Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 763027, 20 pages
http://dx.doi.org/10.1155/2013/763027

Research Article

Identifying Optimal Spatial Groups for Maximum Coverage in Ubiquitous Sensor Network by Using Clustering Algorithms

Simon Fong,[1] Weng Fai Ip,[1] Elaine Liu,[1] and Kyungeun Cho[2]

[1] *Department of Computer and Information Science, University of Macau, Macau*
[2] *Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 100-715, Republic of Korea*

Correspondence should be addressed to Simon Fong; ccfong@umac.mo

Received 23 March 2013; Accepted 2 June 2013

Academic Editor: Sabah Mohammed

Ubiquitous sensor network has a history of applications varying from monitoring troop movement during battles in WWII to measuring traffic flows on modern highways. In particular, there lies a computational challenge in how these data can be efficiently processed for real-time intelligence. Given the data collected from ubiquitous sensor networks that have different densities distributed over a large geographical area, one can see how separate groups could be formed over them in order to maximize the total coverage by these groups. The applications could be either destructive or constructive in nature; for example, a jet fighter pilot needs to make a real-time critical decision at a split of second to locate several separate targets to hit (assuming limited weapon payloads) in order to cause maximum damage, when it flies over an enemy terrain; a town planner is considering where to station certain resources (sites for schools, hospitals, security patrol route planning, airborne food ration drops for humanitarian aid, etc.) for maximum effect, given a vast area of different densities for benevolent purposes. This paper explores this problem via optimal "spatial groups" clustering. Simulation experiments by using clustering algorithms and linear programming are to be conducted, for evaluating their effectiveness comparatively.

1. Introduction

Ubiquitous sensor network is a kind of wireless sensor technology [1] that has sensors distributed far and wide, usually covering a large geographical area like forest, battle field, or road networks of an urban city. Few successful case scenarios have been in place in the literature, such as monitoring vegetable freshness by using oxygen and carbon dioxide sensors in farms [2], chemical leak detection in hazardous sites [3], general-purpose sensor networks that monitor fire [4], and operation underwater [5]. What these applications have in common is the need of a postprocessing step that crunch over the data, possibly in real time, and to make a quick and accurate prediction out of the analysis.

In this paper, we consider a special case of postprocessing of such ubiquitous sensor network data. Given a vast distribution of sensors each of which collects some information about the local proximity, some groups or clusters are to be formed over those. The groups should be formed in such a way that the total overall "value" of all the values from all the groups must be maximized. The value(s) which should be part of the attribute information being collected by the sensors may be something that is of the user's concern. The values usually represent the density of a proximity where a sensor stands, for example, concentrate of some chemical gas, traffic volume, importance of military target, or even head counts of castles or humans.

Intuitively one would prefer the groups to be centered on the most valuable values over the area; the groups should not overlap much of each other, so the overlapped effect may even get cancelled out or wasted in vain. Here some reasonable assumptions would have to be held valid: each group would have a limited diameter of effect; each group is in the shape of a concentric circle; the areas where the circles (groups) cover sum up to a total coverage a.k.a. maximum net effect; and we can form only a limited number of such circles. This would be an interesting mathematical problem but it has a significant impact on ubiquitous sensor network applications. It not only

determines on how we should distribute the sensors, but also after the deployment how these logical groups are being formed possibly for further applications.

For experiments, we attempted to apply several clustering algorithms; the choices of these algorithms are classical and popular in data mining research community. The effectiveness of different clustering algorithms is measured for comparison. However, none of the clustering algorithms can achieve the best results. At the end, we develop a simple and novel method based on linear programming for optimization, which we called LP. LP is shown to be able to achieve optimal grouping over different configurations and cases of experiments.

The contribution of the paper is an in-depth investigation into the grouping problem that arises right after the deployment of a ubiquitous sensor network. We propose a novel solution to achieve optimal groups by using linear programming, though several clustering algorithms have been put into test.

The remaining of the paper is structured as follows. Section 2 introduces the background techniques of spatial clustering. Section 3 surveys on spatial data representation, how the spatial data are encoded for postprocessing. Section 4 describes our methodology for obtaining optimal groups over spatial data. Section 5 reports about the experiments. Section 6 analyses and compares the experimental results. Section 7 concludes the paper.

2. Overview of Spatial Data Clustering Techniques

Clustering is the organization of a dataset into homogeneous and/or well-separated groups with respect to a distance or, equivalently, a similarity measure [6]. Spatial data clustering has numerous applications in pattern recognition [7, 8], spatial data analysis [9–11], market research, and so forth, [12, 13], which gather data to find all non concentrate models and special things among geographical dataset. And spatial data clustering is an important instrument of spatial data mining, which has become a powerful tool for efficient and complex analysis of huge spatial databases [12] with geometric features, and is liked by most people. Conventional methods of clustering algorithm are classified into four types: partition method, hierarchical method, density-based method, and grid-based method.

However, with the extension of research objects and scope, it has been discovered to have shortcomings. Many existing spatial clustering algorithms cannot cluster with irregular obstacles reliably. A grid-density-based hierarchical clustering (HC) algorithm is proposed to tackle this problem. The advantage of grid-based clustering algorithm is reducing the quality of calculation. An alternative approach [14] is proposed that can effectively form clustering in the presence of obstacles. The shapes of the clusters can be arbitrarily defined. Moreover, the hierarchical strategy is used to reduce the complexity in presence of obstacles and constraints and to improve the operation efficiency [13]. And the result is that it can deal with spatial clustering while it faces obstacles and constraints and get better performance. When some data points do not work in any cluster for density clustering, this situation was managed by using grid-based HC instead in this study. And each clustering algorithm has its individual advantages and disadvantages.

The partition approach separates N objects into m groups, meanwhile m satisfies the following constraints: firstly, each group contains one object at least; secondly, each object must belong to one group. In order to achieve a global optimum in grouping, it is necessary to list all possible clusters, and most applications will adopt KM, K-medoid, or fuzzy analysis. However, this partitioning method consists of some problems when applied to spatial mining, towards clustered objects, especially the objects that are obstructed by some environment conditions. Such as river, it is hard to recognize the comparability.

This method can be efficiently carried out by clustering no matter how large the number of objects. Cluster analysis algorithm in general cannot deal with large datasets. It is recommended that the maximum number of objects to deal with in this method should be no more than 1000. It is a stochastic hunting way based on partitioning in the clustering method due to its low efficiency and the capability of this method is much affected by the random selection of the stochastic initial value [14].

HC is another popular clustering method, which is more flexible than partitioning-based clustering but it has a higher time complexity. HC algorithms create a hierarchical decomposition (a.k.a. dendrogram) of dataset based on some criterion [15, 16]. According to the rule of generation in hierarchical decomposition, there are two different types of HC methods: agglomerative and divisive. For agglomerative algorithm, it starts with producing leaves and it combines clusters in a bottom-up way. For divisive algorithm, the clustering starts at the root and it recursively separates the clusters in a top-down way. The process continues until a stopping criterion—usually it stops when the required k clusters are accomplished. However, this hierarchical method consists of some problems: vagueness of termination criteria; once a step is complete, it cannot be revoked. As long as the neighborhood density (the number of objects or data points) does not grow over a certain threshold, clustering continues [17]. In other words, for a given point in each cluster, in the neighborhood of a given radius it must contain at least a minimum number of points. As a result, the noisy data can be filtered, and better clusters with arbitrary shape can be found. DBScan and its expansion algorithm, which is called OPTICS, are two of these classical density-based methods. They perform clustering based on the type of density-based connectivity.

Grid-based method quantifies object space to a restricted number of cells, forming a grid structure. All clustering operations are in the grid structure (i.e., quantitative space). The main benefit of this method is its high speed; the run time is usually not restricted by the data size, which only depends on the number of cells in each dimension. The algorithm STING (statistical information grid-based method) [18] works with numerical attributes (spatial data) and is designed to facilitate "region-oriented" queries.

Nevertheless, the spatial groups obtained by classic algorithms have certain limitations; that is overlaps cannot be controlled and the maximum coverage by the resultant groups is not guaranteed. Overlaps lead to resource waste and potentially resource mismatch. Besides spatial clustering, this situation occurs in other fields of applications such as the information retrieval (several thematic for a single document); biological data (several metabolic functions for one gene), and martial purpose (discover object denseness regions independently). However, there has been no study reported in the literature that the authors are aware of that applies LP method to discover spatial groups that are free of the limitations inherited from clustering algorithms. Thus, this research provides an alternative method to achieve spatial groups for maximum coverage in real environment. Maximum coverage in this context here is defined as the greatest possible area of effect covered by the spatial groups with none or minimum overlaps among the groups.

3. Spatial Data Representation

Two main categories of spatial data representation exist: spatial data and attribute data. Spatial data means referenced data in the earth, such as maps, photographs, and satellite imageries. Though these representation techniques originated from GIS, the underlying coding formats are common compared to those for wireless sensor networks as long as they are distributed over a wide spatial area in nature. Generally, spatial data represents geographic features in complete and relative locations. Attribute data represents the spatial features in characteristics, which can be in quantity and/or in quality in real world. Attribute data is often referred to as tabular data. In our experiments, we test both types of data models versus different clustering algorithms for a thorough investigation.

3.1. Spatial Data Model. In early days, spatial data is stored and represented in a map format. There are three fundamental types of spatial data models for recording the geographic data digitally. They are vector, raster, and image.

Figure 1 as shown in the following illustrates the encoding techniques of two important spatial data [19], which are raster and vector, over a sample aerial image of Adriatic Sea and coast in Italy. The image type of encoding is very similar to raster data in terms of usability of techniques. But it has a limit in internal formats when it comes to modeling and analysis of the data. Images represent photographs or pictures in the landscape in a coarse matrix of pixel values.

3.2. Vector Data Model. The three kinds of a forementioned spatial data models are used in storing the geographic location with spatial features in dataset. The vector data model uses x, y coordinates to define the locations of features; thereafter they mark points, lines, areas, or polygons. Therefore, vector data tend to define centers, edges, and outlines of features. It characterizes the features by linear segments using sequential points or vertices. A vertex consists of a pair of x and y coordinates. The beginning or ending of a node is defined in each vertex with arc segment. A single coordinate pair of vertexes defines a feature point. A group of coordinate pairs define polygonal features. In vector representation, as well as the connectivity between features, the storage of the vertices for each feature is important, as well as the sharing of common vertices where features connect.

By using the same size polygonal, we divide a complete map into small units based on the character of our database, which is represented to be (x, y, v), where x and y consist of an coordinate pair that represents the referenced spatial position and v represents something of interest or just called "feature" which could be a military target, a critical resource, or just an inhabitant clan, for example. The greater the v, the more valuable the feature is. In spatial grouping for maximum coverage, we opt to include these features that amount to a highest total value. A sample of vector format that represents a spatial location in reference to 2D is shown in Figure 2 [19].

