ satisfying the clustering constraints, such that the cost function is minimized with a high level of services in the presence of obstacles such as lakes, highways, and mountains.

Output: The k clusters, location of the MSAN, wire branching from each MSAN to the subscribers, boundaries of each cluster, and the nearest base station to the cluster when the cluster is small.

The proposed algorithm comprises three phases, as described in the following subsections.

3.1 Phase I: Preprocessing

This phase has two goals: to preprocess the input data to improve the quality of data and to represent the data so as to improve the efficiency and simplify data mining. The following subsections explain these processes.

3.1.1 The GIS Subsystem

A GIS is a computer system for capturing, storing, querying, analyzing, and displaying geographic data. Figure 5 shows the presentation of data in a GIS. Below we discuss the representation of the data and the improvements carried out on the data.

3.2 Raster Image Enhancement

The maps used for planning are processed as raster images obtained by the user from Google, by scanning an existing map, or through conversion from a digital map to

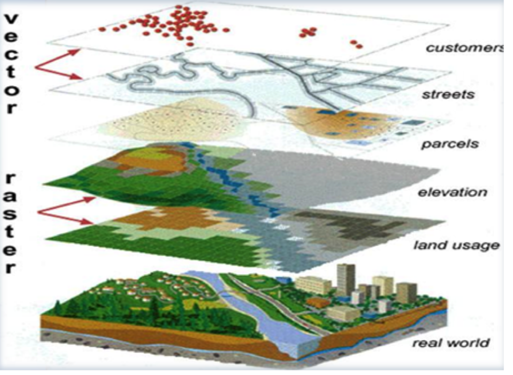

Figure 5. Presentation of data in vector and raster forms

a raster image. Raster images in general, and specifically those that are scanned, require some preprocessing to improve the quality of the image that may have been affected by the scanning operation.

The input to the system is a bitmap file representing the map of the area to be planned. The system deals with the raster file to simplify identification of the terrain and the morphology of the map.

The bitmap used may have suffered a great deal of distortion during the scanning process and noise may be found on such maps. The system addresses this problem by first filtering the map using:

a) Color filtering

This sets each pixel to either white or black depending on the color of the pixel and its intensity.

b) Edge detection

In this process, the system scans the map for any black edges and sharpens them to make the map clearer and more accurate.

c) Closed polygon detection

The objective of this process is to identify the boundaries of an object (lake or mountain) and its height. The process starts with user selection of a polygon by mouse clicking anywhere in the polygon. At this point, the color of the neighboring pixels in all directions is tested to find a black pixel representing the edge of the polygon. Each neighboring pixel in turn searches its neighbors for a black pixel. Whenever a black pixel is found, it is added to a dynamic linked list representing all the pixels on an edge of the closed polygon. The process of searching iterates over the polygon until all pixels have been tested and hence, the edge is completely identified in the linked list.

3.2.1 Data Entry Modes

A data point is the projection of an object on the map. Each object on the map (streets, intersection nodes, and obstacles) is projected onto one or more data points.

a) Representation of intersection nodes

A single data point has the following attributes:

1. position (X, Y)
2. weight, which is an integer value representing the number of subscribers at that position.

B) Representation of streets

We represent streets on the raster maps, with the beginning and end of each street transformed into data nodes, defined by their coordinates. The streets themselves are transformed into links between the data nodes. The subscriber loads are considered to be the weights for each street.

C) Representation of 2D obstacles

The 2D obstacles like lakes, rivers, and highways are represented as polygons with zero height.

3.2.2 Representation of a Mountain as a 3D Obstruction

We used the concept of GIS layers to represent mountains. Each level (height) in the mountain is represented by a polygon. Figure 6 shows a map obtained from Google, which represents an area within Saudi Arabia, which has many mountains. Each mountain has a contour (polygon) at each level and its height is written beside it in the figure and stored in the database. After representing the mountain in the GIS, each of its polygons with the associated height is stored. Figure 7 shows the representation of a single mountain.

Figure 8 describes the function of each button in the system interface, while Figure 9 shows the original scanned map.

3.3 Phase II: Main Planning Phase

COD-NETPLAN uses many algorithms during the planning process as explained in the following subsections.