3.3. Raster Data Model. Raster data models make use of a grid of squares to define where features are located. These squares which are also called pixels or cells typically are of uniform size.

From our dataset, we separate the whole image by imposing a grid on it hence producing many individual features, with one feature corresponding to each cell. We consider using raster data model to represent the dataset, and we store the features by the following two different encoding formats.

(1) Raster data are stored as an ordered list of cell values in pairs of (i, v), where i is a sequential number of the cell indices and v is the value of the ith feature, for example, (1, 80), (2, 80), (3, 74), (4, 62), (5, 45), and so on, as shown in Figure 3.

(2) Raster data are stored as points (x, y, v), with x and y as position coordinates locating to the corresponding spatial feature with value v, for example, (1, 1, 513), (1, 2, 514), (1, 3, 517), (2, 1, 512), (2, 2, 515), and so on, as shown in Figure 4. In this case, the value v refers to the center point of the grid cell. This encoding will be useful for representing measured values at the center point of the cell, for example, raster of elevation.

(3) During the experiment, the grid size is transformed for efficient operation. So, we put i^2 cells together as one unit representing one new grid cell, as shown in Figure 5.

In particular, the quad tree data structure for storing the data is found to be useful as an alternative encoding method to raster data model. Raster embraces digital aerial photographs, imagery from satellites, digital pictures, or even scanned maps. Details on how different sorts of objects like point, line, polygon, and terrain are represented by the data models can be found in [19–21].

4. Proposed Methodology

The aim of the methodology is to determine a certain number of clusters and their corresponding locations from some

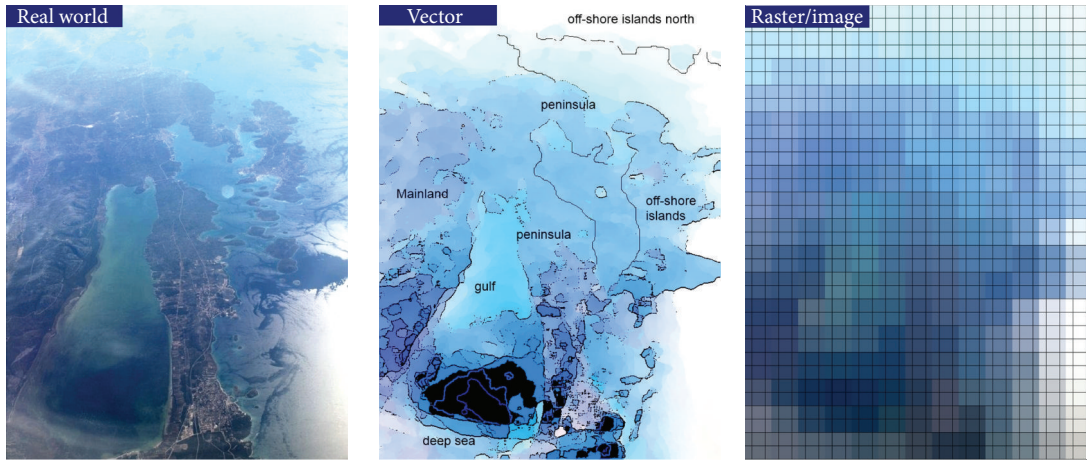


Figure 1: Representation of how a real-world spatial area is represented by vector and raster encoding formats.

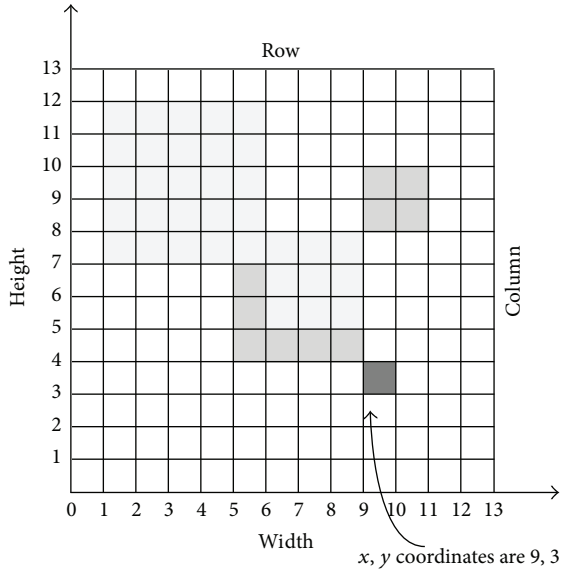


Figure 2: Vector format.

80	74	62	45	45	34	39	56
80	74	74	62	45	34	39	56
74	74	62	62	45	34	39	39
62	62	45	45	34	34	34	39
45	45	45	34	34	30	34	39

Figure 3: Raster format in ordered list.

513	512	516	517	520
514	515	519	521	523
517	517	523	528	527
511	512	510	520	523
510	511	512	516	518

Figure 4: Raster data with center point.

80	74	62	45	45	34	39	56
80	74	74	62	45	34	39	56
74	74	62	62	45	34	39	39
62	62	45	45	34	34	34	39
45	45	45	34	34	30	34	39

Figure 5: Raster format with 2^2 and 3^2 grids.

collected spatial data. In this process, different methods are tested for choosing the one which covers the most area as well as the highest feature values, from the suggested clusters. The flow of this process, including preprocessing of sensor data, data transformation, clustering, and finding cluster center-points, is shown in Figure 6.

In case of a satellite image or image captured by fighter-jet or other surveillance camera, image processing is needed to

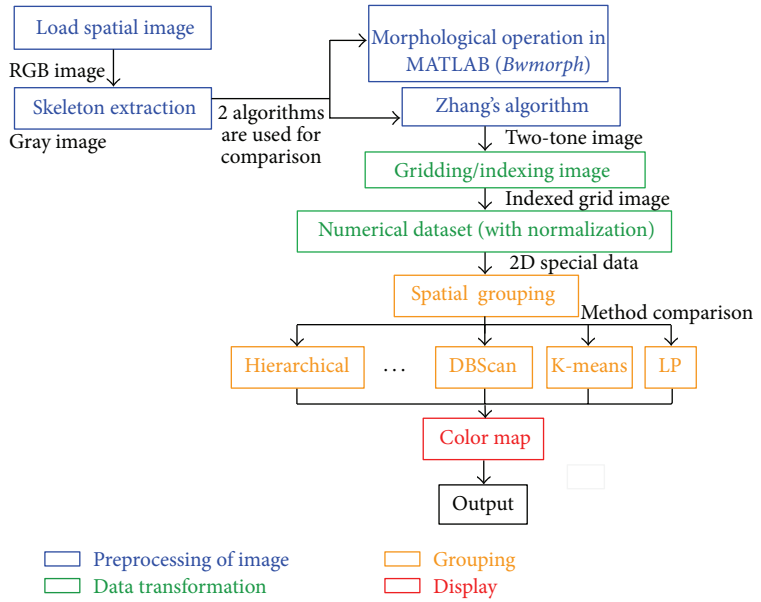


Figure 6: Workflow of proposed methodology.

extract the density information from pictures. But in our case of sensor network, we can safely assume that the data fed from a net of sensors would have the sensor ID attached. The sensor IDs are known, so are their positions. From the locations of the sensors and their sensor ID, we could possibly relate the data that was collected to their corresponding locations, in the x-y format of coordinates (assume the terrain is of 2D). In order to reduce the huge amount of calculation and storage space, a grid was used to divide the whole map into smaller pieces. The grid indexing operation is repeated for a range of different coarse layers, thereby providing different resolutions of data partitions. Similar technique is reported in [22], which is computed by Euclidian distance. Obviously, the method of grid indexing helps separate data into cells based on their geographic locations.

To obtain a better result of spatial groups for maximum coverage and its corresponding cluster center point with certain constrains, the research adopts several popular clustering methods and linear programming method by using software programs such as XLMiner (http://www.solver.com/xlminer-data-mining/), MATLAB (http://www.mathworks.com/products/matlab/), and Weka (http://www.cs.waikato.ac.nz/ml/weka/).

The core purpose of cluster analysis is to comprehend and to distinguish the extent of similarity or dissimilarity amount of the independent clustered objects. There are five major methods of clustering—KM, EM, XM, HC, and DBScan.

K-means (KM) by MacQueen, 1967, is one of the simplest algorithms that solve the well-known clustering problem [23]. It is an easy and simple method to divide a dataset into a certain number of clusters initially, assuming that the number of clusters is k fixed a priori for each cluster, which is the main idea. The random choice of the initial location of centroids leads to various results. A better choice is to place them as much far away from each other as possible.

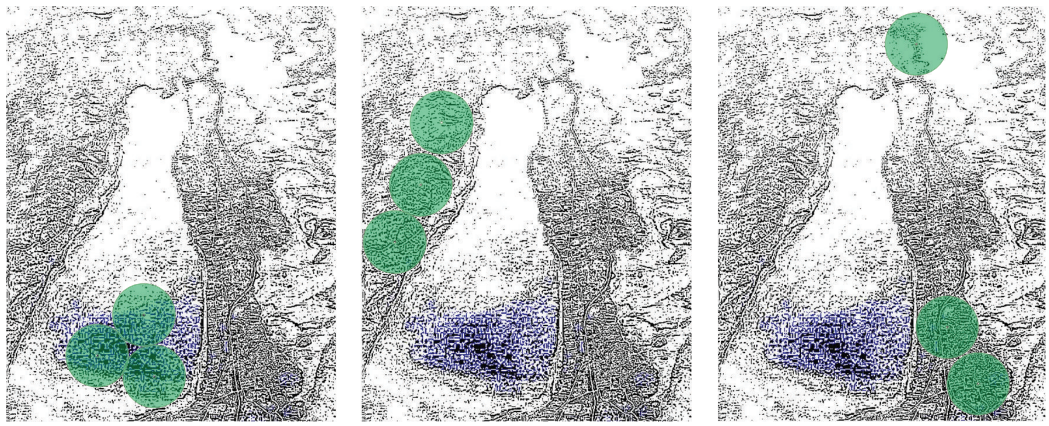
The KM algorithm aims at minimizing an objective function. In this case, a squared error function is as follows:

$$j = \sum_{\forall i} \sum_{\forall j} \left\| x_i(j) - c_j \right\|^2, \tag{1}$$

where j ranges from 1 to k, i range, from 1 to n, and $\|x_i(j) - c_j\|^2$ is a chosen distance measure between a data point $x_i(j)$ and the cluster center c_j, which is an indicator of the distance of the n data points from their respective cluster centers. The sum of distances or sum of squared Euclidean distances from the mean of each cluster is a quite normal or usual measure for causing scattering in all directions in the cluster in order to test the suitability of the KM algorithm. Clusters are often computed using a fast, heuristic method, which generally produces good (but not necessarily optimal) solutions.