3.3.1 Dijkstra's Algorithm

In this work, we used Dijkstra's algorithm to calculate the shortest path from one source to several destinations, since in the DBSCAN algorithm, we need to calculate the shortest path from the MSAN to all nodes (the reason being to determine suitable

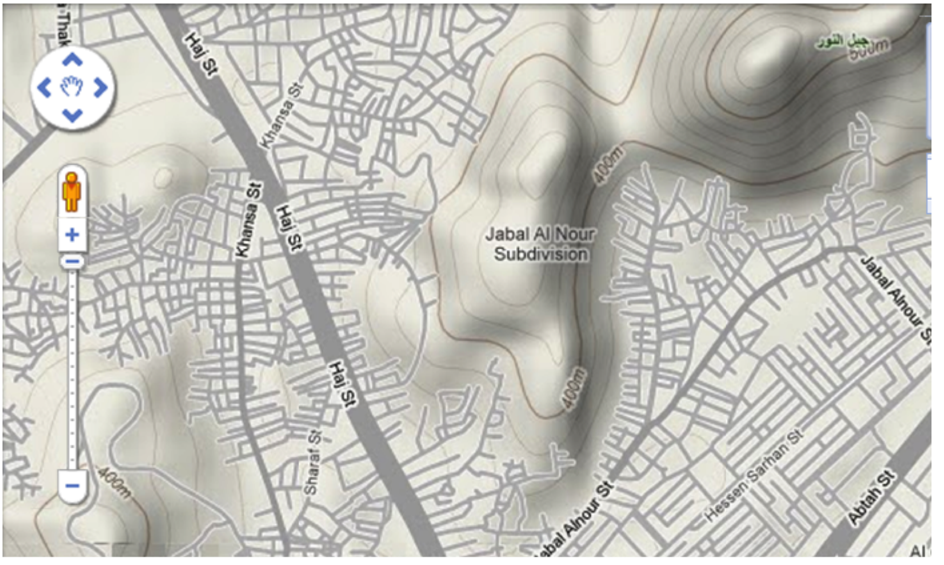

Figure 6. District with many mountainous areas

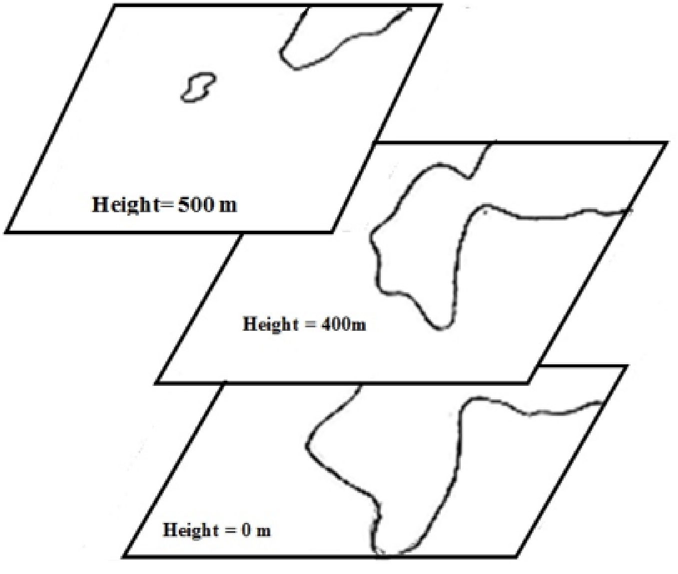

Figure 7. Representation of a mountain in a GIS

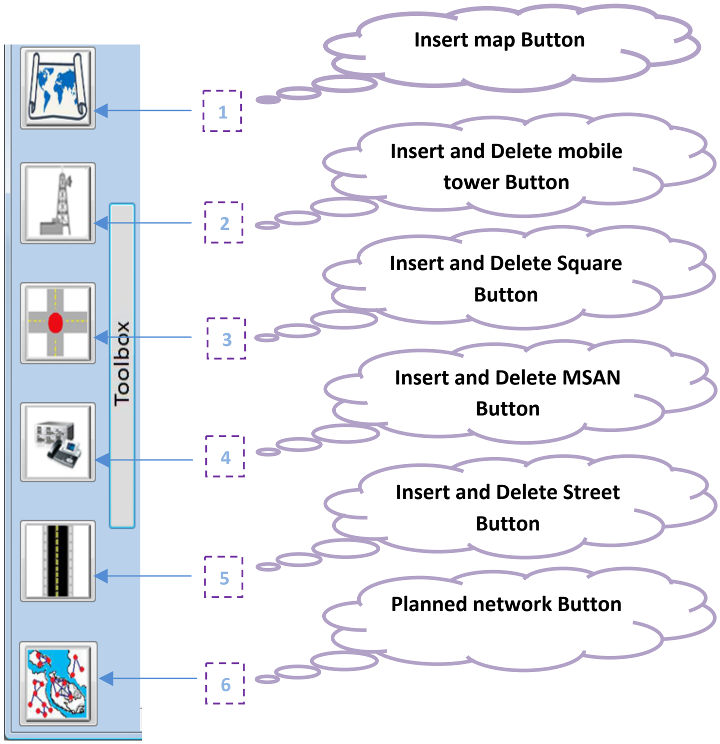

Figure 8. Description of the function of each button in the system interface

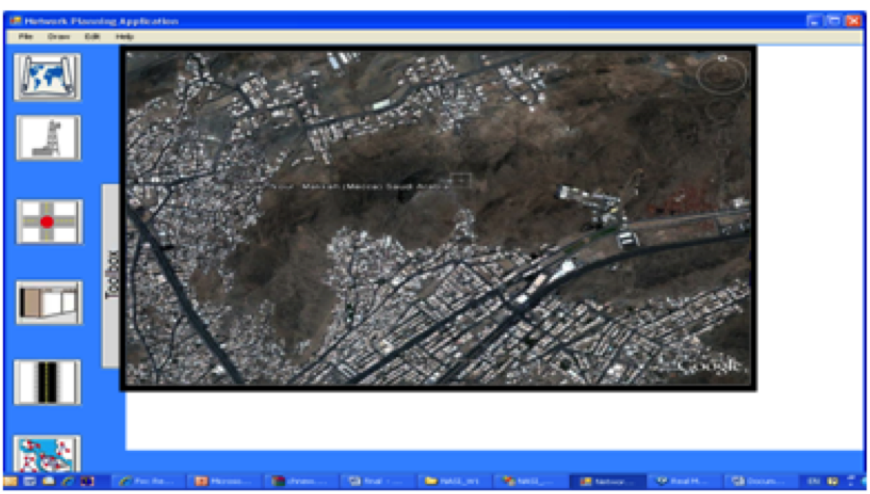

Figure 9. Map before processing

MSANs that serve at least 101 subscribers) and from one node to all MSANs (in order to determine the nearest suitable MSAN that can serve this node). Figure 10 gives the pseudo code for Dijkstra's algorithm.

```
Function Dijkstra(G, w, s)
  For each vertex v in V[G] // Initializations
    D[v]:=infinity
    Previous[v]:=undefined
  D[s]:=0
  S:=empty set
  Q:=set of all vertices
While Q is not an empty set // the algorithm itself
  U:=Extract_Min(Q)
  S:=S union {u}
  For each edge (u, v) outgoing from u
    if d[v] > d[u] + w(u, v) // Relax (u, v)
        d[v]:=d[u] + w(u, v)
      previous[v]:=u
End Function
```

Figure 10. Pseudo code for Dijkstra's algorithm

3.3.2 Calculate the Reachability of One Node from a Core Node

A point K belongs to a core point C if it is reachable from core C. To determine if it is reachable, we use the street matrix $A[C, K]$ (which contains the existing streets between node C and node K). If $A[C, K] > 0$ then we have a direct street, else from node C we investigate all paths in the street matrix. If we succeed in reaching node K then it is reachable from core C, else this point does not belong to the core point; in other words there in an obstruction between C and K.