X-Means [24] is an optimal method of KM, which improves structure part in the algorithm. Division of the centers is attempted in its region. It makes decision between the root and children of each center to doing the comparison between the two structures. Another improved variant of KM, called EM which execrates maximization, makes an assignment on a probability distribution to each further point which represents the probability. How many clusters to be set up are to be decided by EM using cross-validation.

Density-based algorithms regard clusters as dense areas of objects that are separated by less dense areas [25]. Because they have no limit to look for clusters with spherical shape, they can produce clusters with arbitrary shapes. DBScan is a typical implementation of density-based algorithms, called density-based spatial clustering of applications with noise



(a) (b) (c)

Figure 7: Illustration of possible ways of assigning clusters for maximum (a) fish population, (b) altitude of terrain, and (c) human inhabitant population.

[25]. The notions of density reachability and density connectivity are used as performance indicators for the quality of clustering [26]. A cluster is composed of the group of objects in a dataset that are density connected to a particular center. Any object that falls beyond a cluster is considered as noise.

Ward proposed a clustering method called hierarchical clustering (HC) in 1963 [27]. It tries to find how to form something to divide $P_n, P_{n-1}, \ldots, P_1$ in a way that reduces the relationship with each group. In each step analysis step, it considered every possible cluster pair in group and combined the two clusters with a very close joining of results in "information loss," which is given definition by Ward around ESS (an error sum-of-squares criterion). The idea that supports Ward's proposal can be described most simply by thinking of a little single data. Take ten objects with scores as an example: (2, 7, 6, 6, 7, 2, 2, 0, 2, 0). The loss of information would be achieved by calculating ESS with a mean of 3.4, which takes into account the ten scores as a unit as follows: ESS One group $= (2-3.4)^2 + (7-3.4)^2 + \cdots + (0-3.4)^2 = 47.28$. However, those 10 objects can also be separated into four groups according to their scores: $\{0, 0, 0\}$, $\{2, 2, 2, 2\}$, $\{6, 6\}$, and $\{7, 7\}$. Finally, for evaluation of the ESS as a sum of squares, we can obtain four independent error sums of each square. Overall, the result that divides the 10 objects into 4 clusters has no loss of information as follows:

$$\begin{aligned}\text{ESS One group} = {} & \text{ESS group1} + \text{ESS group2} \\ & + \text{ESS group3} + \text{ESS group4} = 0.\end{aligned} \tag{2}$$

The last method we adopted here is linear programming (LP), which contains instituting and producing an answer to optimization problems with linear objective functions and linear constraints. This powerful tool can be used in many fields especially where many options are possible in the answers. In spatial grouping over a large grid, many possible combinations of positioning the clusters exist. The problem here is to find a certain number of clusters of equal size over the area, meanwhile the chosen centers of the clusters must allow sufficient distance apart from each other so as to avoid overlapping. As an example, shown in Figure 7, three clusters would have to be assigned over a spatial area in a way that they would have to cover certain resources. The assignment of the clusters, however, would have to yield a maximum total value summed from covered resources. In the example, the left diagram shows allocating three clusters over the deep water assuming that the resources are fish hence the maximum harvest. The second example in the middle of Figure 7 is clustering the high altitude over the area. The last example is trying to cover the maximum human inhabitants which are concentrated at the coves. Given many possible ways of setting up these clusters, LP is used to formulate this allocation problem with an objective of maximizing the values of the covered resources.

Assuming that the resources could be dynamic, for example, animal herds or moving targets whose positions may swarm and change over time, the optimization is a typical maximal flow problem (or max flow problem). The optimization is a type of network flow problem in which the goal is to determine the maximum amount of flow that can occur over arc whish is limited by some capacity restriction. This type of network might be used to model the flow of oil in pipeline (in which the amount of oil that can flow through a pipe in a unit of time is limited by the diameter of the pipe). Traffic engineers also use this type of network to determine the maximum number of cars that can travel through a collection of streets with different capacities imposed by the number of lanes in the streets and speed limits [28].

For our spatial clustering, we consider each cell of the grid as a node; each node is defined as a tuple m that contains the coordinates and the value of the resource that is held in the node such that $m(x_i, y_i, z_i)$ represents an ith node, in which x_i, y_i represent the position and z_i represents the value of resource in the node, respectively. For the clusters, each node

(1) Load the grid-based spatial information into array $A(x, y, z)$ % A is a three dimensional array
(2) Repeat (through all coordinates of x)
(3) Repeat (through all coordinates of y)
(4) If (boundary constraints and overlapping constraints are satisfied) Then
(5) $S(x_i, y_i, z_i) = A(x_i, y_i, z_i)$
(6) End-if
(7) End-loop
(8) End-loop
(9) If *size of* $(S) \geq K$
(10) Find top K clusters where max $\sum z_i \bigoplus C_k$, copy $S(x_i, y_i, z_i)$ to new array $C(x_i, y_i, z_i), \forall i \in C_k$
(11) Else-if
(12) $C(x_i, y_i, z_i) = S(x_i, y_i, z_i) \forall i$
(13) End-if

Pseudocode 1: Pseudocode of the proposed LP model for spatial clustering.

can potentially be a center of a cluster, and the cluster has a fixed radius of length r. The LP model for our problem is mathematically shown as follows:

$$\begin{aligned}\text{Total value} &= \bigcup_{\text{selected clusters } \langle C_k | k=1 \cdots K \rangle} \sum_{m_i \in C_k} m_i (*, *, z_i) \\ &= \arg\max \sum_{\substack{0 \leq x_i \leq X \\ 0 \leq y_i \leq Y}}^{X,Y} \sum_{k=1}^{K} z_l \ni m_l (x_i, y_j, z) \oplus c_k. \end{aligned} \tag{3}$$

Subject to the boundary constraints of $2r \leq |x_i - x_j|$, and $2r \leq |x_i - x_j|$ for all i and j, but $i \neq j$, where X is the maximum width and Y is the maximum length of the 2D spatial area, respectively, $k \in K$ is the maximum number of clusters, and c_k is the kth cluster under consideration in the optimization.

In order to implement the computation as depicted in (3) for each node, we sum each group resources in a shape of diamond (which geometrically approximates a circle). By iterating through every combination of K nodes in the grid of size X by Y, each current node in the combinations is being tested by considering it as the center of a cluster that has a radius of r, hence storing the resource values of the nodes from the potential clusters into a temporary array buffer $A(*, *, z_i)$. The results from those potential clusters which do satisfy the boundary and nonoverlapping constraints are then copied to a candidate buffer. Out of the clusters whose resource values are stored in the candidate buffer S, the combination of K clusters that has the great total resource value is selected and their values are placed in the final buffer C. The corresponding pseudocode is shown in Pseudocode 1.

5. Experimental Results and Analysis

In this section, the performance of the proposed methodology is shown by presenting both numerical and visualized results for all performance aspects over various algorithms. A case study of road traffic is used in the experiment. The spatial area is a metropolitan traffic map with roads and streets spanning all over the place. The resource value in this case is the concentration or density of vehicle traffic flows. Sensors are assumed to have been deployed in every appropriate point of the roads; thereby a typical traffic volume is each of these points is known. The optimization of spatial clustering in this case can be thought of as optimal resource allocation; for example, cost-effective police patrols, gas stations, or environment-pollution controls are needed among those dense traffic spots.

Table 1: Comparison between *Bwmorph* function and thinning algorithm.

	Bwmorph function		Thinning algorithm	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2
Degree of thinning	Incomplete		Complete	
Elapsed time (secs)	20	38	100	198
Complexity	$O(n)$		$O(n^2)$	

5.1. Data Preprocessing. Two different factual datasets are used for experiments. The first dataset is published by Maricopa Association of Governments in 2008, which is a traffic volume map. Traffic volumes were derived from the national traffic recording devices. Seasonal variation is factored into the volumes. The second dataset is an annual average daily traffic of Baltimore County Traffic Volume Map in 2011 in USA, prepared by the Maryland Department of Transportation, and published by March 19, 2012. The traffic count estimates are derived by taking 48-hour machine count data and applying factors from permanent count stations. The traffic counts represent the resource values in a general sense.

After using skeleton extraction, a two-tone image was obtained from the original map. Readers are referred to the respective websites where they can see the traffic volume data that are associated with our two datasets: (a) Representative traffic volume map of dataset 1—Traffic Volume Map of Phoenix, AZ, USA (http://phoenix.gov/streets/traffic/volume-map/); (b) Representative traffic volume map of dataset 2—Traffic Volume Map of Baltimore, MD, USA (http://www.marylandroads.com/Traffic_Volume_Maps/Traffic_Volume_Maps.pdf/). And the corresponding result skeleton extraction

Table 2: Important statistics from the clustering and LP experiments.

Method	Cluster number	Number of cells covered	Minimum	Maximum	Overlap
KM	Cluster 1	428	0	349932.7	0
	Cluster 2	468	0	546896	0
	Cluster 3	448	0	205030.07	0
	Cluster 4	614	0	68946.67	0
	Cluster 5	618	0	9009.08	0
XM	Cluster 1	615	0	59126.5	0
	Cluster 2	457	0	546896	0
	Cluster 3	609	0	9009.08	0
	Cluster 4	465	0	349932.7	0
	Cluster 5	430	0	205030.07	0
EM	Cluster 1	1223	0	2292	618172.29
	Cluster 2	7	14104.8	24370.5	313018
	Cluster 3	81	0	30337.33	1311465.77
	Cluster 4	64	26752	546896	3308812.49
	Cluster 5	1201	0	130002.6	2179504.71
DB	Cluster 1	13	23614	33146	3272229.11
	Cluster 2	11	16868.25	21001	3639658.18
	Cluster 3	13	17888.8	29452.83	1961183.93
	Cluster 4	11	8477.33	21100.8	589408.77
	Cluster 5	2528	0	546896	2055417.6
HC	Cluster 1	291	0	349932.7	0
	Cluster 2	191	0	205030.07	967622.83
	Cluster 3	294	0	1590971	0
	Cluster 4	224	0	189812	1267355.5
	Cluster 5	243	0	546896	0
LP	Cluster 1	221	0	349932.7	0
	Cluster 2	221	0	205030.07	0
	Cluster 3	221	0	1590971	0
	Cluster 4	221	0	189812	0
	Cluster 5	221	0	546896	0

Table 3: Comparison for running time of the first dataset.

Formats	KM	HC	DBscan	XM	EM	LP
Vector database	3.27	12.52	23.24	2.78	9.30	1.83
Raster database	**3.42**	**15.36**	**28.20**	**2.84**	**9.84**	**2.01**
RasterP (16 grids)	1.98	1.34	5.08	0.46	0.57	0.78
RasterP (25 grids)	0.09	0.14	1.15	0.21	0.12	0.53

in dataset 1 is shown in Figure 8, where (a) adopted a kind of morphological operation method, and (b) adopted thinning algorithm, respectively. Likewise, the corresponding result skeleton extraction in the second dataset is shown in Figure 9, where (a) adopted a kind of morphological operation method, and (b) adopted thinning algorithm, respectively. The comparison result of the two datasets is shown in Table 1.