3.3.3 Calculate Minimum Distance Between Core and Base Station of the Mobile Network

A cluster with load less than or equal to 100 subscribers is served by the nearest mobile tower. The distance between the core of this cluster and the base station may intersect an obstacle as shown in Figure 11. We calculate the minimum distance between the core and base station that does not intersect any obstacle. If the line segment between the core cluster and the base station intersects a mountain the signal will not reach the subscribers. In Figure 11 the cluster is served by base station B because although the line segment between the core and base station A is the shortest, it intersects a mountain.

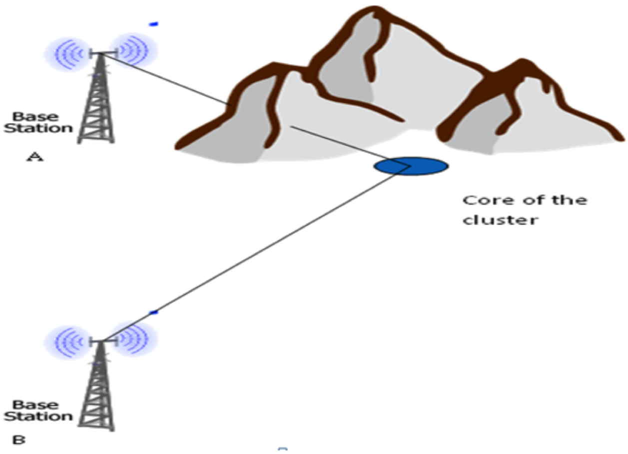

Figure 11. Distance between core and base station

3.3.4 Implementation of COD-NETPLAN Algorithm

Figure 12 gives the pseudo code for the COD-NETPLAN algorithm used, while Figure 13 shows a block diagram of the overall system. The user first inputs the locations of the candidate MSANs. Our system uses these candidate locations as candidate core points and selects the one that satisfies the criteria for a core node. Next the system determines the boundaries of the cluster by calculating the obstacle distance from the node to each core by constructing a binary space partitioning (BSP) tree and visibility graph and allocating the node to the core with the minimum obstacle distance.

3.4 Phase III: Postprocessing

The mined knowledge is output graphically and stored in a database. In the following section we present a case study showing the output knowledge.

4 CASE STUDY

One way of evaluating the effectiveness of a data mining algorithm is to apply it to a real dataset and see what it finds. As a real application, the proposed algorithm was applied to a map representing a district in Saudi Arabia.

The COD-NETPLAN algorithm divided the map into a convenient number of clusters, in which the load of subscribers was distributed.

Figure 9 shows this region, with dark areas depicting mountains. Figure 14 shows the map after applying the COD-NETPLAN algorithm, which divided the map into 15 clusters. MSAN 1 has less than 100 subscribers and base station 3

Algorithm COD-NETPLAN
Input
$D = \{t_1, t_2, t_3, \dots, t_n)$ / * set of elements
Surface of area to be plan
Obstacles location
Output
A partition of the D objects into K cluster
Location of MSAN
Boundaries of each cluster
COD-NETPLAN Algorithm
For ($i = 1$ to candidate No.)
  For ($j = 1$ to number of node)
    If node j is reachable (using reachability Matrix) from node i %% no obstacles between I and C
Then Calculate the obstacle distance from candidate MSAN (i) to node (j) using minimum short path Dijkstra algorithm
  If (shortest path < 2.5 km)
  Then current load (i) = current load (i) + load of node
End For
  If (Current load (i) ≥ 101)
  Then add MSAN to cores
  Else calculate the direct Euclidean distance from core i to nearest base station such than this distance don't intersected with any mountain then add this load to this base station
End for
For each node in the city, select the best switch for it by calculating the shortest path between the nodes and each MSAN (path < 2.5)
Calculate the load of each core
End For
For each two clusters // Agglomerative
If the shortest path between the cores < 1.25
Then
  If (sum of two clusters' load $< 1\,536$)
  Then
First assume that the first core is a core of the two clusters and apply the previous algorithm for each node in the two clusters.
Second assume that the second core is a core of the two clusters and apply the previous algorithm for each node in the two clusters.
If (the two cores are suitable to be a core for all nodes of two clusters)
Then
  Calculate the Cost Function = $\sum$(load $\times$ short path distance from node to core)
  Choose the appropriate core depend on minimum Cost Function value.
Else let the suitable one the core for two clusters.
End For

Figure 12. Implementation of COD-NETPLAN algorithm

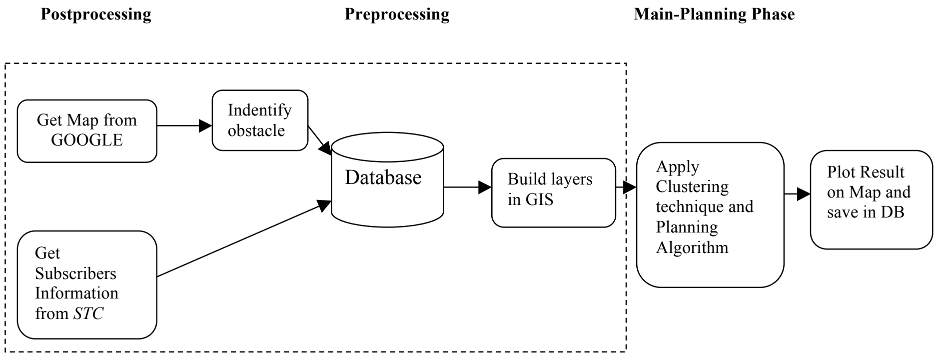

Figure 13. Block diagram of the overall COD-NETPLAN system

is the closest to MSAN 1; however, since a mountain separates them, MSAN 1 is served by base station 2.