For the raw dataset, we firstly perform the image preprocessing over it to obtain numerical database.

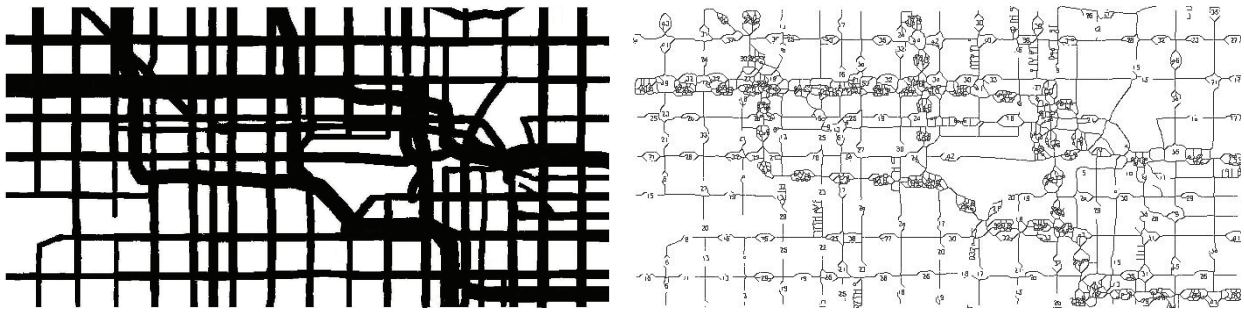
The results of the skeleton extraction, as shown in Figures 8(b) and 9(b), are more clearly and useful for the following processing. Subsequently, the clustering by grid can be readily obtained from the preprocessed images. The extent of image thinning is better and more complete by the thinning algorithm than the *Bwmorph* function in MATLAB. But the elapsed time is longer due to a two-layer iteration nesting procedure in the program code.

The choice of placing a grid on the image follows one principle: mesh segmentation is not trying to fall on a concentrated position of traffic flow. Since there is no endpoint, the midpoint of the two adjacent values was considered a demarcation point. Under this assumption, the traffic flow in each grid is calculated and stored digitally in an Excel file. A digital data for the traffic map serves as the initial data for the subsequent clustering process.

5.2. Comparison Result of KM and HC Clustering. In XLMiner, two methods were used to perform clustering: KM and HC. In order to compare the two methods for the two datasets, input variables were normalized, and the number of clusters is set at five and maximum iterations at 100. The initial centroids are chosen randomly at start. Furthermore,

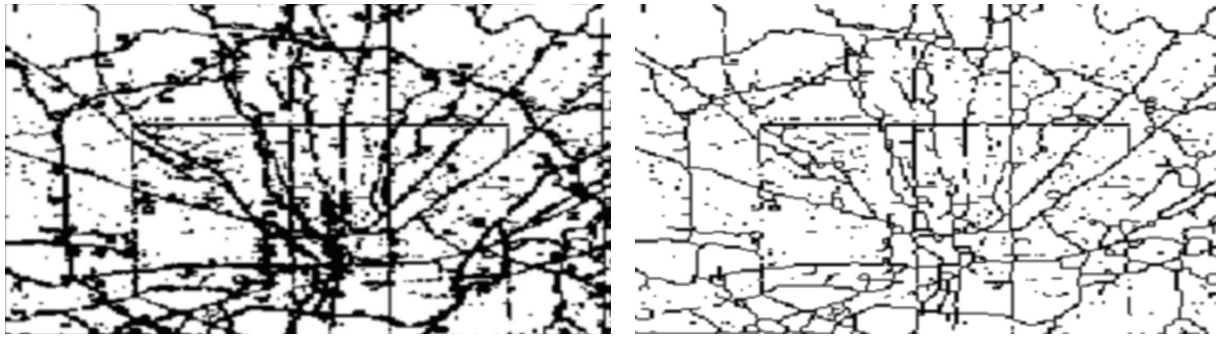
Table 4: Comparison for log-likelihood of first dataset.

Formats	KM	HC	DBScan	XM	EM
Vector database	−12.41868	−14.07265	−13.28599	−11.9533	−12.49562
Raster database	**−13.42238**	**−15.02863**	**−13.78889**	**−12.9632**	**−13.39769**
RasterP (16 grids)	12.62264	−14.02266	−12.48583	−12.39419	−12.44993
RasterP (25 grids)	**−12.41868**	**−13.19417**	**−11.22207**	**−12.48201**	**−11.62048**



(a) (b)

Figure 8: (a) Result of skeleton extraction in dataset 1 using *Bwmorph* function. (b) Result of skeleton extraction in dataset 1 using thinning algorithm.



(a) (b)

Figure 9: (a) Result of skeleton extraction in dataset 2 using *Bwmorph* function. (b) Result of skeleton extraction in dataset 2 using thinning algorithm.

Table 5: Comparison for running time of the second dataset.

Formats	KM	HC	DBScan	XM	EM	LP
Vector database	1.39	13.4	15.53	1.53	10.05	3.37
Raster database	2.41	14.78	18.34	2.17	8.23	1.96
RasterP (16 grids)	0.47	8.01	12.74	0.45	3.77	1.44
RasterP (25 grids)	0.35	6.20	10.98	0.36	2.96	1.18

the weights for the corresponding three attributes (x, y, v) for each grid $(g_i = (x_i, y_i, v_i))$, based on defining weight of x and y could be varied (fine-tuned) and the sum of weights must be equal to 1. We tested several variations searching for the best clustering results: (1) weight of v is 20%; (2) weight of v is 40%; (3) weight of v is 50%; (4) weight of v is 60%; (5) weight of v is 80%; (6) all of them have same weight at 33.3%; (7) weight of v is 0; (8) same weight except when $g_i(v_i = 0)$; and (9) weights of x and y are both 0 except when $g_i(v_i = 0)$.

In HC method, normalization of the input data was chosen. Another option available is similarity measure. It adopts Euclidean distance to measure raw numeric data. Meanwhile, the other two options, Jaccard's coefficients and matching coefficient are activated only when the data is binary.

For the above nine cases, results of cases (1) to (6) are similar in their separate methods. And result of (9) is the worst, which does not accomplish any clustering. Results of cases (2), (3), (7), and (8) are demonstrated in Figure 10.

For the distribution of clusters in the result of KM clustering method, more than half of data points are clamped into one oversized cluster. The result of this method is, therefore, not helpful for further operation. For HC method, data on average are allocated into separate clusters. The result

Table 6: Comparison for log-likelihood of second dataset.

Formats	KM	HC	DBScan	XM	EM
Vector database	−17.35412	−19.62367	−17.53576	−17.21513	−16.57263
Raster database	−18.15926	−20.12568	−19.70756	−18.15791	−18.48209
RasterP (16 grids)	−15.51437	−17.24736	−16.37147	−17.01283	−15.66231
RasterP (25 grids)	−14.84761	−16.63789	−15.09146	−16.67312	−16.47823

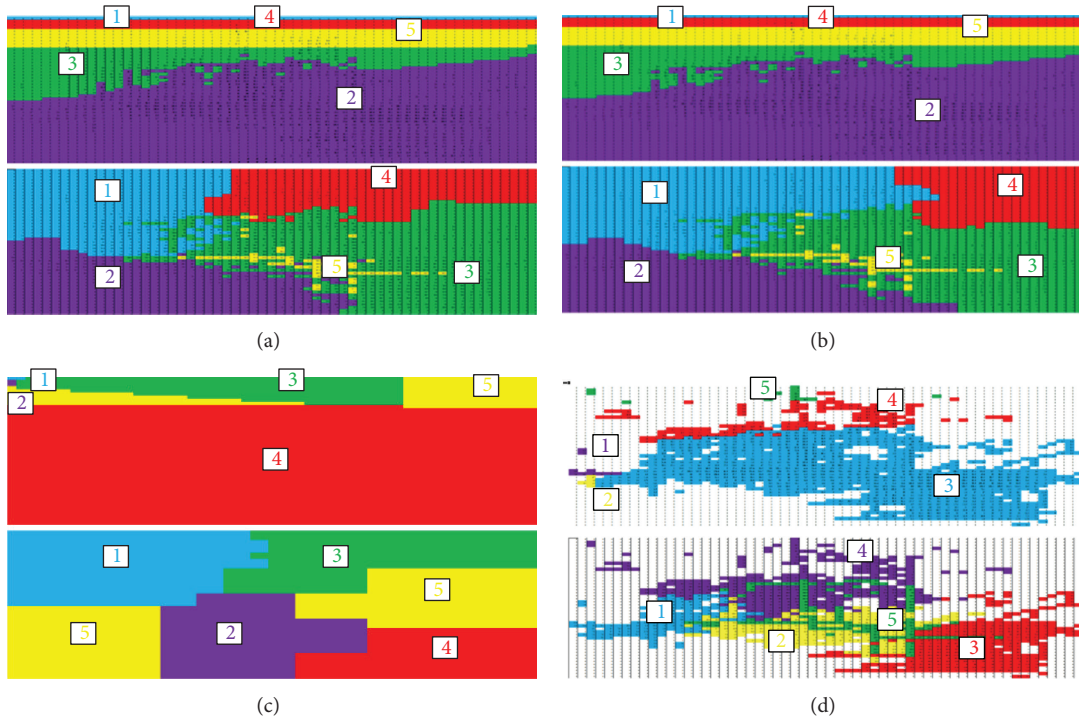


Figure 10: (a) Clustering results for the first dataset with setting case (2) where weight of v is 40%, top half uses KM clustering method, and bottom half uses HC method. (b) Clustering results for the first dataset with setting case (3) where weight of v is 50%, top half uses KM clustering method, and bottom half uses HC method. (c) Clustering results for the first dataset with setting case (7) where weight of v is 0%, top half uses KM clustering method, and bottom half uses HC method. (d) Clustering results for the first dataset with setting case (3) where all share the same weight except $g_i(v_i = 0)$, top half uses KM clustering method, and bottom half uses HC method.

Table 7: Comparison of running time (in seconds) of four different sizes of dataset.

Dataset size	KM	HC	DBScan	XM	EM	LP
100 grid cells	*0.06*	0.07	1.05	2.19	3.21	0.18
4600 grid cells	0.42	2.95	39.89	2.73	19.05	9.37
10000 grid cells	2.62	46.67	97.55	2.97	37.85	24.21
80000 grid cells	19.75	189.61	**684**	6.47	198.31	90.83

in Figure 10(c) is the best, showing only the one with distinct position attributes (x and y). The other three results (Figures 10(a), 10(b), and 10(d)) are stained with cluster overlaps. Therefore, allocation of critical resource, for example, in each cluster may result in a waste of resources. The degree of overlap is the least in the result of Figure 10(b). If only location is being considered, the result of Figure 10(c) is the best choice. Otherwise, the result in Figure 10(b) is better than the other two for the sake of cluster distribution.

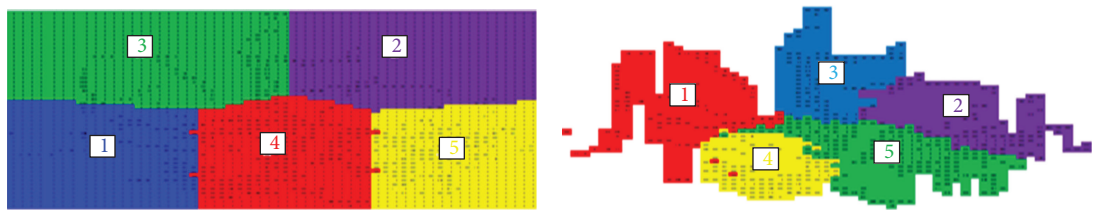
The clustering results of the second dataset performance by using the two methods, KM and HC, are shown in Figure 11.

From the results of the cluster distribution of the second dataset obtained by both clustering methods, the size of each cluster is more or less similar, which is better than that of the first dataset. And there is no overlap phenomenon in the KM results. This is a promising feature of KM method for spatial clustering. However, there is little overlap in the result of HC method as the clusters seem to take irregular shapes. Above all, for the second dataset, KM is a better choice for consideration of even cluster distribution and overlap avoidance by using both clustering methods.

5.3. Results of Grouping. In this part, we compare the colored map of Raster (x, y, v) data model in two datasets using five clustering methods in Weka and the LP method. The common requirement is no overlap for each of the resulting maps. The number of cluster is arbitrarily chosen at five. The

Table 8: Numeric results of coverage of each cluster by using the six methods for dataset 1.

Cov-db1	KM	EM	DBScan	XM	HC	LP
Cluster 0	0.029436	0.003786	0.017902	0.075178	0.013153	0.028985
Cluster 1	0.301538	0.269602	0.208078	0.049761	0.026016	0.377034
Cluster 2	0.215277	0.001627	0.158439	0.084049	0.12436	0.080099
Cluster 3	0.046788	0.096221	0.079177	0.209390	0.001172	0.217204
Cluster 4	0.002712	0.161799	0.044197	0.043152	0.3043	0.007704
Total coverage	0.595751	0.533036	0.507793	0.461531	0.469	0.711025



(a) (b)

Figure 11: (a) Clustering results for the second dataset by using KM method. (b) Clustering results for the second dataset by using HC method.

result of first dataset is shown in Figure 12. The first part (i) of Figure 12 shows the spatial clustering result; the second part (ii) visualizes the corresponding spatial groups by using (a) EM method, (b) KM method, (c) HC method, (d) XM method, and (e) DBScan method. The centers of the clusters are computed after clustering is done, and then the groups are visualized over the clustering results according to the computed centers.

In Figure 12, for the results of (a) and (e), the sizes of clusters are quite uneven, more than half of dataset fall into one cluster. Thus, this result reveals a fact that the technique cannot organize a dataset into homogeneous and/or well-separated groups with respect to a distance or, equivalently, a similarity measure. The corresponding groups have overlap phenomenon too. For the result of (c), the sizes of the clusters are uneven too. For the result of (b) and (d), the sizes of cluster seem to be similar to each other. There is also no overlap in the clustering result, but for group result; the groups in (d) have far more overlaps than those in (b). Overlap means some part or the cluster gets in the way of another one, which means that there is superposition between two or more different clusters. Again, it may cause resource waste and even false allocation. This situation occurs in important fields of applications such as information retrieval (several thematic for a single document) and biological data (several metabolic functions for one gene). For this reason, (b) is better than (d). According to the above analysis, for the result of clustering and corresponding groups, (d) XM is so far the best choice of clustering algorithm as evidenced by the colored maps thereafter.

With the same experiment setup and operating environment, the spatial clustering experiments are performed over the second dataset. The results of second dataset are shown in Figure 13, where (i) represents the spatial clustering result and (ii) represents the corresponding spatial group by using (a) EM method, (b) KM method, (c) HC method, (d) XM method, and (e) DBScan method.

In Figures 13(a) and 13(e), it is noticed that the clusters are imbalanced, and there are overlaps in the corresponding spatial groups using the method of (a) EM and (e) DBScan. The results of (b) KM and (d) XM, however, avoid the shortcomings of (a) and (e) though they still have slight overlaps. For (c) HC, we remove the empty cells in the boundary to reduce the size of dataset; the clustering result is perfect. There is no overlap and clusters are balanced between each other. But there is still overlap in the spatial groups. Thus, LP method is adopted to solve this problem, and in possession of same size of groups. The result of LP method yields perfectly balanced groups without any overlap as shown in Figure 13(f).

By visually comparing the clustering results of the two datasets, the clustering results seem to be similar, but the spatial groups are somewhat different. Occurrence of overlaps in spatial groups is more severe in the first dataset than in the second one. The overlaps are likely due to data distribution and balance in sizes between each cluster. For the first dataset, the heavy spatial values, which are traffic volumes in this case, are mainly concentrated in the center region (city center), so locations of the computed clusters tend to cram very near at a crowded spot. In contrast, the traffic volumes in the second dataset are dispersed over a large area. As seen from the visual spatial groups of the second dataset, the cluster positions are a little far apart when compared to those in the first dataset.

Based on the results generated from the clustering and LP experiments, some statistic information of dataset 2 is collected and it is shown in Table 2. The numeric results in

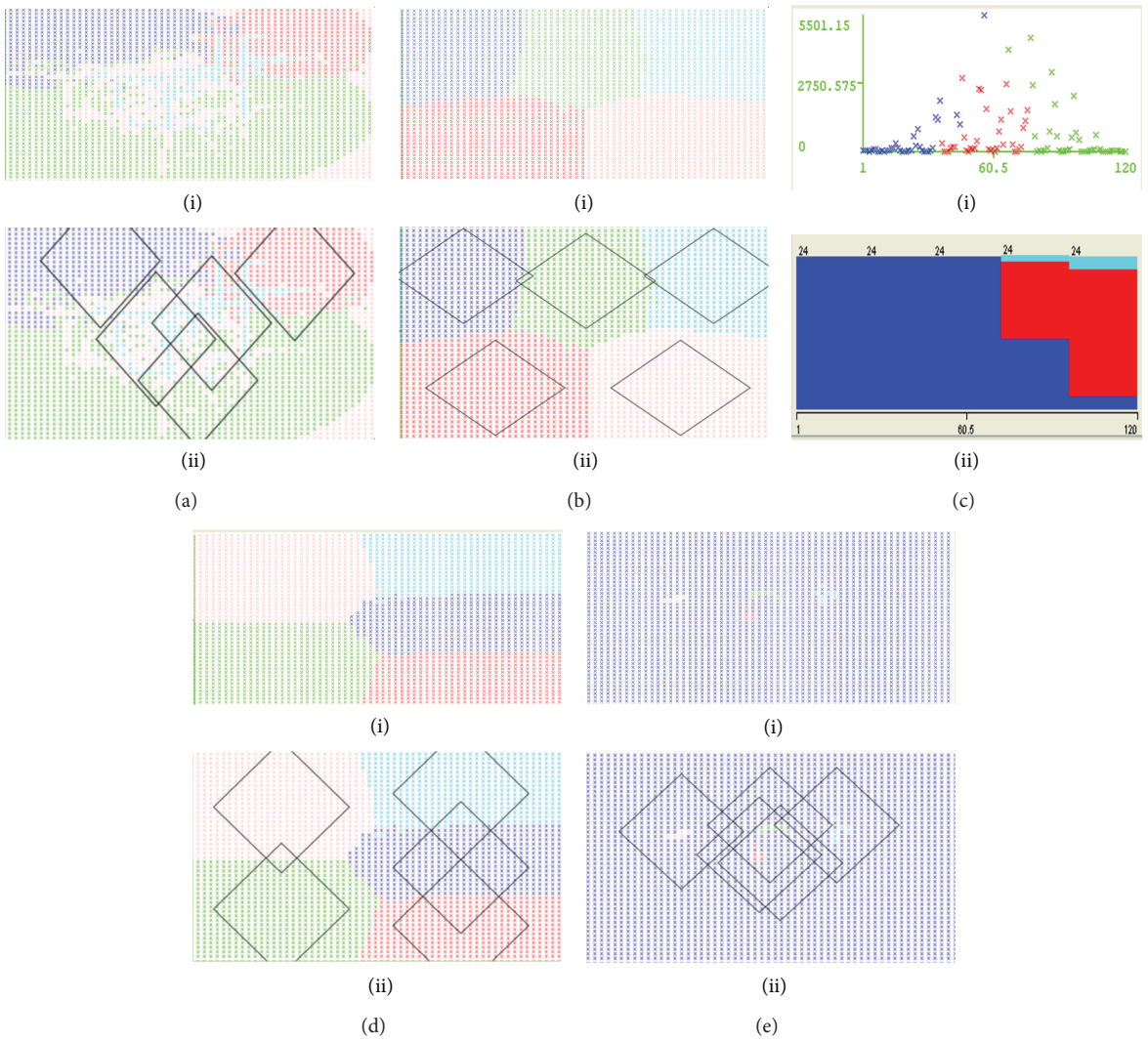


FIGURE 12: (a) (i) Spatial clustering on dataset 1, by using EM. (ii) Spatial groups on dataset 1, from the results of using EM. (b) (i) spatial clustering on dataset 1, by using KM. (ii) Spatial groups on dataset 1, from the results of using KM. (c) (i) spatial clustering on dataset 1, by using HC. (ii) Spatial groups on dataset 1, from the results of using HC. (d) (i) spatial clustering on dataset 1, by using XM. (ii) Spatial groups on dataset 1, from the results of using XM. (e) (i) spatial clustering on dataset 1, by using DBScan. (ii) Spatial group on dataset 1, from the results of using DBScan.

Table 3 support the qualitative analysis by visual inspection in the previous section. By comparing HC and LP methods as an example, the quantitative results show that they have the greatest differences in cell numbers covered by the clusters; also the amount of overlap in HC is the highest of all. By the LP method, the size of each cluster is exactly the same, and they are totally free from overlap.