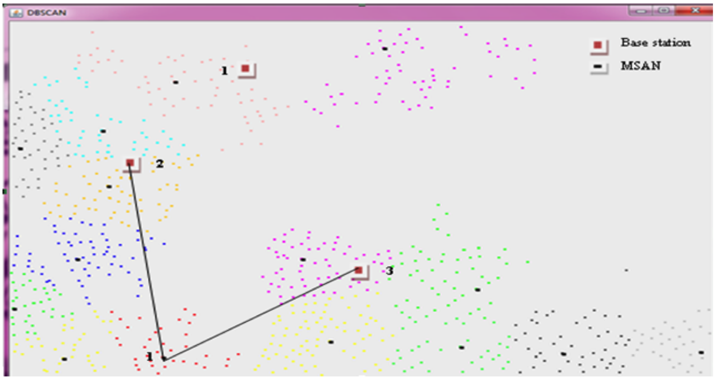

Figure 14. Resulting cluster using the COD-NETPLAN algorithm considering the location of the obstruction

5 RELATED WORK

Table 1 compares related work. In the Gravity Center algorithm [11], the city is divided into four quadrants at the center of gravity, which is the number of clusters. The network constraints are checked for each quadrant; if the constraints are satisfied, the number of clusters is four quadrants (clusters). The switches are located at the center of gravity of each cluster. If the constraints are not satisfied in any of the four quadrants, the same partitioning method is applied to the quadrant that does not satisfy the constraints, yielding seven equal partitions as the number of clusters. This method is iterated until the network constraints are satisfied. The resulting number of clusters may be 4, 7, 10, and so on. This work does not reflect the real nature of the clusters or the number of suitable clusters, as it merely increments the number of clusters by three.

The COD-CLARANS [31] and CSPw-CLARANS [6, 7] algorithms depend mainly on CLARANS, which was designed to deal with a large database by using multiple different samples. These two algorithms are very powerful when planning a large city, but not that accurate when planning a small city owing to the use of sampling.

The ant-colony-based network planning algorithm [13] uses the gravity center to find the location of the switch and applies a modified version of the ant-colony algorithm to find the shortest path. The algorithm is very powerful if the network is complicated with a large number of intersections and streets.

The CWSP-PAM [9] algorithm depends mainly on the PAM clustering algorithm. This algorithm uses the Floyd-Warshall algorithm to find the shortest path.

The CWSP-PAM-ANT [11] algorithm uses a modified PAM clustering technique and the ant-colony algorithm for the network planning problem. This algorithm uses weighted shortest paths satisfying the network constraints where the weights used are the subscriber loads.

COD-DBSCAN [33] uses a density-based clustering algorithm with distances calculated as the obstacle distance in 2D and satisfying the network constraints.

5.1 Effectiveness of the COD-NETPLAN Algorithm

The experimental results and analysis indicate that the COD-NETPLAN algorithm is effective and leads to minimum network construction costs in a low density area with a small number of subscribers owing to the use of a mobile network.

Figure 15 gives a comparison of the total cost calculated by the Gravity Center, COD-CLARANS, CSPw-CLARANS, CWSP-PAM-ANT, and COD-NETPLAN algorithms when the database size is increased gradually.

There is a significant improvement in the results using the COD-NETPLAN algorithm compared with the other algorithms. The figure shows a decrease in cost when using COD-NETPLAN. The Gravity Center algorithm, on the other hand, has the largest number of clusters, which means increasing the number of switches and consequently, the cost. In the COD-NETPLAN algorithm, there are fewer MSANs owing to the use of a mobile network when there is only a small number of subscribers in a low density area and the use of agglomerative clustering algorithms to merge small clusters together. Conversely, the COD-CLARANS, CSPw-CLARANS, and CWSP-PAM-ANT algorithms introduce an MSAN in any constructed cluster even if the number of subscribers is small (less than 100 subscribers), thereby increasing the total cost.

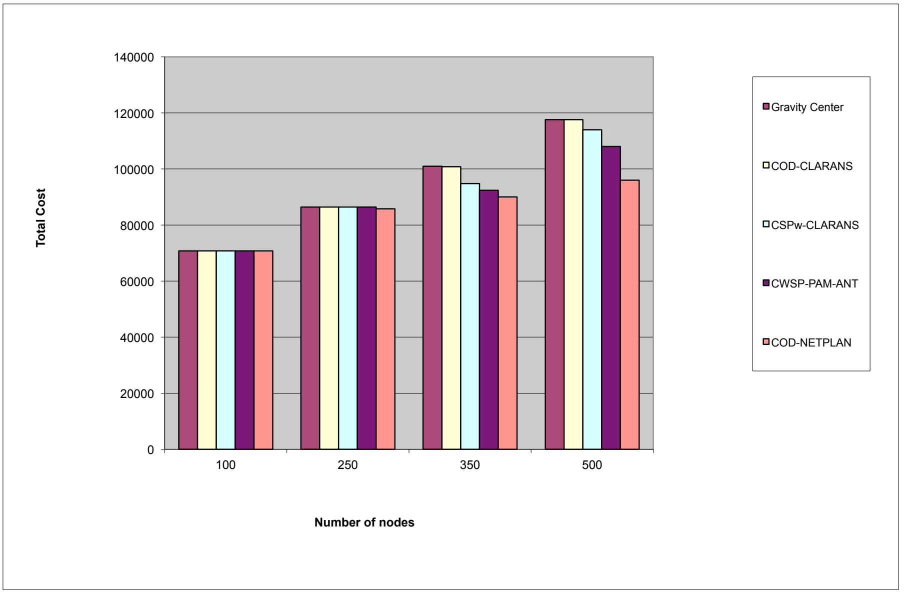

Figure 15. Cost comparison between gravity center, COD-CLARANS, CSPw-CLARANS, CWSP-PAM-ANT, and COD-NETPLAN algorithms