6. Technical Analysis of Clustering Results

6.1. Experimental Evaluation Method. For the purpose of assessing how the qualities of spatial groups from clustering are, several evaluation factors are defined here: running time (short for time), balance, log-likelihood, overlap, density, and coverage. For a fair comparison, the datasets are run in the same software environment of the same computer. And assume the number of groups to be five with six different methods. Running time is the time we used to run each method using the same software in the same computer to completion. Balance is used to measure the sizes of groups; if balanced, the size of each group is the same. Log-likelihood is an important measure for clustering quality, the bigger the value the better. Weka tests for goodness-of-fit by the likelihood in logarithm, which is called log-likelihood. A large log-likelihood means that the clustering model is suitable for the data under test. Overlap means that the spatial values (e.g., traffic volumes sensed by the sensors) do belong to more than one cluster. Density is the average spatial values (traffic volumes) per grid cell in each cluster. Coverage of a cluster means the proportion of traffic volumes that are covered by the grid cells within the cluster, over the whole dataset; meanwhile, total coverage is the sum of

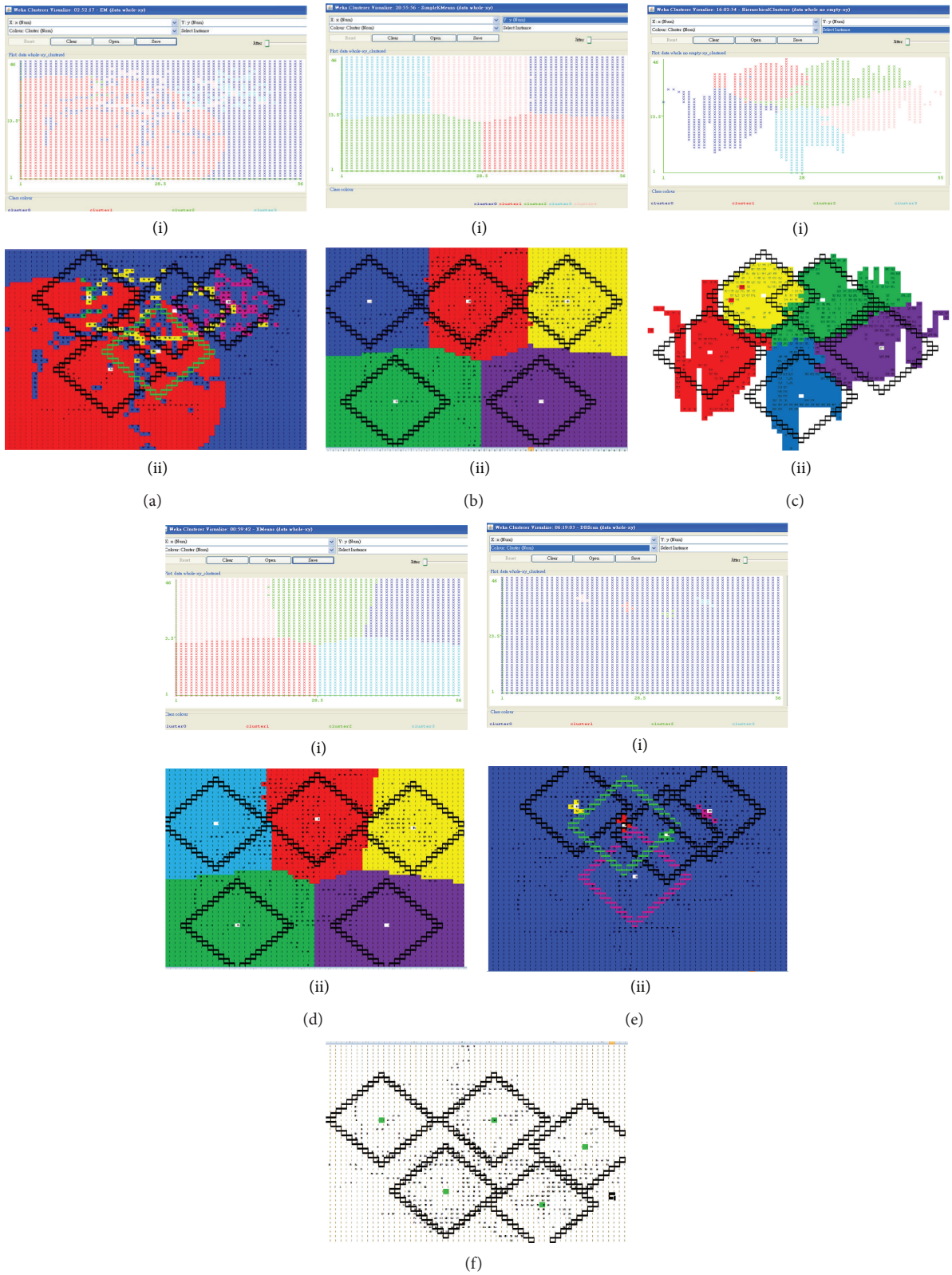


FIGURE 13: (a) (i) spatial clustering on dataset 2, by using EM. (ii) Spatial groups on dataset 2, from the results of using EM. (b) (i) spatial clustering on dataset 2, by using KM. (ii) Spatial groups on dataset 2, from the results of using KM. (c) (i) spatial clustering on dataset 2, by using HC. (ii) Spatial groups on dataset 2, from the results of using HC. (d) (i) spatial clustering on dataset 2, by using XM. (ii) Spatial groups on dataset 2, from the results of using XM. (e) (i) spatial clustering on dataset 2, by using DBScan. (ii) Spatial group on dataset 2, from the results of using DBScan. (f) Spatial group in LP method on dataset 2.

traffic volumes that are covered by all the clusters minus the overlap if any. The corresponding definitions are shown in the equations below.

$$\begin{aligned}
&\text{Density (cluster } i) = \frac{\sum \text{Traffic Volumes (cluster } i)}{\text{Grid Cell Number (cluster } i)} \\
&\text{Coverage (cluster } i) = \frac{\sum \text{Traffic Volumes (cluster } i)}{\sum \text{Grid Cell Number}} \\
&\text{Total Coverage} = \sum \text{Traffic Volumes} - \text{Overlaps} \\
&\text{Proportion of Cluster } (i) \text{ Size (Balance)} \\
&\quad = \frac{\text{Grid Cell Number (cluster } i)}{\sum \text{Grid Cell Number}}.
\end{aligned} \tag{4}$$

6.2. Comparison Experimental Result. After conducting a number of experiment runs, we select four different formats of datasets to perform the clustering algorithm for the first dataset. Vector (n, v) represents sequence n and traffic volume v; Raster (x, y, v) represents coordinates (x, y) and traffic volume v; RasterP (16 grids) means every four neighborhood cells over a grid merged into a single unit; and RasterP (25 grids) means every five neighborhood cells over a grid merged as one. In the other two types of formats, the data information is straightforwardly laid on a grid, and some noises such as outlier values are eliminated from the grid. We selected grids of sizes 16 and 25 for the two formats. The original datasets are then encoded by the four different data formatting types. The four formatted data are subject to the five clustering methods and LP method. We measure the corresponding running time and log-likelihood. The results of the two measurements are shown in Tables 3 and 4, respectively.

According to Table 3, we can see that KM spent the least running time for the four different kinds of data, but the runtime of RasterP (25 grids) dataset is the fastest. Contrariwise, clustering of vector dataset using DBScan method spent the longest running time. Among the clustering methods, KM spent the least time for different datasets and DBScan took the longest.

In Table 4, we evaluate the log-likelihood of the clusters found by each cluster, which is a main evaluation metric for ensuring quantitatively the quality of the clusters. From this table, we can see that the value of log-likelihood of the five methods is quite similar. Among them, clustering of Raster dataset using HC method is the best one, but clustering of RasterP (25 grids) using DBScan is the worst one.

In the same experimental environment, the running time and log-likelihood are shown in Tables 5 and 6 for the second dataset. And in order to stressfully test the performance, we elongate the dataset to larger sizes by expanding the data map via duplication. Running time trends are, therefore, produced; the result is shown in Table 7, and corresponding trend line is shown in Figure 14.

According to Table 5, we can see that KM spent the shortest running time for the four different formats of data, but the time of RasterP (25 grids) dataset is the fastest which is expected because it abstracts every 25 cells into one. On the other hand, clustering of Raster dataset using DBScan method spent the most running time. For the different six methods, KM spent the shortest time for different datasets and DBScan spent the longest time, generally.

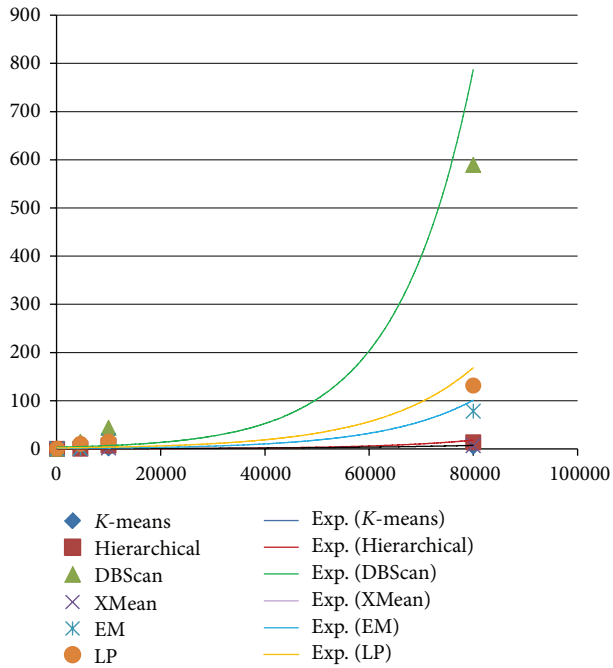


Figure 14: Comparison of running time (in seconds) of different sizes of dataset.

In Table 6, we can see that the values of log-likelihood of different six methods are quite similar. Among them, clustering of Raster dataset using HC method is the best one, but clustering of RasterP (25 grids) using KM is the worst one.

In Table 7, we can see that the slowest is DBScan, and the quickest is KM method. In terms of time trend, DBScan increases in larger magnitude of time consumption than other methods, but time trends of LP, KM, and XM are of lower gradients. In particular, there is an intersection between the trend lines of HC and EM. It means that when the size of dataset exceeds that amount at the intersection, EM method becomes a better choice than HC.

The following charts and tables present the other technical, indicators such as coverage, density, and balance, of each cluster for the two datasets.

From Figure 15, we can see that one cluster of DBScan dominates the biggest coverage in all clusters as results from the six methods in the first dataset. But for the second dataset, LP method yields the biggest coverage cluster. Generally, the individual coverage of each cluster in the second dataset is apparently larger than those resulted from the first dataset (Tables 8 and 9). This means that the second dataset is suitable for achieving spatial groups with the six methods due to its even data distribution. In terms of total coverage, LP achieves the highest values in both cases of datasets. In summary, LP is by far an effective method to determine spatial groups with the best coverage.

Table 9: Numeric results of coverage of each cluster by using the six methods for dataset 2.