5.2 Efficiency of COD-NETPLAN Algorithm

COD-NETPLAN is more efficient than the other algorithm, as clearly shown by complexity analysis of these algorithms. The complexity of COD-NETPLAN is $O(nc * n)$, where nc is the number of candidate locations and n is the number of nodes. The complexity of COD-CLARANS and CSPw-CLARANS is $O(n * n)$. Finally, the complexity of CWSP-PAM-ANT is $O(k(n - K) * (n - k))$. For large values of n and k these algorithms are too costly, whereas the complexity of COD-NETPLAN remains small, since nc is linear with n.

6 CONCLUSION

Clustering analysis is a key task in various research areas. Clustering aims to identify and extract significant groups in the underlying data. Based on certain clustering criteria, the data are grouped so that the data points in a cluster are more similar to each other than points in different clusters.

Owing to saturation in existing telephone networks and the ever increasing demand for wire services, telecommunication engineers are looking at technologies that can deliver sites and satisfy the demand and level of service constraints in an area with and without obstacles.

Algorithm Name	Algorithm Type	Input Parameters	Results	Constraint	Location of Exchange	Type of Distance
Gravity Center	Gravity Center Algorithm	Data Points	Divide a block in 4,7,10… block	Yes, network constraints	At the gravity center $X_c = \frac{\sum N_i * X_i}{\sum N_i}$ $Y_c = \frac{\sum N_i * Y_i}{\sum N_i}$ N_i= number of subscribers in location i with coordinates X_i, Y_i.	Shortest path distance Floyd-Warshall algorithm
COD-CLARANS	partitioning method CLARANS algorithm	Data points Number of clusters (k) Maximum number of neighbors	Medoids of clusters	Yes, obstacles constraints	At the medoids	Obstructed distance
CSPw-CLARANS	partitioning method CLARANS algorithm	Data points	Medoids of clusters	Yes, network constraints	At medoids with min $C=\sum_{i=1}^{k}\sum_{p\in C_i} L_{ij} d''(c_i, p_j)$ Where c_i is the medoids of C_i, $d''(c_i, p_j)$ is the shortest path from p_j to c_i, L_{ij} is the load cost of this shortest	Shortest path distance Floyd-Warshall algorithm
CWSP-PAM	partitioning method PAM algorithm	Data points	Medoids of clusters	Yes, network constraints	At medoids with min $NTC = \sum_{i=1}^{K}\sum_{n_h\in K_i} L_{hi} dis(n_h, n_i)$ Where n_i is the medoids of cluster K_i, $dis(n_h, n_i)$ is the shortest path from n_j to n_h, L_{hi} is the subscribers load cost of this shortest path	Shortest path distance Floyd-Warshall algorithm
CWSP-PAM-ANT		Data points	Medoids of clusters	Yes, network constraints		Shortest path distance Ant-Colony Algorithm
Ant-Colony-Based Network Planning Algorithm	Gravity Center Algorithm	Data points	Divide a block in 4,7,10… block	Yes, network constraints	At the gravity center	
NetPlan	Density-based DBSCAN algorithm & agglomerative clustering algorithms	- Data points - Candidate switch Location	- Core of the cluster	Yes, network constraints	At core of the cluster	Shortest path distance Dijkstra algorithm
COD-DBSCAN				Yes network and 2D obstruct		BSP-tree algorithm
COD-NETPLAN	Density-based DBSCAN algorithm & Reachability matrix	Data points	Core of the cluster	Yes, network And Geographic Constraints (2D &3D obstructs)		Shortest Path distance Dijkstra algorithm which not intersect the 2D & 3D obstructs

Table 1. Comparison of related work

In this paper, we study the problem of clustering in the presence of obstacles to solve the network planning problem. We presented a spatial clustering solution to the problem of network planning, the COD-NETPLAN algorithm, when the district is complained of the presence of obstacles such as mountains (3D), lakes, rivers and highways (2D) etc. This algorithm uses a clustering technique, which is a density-based method that uses distances calculated from obstacle distances. This algorithm satisfies the network constraints, uses wire and wireless technology to satisfy subscriber demand, and places switches in actual locations within district in the presence of obstacles. We used wireless technology when the number of subscribers is less than 100. To the best of our knowledge, this algorithm is the first algorithm that solves the network planning problem in the presence of 2D and 3D obstructions. The experimental results and analysis indicate that application of the algorithm leads to an effective network construction (minimum cost) while maintaining the best level of service by retaining the cable length not larger than the maximum length which keeps the attenuation in acceptable level. Finally, we presented experimental results showing that the COD-NETPLAN algorithm itself is more efficient than existing clustering methods.

Acknowledgements

This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. 611-014-D1434. The authors, therefore, acknowledge with thanks DSR technical and financial support.

This work was partially supported by the Saudi Telecom Company, STC. We would like to acknowledge the support of the STC in terms of providing expert domain knowledge for network planning and execution.

REFERENCES

[1] Han, J.—Kamber, M.—Pei, J.: Data Mining Concept and Techniques. Elsevier, 2012.

[2] Yanbing, W.: 3D GIS Spatial Modeling for City Surface and Subsurface Integration. Proceedings of the Geoscience and Remote Sensing Symposium, IGARSS 2006.

[3] Xu Y.: The Application of GIS Visualization in Landscape Planning and Design: A Case Study of Nanjing Zhongshan Scenic Area. Proceedings of 19th International Conference on Geoinformatics, 24–26 June 2011 Shanghais.

[4] Fahmyn, H. I.—Douligeris, C.: Application Of Neural Networks And Machine Learning In Network Design. IEEE Journal In Selected Areas In Communications, Vol. 15, 1997, No. 2.

[5] El-Dessouki, A.—Wahdan, A.—Ibrahim, L. F.: The Use of Knowledge-Based System for Urban Telephone Planning. Proceedings of International Telecommunication Union, ITU/ITC/LAS Regional Seminar on Tele-Traffic Engineering for Arab States, Damascus, Nov. 7–12, 1999.

[6] Fattouh, L.—Karam, O.—El Sharkawy, M. A.—Khaled, W.: Clustering For Network Planning. WSEAS Transactions on Computers, Issue 1, Volume 2, ISSN 1109-2750, January 2003.

[7] Khaled, W.—Karam, O.—Ibrahim, L. F.—El Sharkawy, M. A.: CSPw-CLARANS: An Enhanced Clustering Technique for Network Planning Using Minimum Spanning Trees, Proceedings of International Arab Conference on Information Technology ACIT'2003, Alexandria, Egypt, December 20–23, 2003.

[8] Ibrahim, L. F.—Al Harbi, M. H.: Employing Clustering Techniques in Mobile Network Planning. Proceedings of Second IFIP International Conference on New Technologies, Mobility and Security, Tangier, Morocco, 5–7 November 2008.

[9] Ibrahim, L. F.—Al Harbi, M. H.: Employing of Clustering Algorithm CWN-PAM in Mobile Network Planning. Proceedings of The Third International Conference on Systems and Networks Communications, ICSNC 2008, Sliema, Malta, October 26–31, 2008.

[10] Ibrahim, L. F.—Al Harbi, M. H.: Using Clustering Technique M-PAM in Mobile Network Planning. Proceedings of the 12$^{\text{th}}$ WSEAS International Conference on Computers, Heraklion, Crete Island, Greece, July 23–25, 2008.

[11] Ibrahim, L. F.: Using of Clustering and Ant-Colony Algorithms CWSP-PAM-ANT in Network Planning. Proceedings of International Conference on Digital Telecommunications (ICDT 2006), Cap Esterel, French Riviera, France, 26–31 August 2006.

[12] Ibrahim, L. F.: Using of Clustering Algorithm CWSP-PAM for Rural Network Planning. Proceedings of 3$^{\text{rd}}$ International Conference on Information Technology and Applications, ICITA 2005, Sydney, Australia, July 4–7, 2005.

[13] Ibrahim, L. F.—Metwaly, O.—Kabil, A.—Abid, A.: Enhancing the Behavior of the Ant Algorithm To Solve Network Planning Problem. Proceedings of Third ACIS International Conference on Software Engineering, Research, Management and Applications (SERA 2005), Mt. Pleasant, Michigan, USA, 11–13 August 2005.

[14] Ibrahim, L. F. —Salman, H. A.: Using Hyper Clustering Algorithms in Mobile Network Planning. Am. J. Applied Sci., Vol. 8, 2011, pp. 1004–1013.