Cov-db2	KM	EM	DBScan	XM	HC	LP
Cluster 0	0.042721	0.001777	0.450720	0.022150	0.013153	0.165305
Cluster 1	0.094175	0.086211	0.008018	0.010064	0.026016	0.127705
Cluster 2	0.328026	0.032893	0.010517	0.126953	0.124360	0.095597
Cluster 3	0.022797	0.351221	0.000501	0.311761	0.001172	0.089008
Cluster 4	0.062281	0.101199	0.000244	0.112973	0.304300	0.122085
Total coverage	0.550000	0.573301	0.470000	0.583900	0.469000	0.599700

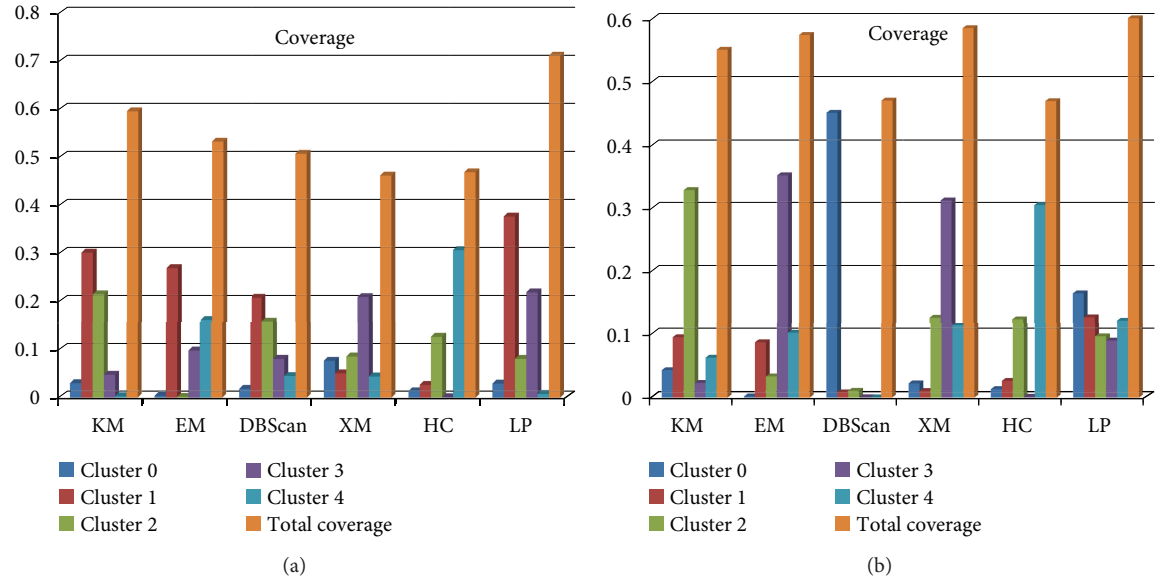


Figure 15: (a) Coverage of each cluster by using the six methods for dataset 1. (b) Coverage of each cluster by using the six methods for dataset 2.

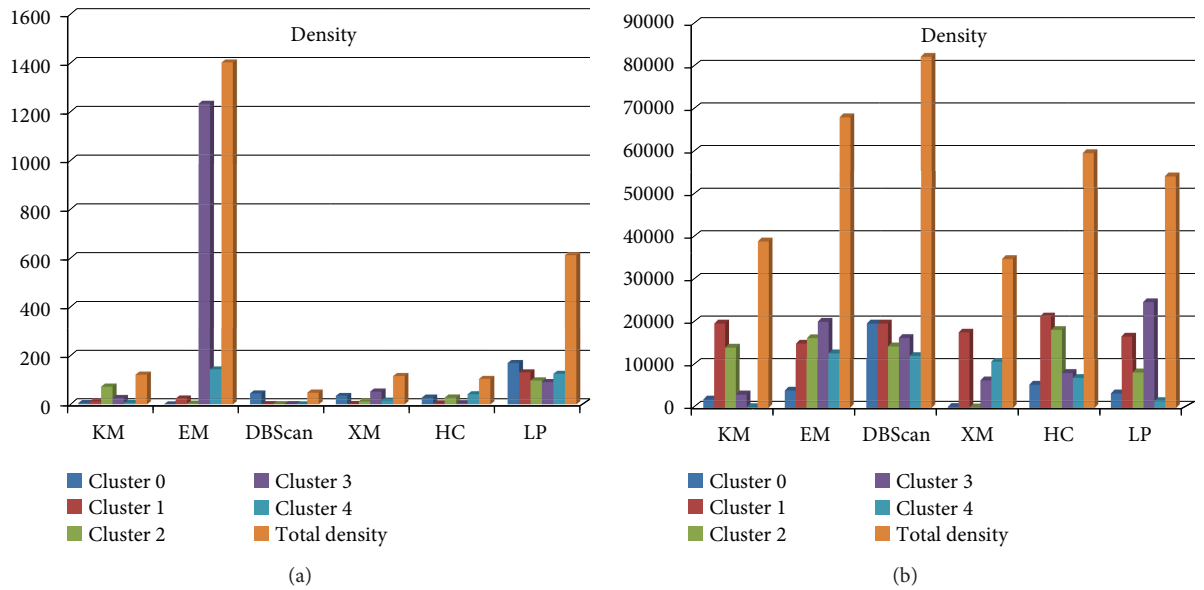


Figure 16: (a) Density of each cluster by using the six methods for dataset 1. (b) Density of each cluster by using the six methods for dataset 2.

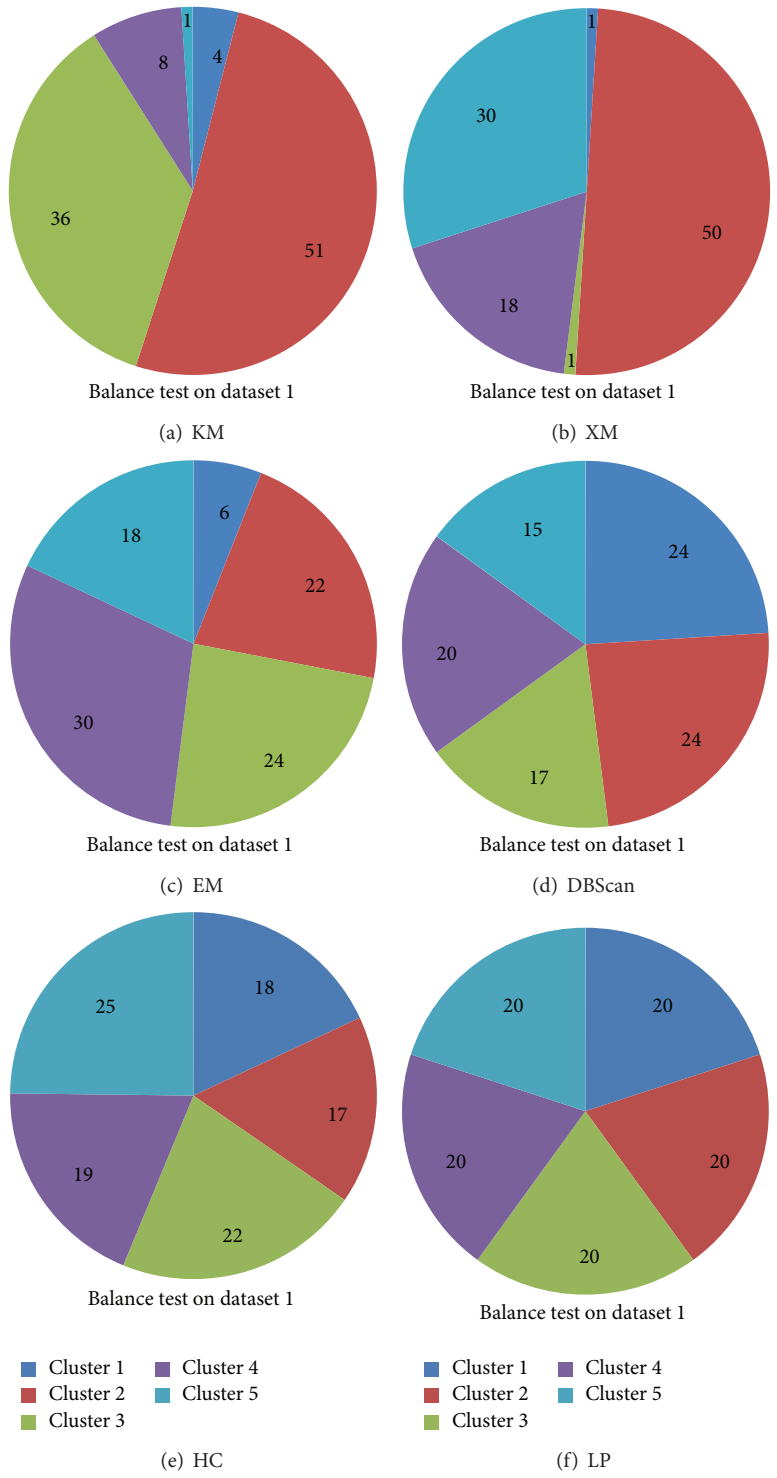


FIGURE 17: Proportions of cluster sizes (balance) of dataset 1 in % by using (a) KM, (b) XM, (c) EM, (d) DBScan, (e) HC, and (f) LP.

From Figure 16(a), we can see that one cluster of EM occupies the biggest density in all clusters of the six methods in the first dataset. But, the LP method obtains the largest total density evenly from all the clusters. Generally, the individual density of each cluster in the second dataset is much bigger than that of the first dataset (Tables 10 and 11). Again it means that the second dataset has an even data distribution that is suitable for achieving spatial groups with high density. And in terms of total density, EM is the best performer in the first dataset, but DBScan achieves the best results in the second dataset. DBScan has an advantage of merging scattered data into density groups as long as the data are well scattered.