[15] Ibrahim, L. F.—Minshawi, W. M.—Ekkab, I. Y.—Al-Jurf, N. M.—Babrahim, A. S.—Al-Halees, S. F.: Enhancing the DBSCAN and Agglomerative Clustering Algorithms to Solve Network Planning Problem. In Proc. Data Mining Workshops, 2009. ICDMW '09. IEEE International Conference, pp. 662–667, 6 Dec. 2009.

[16] Kaufman, L.—Rousseeuw, P. J.: Finding Groups in Data: An Introduction to Cluster. John Wiley and Sons, 1990.

[17] Han, L.—Kamber, M.—Tung, A. K. H.: Spatial Clustering Methods in Data Mining: A Survey, Geographic Data Mining and Knowledge Discovery, 2001.

[18] Bradly, P.—Fayyad, U.—Reina, C.: Scaling Clustering Algorithms to Large Databases. In Proc. 1998 Int. Conf. Knowledge Discovery and Data mining, 1998.

[19] Zhang, T.—Ramakrishnan, R.—Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In Proc. 1996 ACM-SIGMOD Int. Conf. Management of data (SIGMOD '96), 1996.

[20] Guha, S.—Rastogi, R.—Shim, K.—Cure: An Efficient Clustering Algorithm for Large Databases. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '98), 1998.

[21] Ester, M.—Kriegel, H. P.—Sander, J.—Xu, X.: A Density Based Algorithm for Discovering Clusters in Large Spatial Databases. In Proc. 1996 Inc. Conf. Knowledge discovery and Data mining (KDD '96).

[22] Ankerst, M.—Breunig, M.—Kriegel, H. P—Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. In Proc. 1999 ACM-SIGMOD Int. Conf. Management of data (SIGMOD '96), 1999.

[23] Hinneburg—Keim, D. A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In Proc. 1998 Int. Conf. Knowledge discovery and Data mining (KDD '98), 1998.

[24] Wang, W.—Yang, J.—Muntz, R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. In Proc. 1997 Int. Conf. Very Large Data Bases (VLDB '97), 1997.

[25] Sheikholeslami, G.—Chatterjee, S.—Zhang, A.: Wave Cluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. In Proc. 1997 Int. Conf. Very Large Data Bases (VLDB '97), 1998.

[26] Agrawal, R.—Gehrke, J.—Gunopulos, D.—Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Application. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '98), 1998.

[27] Shavlik, J. W.—Dietterich, T. G: Reading in Machine Learning. 1990.

[28] Kohonen, T.: Self Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics, 1982.

[29] Nanopoulos, A.—Theodoridis, Y.—Manolopoulos, Y.: C2P: Clustering Based on Closest Pairs. Proceedings of the 27th Proceedings of International Conference on Very Large Data Bases, p. 331–340, September 11–14, 2001.

[30] Yiu, M. L.—Mamoulis, N.: Clustering Objects on a Spatial Network. In Proc. SIGMOD Conference 2004, pp. 443–454.

[31] Tan, T. N.—Steinback, M.—Kumar, V.: Introduction to Data Mining. Addison Wesley, 2006.

[32] Dunham, M. H.: Data Mining Introductory and Advanced Topics. Pearson Education, Inc 2003.

[33] Ibrahim, L. F.: Enhancing Clustering Network Planning Algorithm in the Presence of Obstacles, In Proc. KDIR International Conference on Knowledge Discovery and Information Retrieval, KDIR is part of IC3K, the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Paris, France 26–29 October 2011.

Lamiaa Fattouh Ibrahim received her Ph. D. from Cairo University in 1999. She has over 26 years of experience in the fields of network engineering and artificial intelligence, with particular focus on applying knowledge base and data mining techniques in wire and wireless network planning. She has been a faculty member in the Department of Computer Sciences and Information Systems – Institute of Statistical Studies and Research – Cairo University since 2000. She is currently working in the Department of Information Technology, Faculty of Computing and Information Technology at King Abdul Aziz University in Jeddah. She has published in several journals and at international conferences on networks, data mining, and wire and mobile network planning.

Hesham Abou El Fetouh Salman received a Masters degree in Engineering from Ain Shams University in 1996, and is registered for a Ph. D. in the Faculty of Computing and Information Systems at the same university. He is currently a Lecturer in the Information System Department, Faculty of Computing and Information Technology, King Abdul-Aziz University. Previously, he was the General Manager of the Technology Competency Center, Ministry of Commerce and Industry, in Cairo, Egypt. Throughout his working life, he has participated in many studies and work related to Information Systems. He has over 32 years of experience in the fields of network engineering and programming applications. He has published in the area of networks, data mining, and wire and mobile network planning.