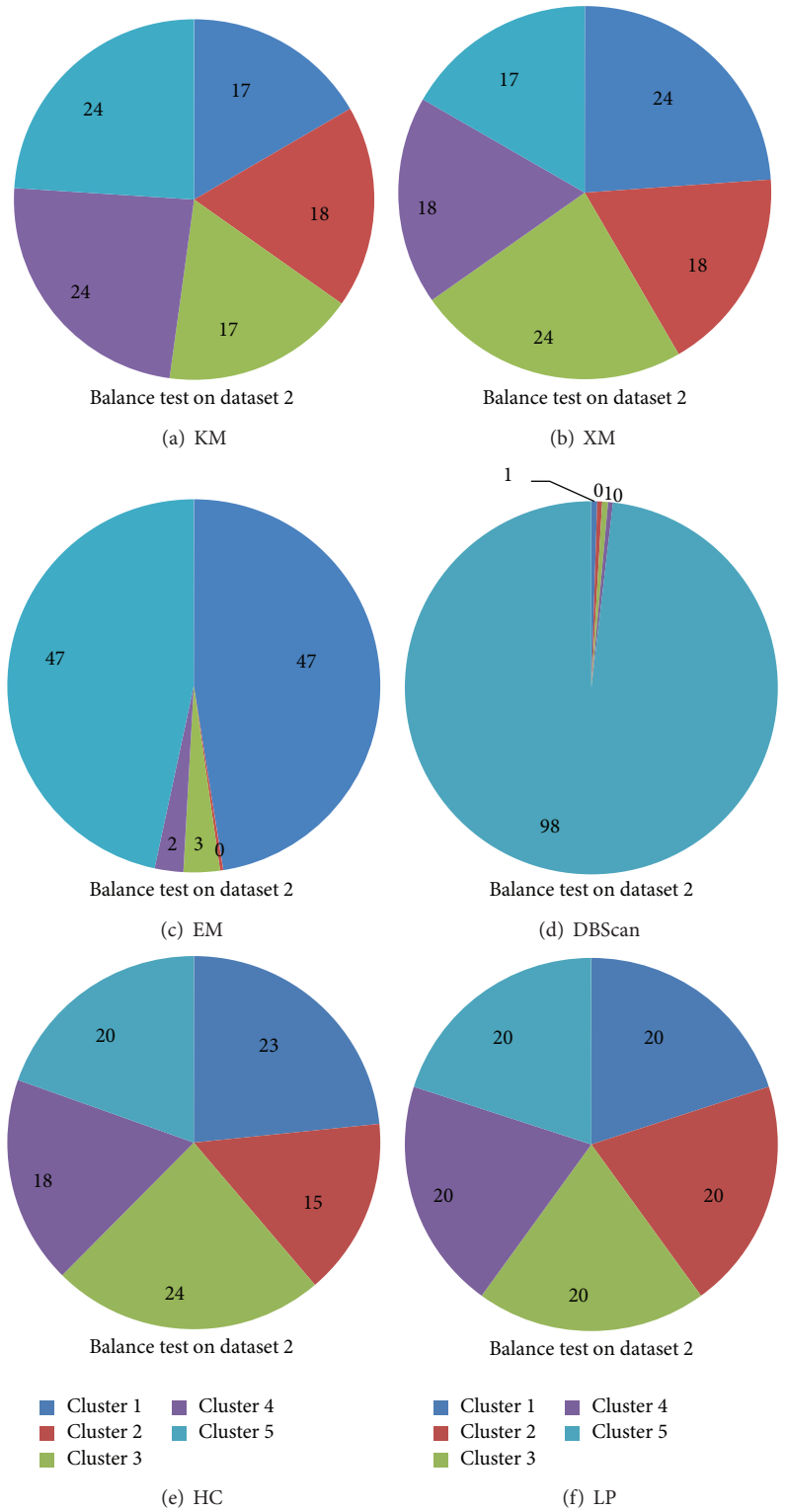


Figure 18: Proportions of Cluster Sizes (Balance) of dataset 2 in % by using (a) KM (b) XM (c) EM (d) DBScan (e) HC (f) LP.

The last evaluation factor is balance; the results are shown in Figures 17 and 18. For both datasets, only LP method can achieve absolute balance for spatial groups completely.

6.3. Discussion of G_{net}. For all the six evaluation factors, each of them can be an individual measure to decide whether a method is good or not in certain aspect. In general, the following indicators (from (5) to (11)) have been defined in

Table 10: Numeric results of density of each cluster by using the six methods for dataset 1.

Density	KM	EM	DBScan	XM	HC	LP
Cluster 0	5.258648	0.080823	44.26289	34.31892	27.13810	167.7869
Cluster 1	11.61390	23.29182	0.994949	1.375497	3.501739	129.6230
Cluster 2	71.86556	2.545750	0.807500	12.18667	27.28017	97.03279
Cluster 3	25.72683	1232.386	1.062069	51.71040	4.265905	90.34426
Cluster 4	5.969350	142.054	0.170455	15.10576	40.88438	123.9180
Total density	120.4343	1400.359	47.29787	114.6972	103.0703	608.7049

Table 11: Numeric results of density of each cluster by using the six methods for dataset 2.

Density	KM	XM	EM	DBScan	HC	LP
Cluster 0	1925.445	247.6642081	3968.13638	19723.94643	5323.785326	3313.18
Cluster 1	19723.95	17634.96208	15026.98729	19723.94643	21404.82869	16678.8
Cluster 2	14081.49	106.489095	16297.95665	14371.89548	18238.21619	8097.989
Cluster 3	3060.449	6293.956697	20151.05986	16363.50955	7991.2225	24744.92
Cluster 4	177.3937	10583.46213	12752.99493	12123.17249	6856.982634	1569.58
Total density	38968.73	34866.53421	68197.13511	82306.47036	59815.03534	54404.47

order to evaluate which method is an appropriate choice when it comes to different datasets and different users' requirements. Among them, the difference in balance is contributed by the difference of grid cell number in each cluster. Meanwhile, we assign each of them a proportional weight ω to adjust the evaluation result G_{net}. The ω value is to be tuned by the users depending on their interests. For example, if a very wide coverage is of priority and others are of less concern, G_c can take a relatively very large value or even 1. If users consider that some attributes are more important, the corresponding weights ω for some factors can be larger than the others. Overall, G_{net}, which is the sum of all factors multiplied by the corresponding performance indicators, is a net indicator signifying how good a clustering process is, by considering all the performance attributes:

$$G_l = \left|\frac{\text{Likelihood}}{\text{Time}}\right|, \tag{5}$$

$$G_b = \frac{\text{Difference of Balance}}{\text{Time}}, \tag{6}$$

$$G_d = \frac{\text{Density}}{\text{Time}}, \tag{7}$$

$$G_c = \frac{\text{Coverage}}{\text{Time}}, \tag{8}$$

$$G_o = \frac{\text{Overlap}}{\text{Time}}, \tag{9}$$

$$G_{\text{net}} = \omega_l G_l + \omega_d G_b + \omega_d * G_d + \omega_c G_c + \omega_o G_o, \tag{10}$$

$$\text{Constraint: } \omega_l + \omega_d + \omega_b + \omega_c + \omega_o = 1. \tag{11}$$

From the results of spatial grouping as experimented in the previous sections, we obtain some statistic information on each group based on the second dataset as a range of indicators depicted from (5) to (11). They are shown in Table 12 which allows us to easily compare various methods and performance aspects.

In Table 12, KM method has the best run time and no overlap. For XM method, DBScan and HC demonstrate their advantage in density and log-likelihood. Nevertheless, LP method is superior in three aspects: coverage, no overlap, and zero difference of balance with other clusters. In order to further verify the correctness of the above analysis, the performance indicators G_l, G_b, G_d, G_c, and G_o are computed for obtaining the net performance values G_{net} assuming equal weights for each method. For the sake of easy comparison, G_{net} is normalized by first setting the lowest G_{net} among the six methods as base value 1; then the G_{net} for the other methods is scaled up accordingly. The comparison result is shown in Table 13.

According to the experiment results conducted so far, LP seems to be the best candidate in almost all the aspects, such as coverage and balance. This is tested across different datasets, different formats, and different sizes of dataset. However, for density and log-likelihood, the result is not so consistent, as LP would be outperformed by DBScan at times. Finally, by the net result of G_{net}, LP is a better choice under the overall consideration of the six performance factors. The choice of weights which imply priorities or preferences on the performance aspects should be chosen by the user's discretion.

7. Conclusion and Future Works

Ubiquitous sensor network generated data that inherently have spatial information. When they are viewed afar, the localizations of the data form some densities spatially distributed over a terrain, and the collected data from the sensors indicate how important the values are in their local proximity. Given this information, the users of the sensor network may subsequently want to form spatial clusters for

Table 12: Performance indicators of the six methods based on dataset 2.

Method	Coverage	Density	Time	Log-likelihood	Overlap	Diff. of balance
KM	0.595751	38968.73	0.41	−17.35	No	190
XM	0.533037	34866.53	0.67	−17.22	No	185
EM	0.507794	68197.14	1.23	−16.57	Yes	1216
DBScan	0.461531	82306.47	15.67	−17.54	Yes	2517
HC	0.677124	59815.04	14.78	−20.13	Yes	103
LP	0.711025	54404.47	7.76	N/A	No	0

Table 13: Comparison of different clustering and LP methods by G_{net} indicator.

Methods	KM	XM	EM	DBScan	HC	LP
G_{net}	1.08	1.15	1.11	1.23	1.00	1.32

purposes such as resource allocation, distribution evaluations, or summing up the geographical data into groups. The focus of this study was to design efficient methods to identify such optimal spatial groups that have certain sizes and positions using clustering algorithms or the equivalent, for obtaining maximum total coverage in total. Some examples include but are not limited to setting up mobile phone base stations among an even distribution of mobile phone users, each may have different demand in usage; distributed sensors that monitor the traffic volumes over a city, and security patrols in an exhibition where the asset values to be protected vary and are distributed over a large area. The study also investigated whether spatial groups identified by using different methods are sufficiently efficient for achieving optimal maximum coverage. Five classic spatial grouping algorithms are discussed and compared in this study by using data mining software programs. The identified spatial groups with different values of data resources were then assessed via six performance factors. Weights were also formulated as factor coefficients. The factors adopted were shown to play a significant role in MAUT (multiattribute utilities theory). The performance under proper factors and weights may vary as the factors could be arbitrarily chosen by users.

The spatial groups obtained by classic clustering algorithms have some limits, such as overlaps. It may cause resource being wasted and even false grouping. However, there has been no study reported in the literature that the authors are aware of using linear programming (LP) method to discover spatial groups and to overcome this limit of overlapping. Thus, in this research, we implemented this new method (LP) to obtain spatial groups for yielding maximum coverage and completely avoiding overlap. A rigorous evaluation was used to assess the grouping results by considering multiple attributes.

For future extended study, we want to further enhance the algorithm, such as combining LP method with existing spatial group algorithms to achieve new hybrid algorithm. Some clustering algorithms (e.g., KM) are known to converge quickly, and LP though not the quickest, it is efficient in finding the optimal groupings without any overlap. It will be good if the advantages from one algorithm to ride over the others in the new fusion algorithms are to be developed.

References

[1] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[2] K. H. Eom, M. C. Kim, S. J. Lee, and C. W. Lee, "The vegetable freshness monitoring system using RFID with oxygen and carbon dioxide sensor," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 472986, 6 pages, 2012.

[3] G. Manes, G. Collodi, R. Fusco, L. Gelpi, and A. Manes, "A wireless sensor network for precise volatile organic compound monitoring," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 820716, 13 pages, 2012.

[4] Y.-G. Ha, H. Kim, and Y.-C. Byun, "Energy-efficient fire monitoring over cluster-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 460754, 11 pages, 2012.

[5] A. Wahid and D. Kim, "An energy efficient localization-free routing protocol for underwater wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 307246, 11 pages, 2012.

[6] T. N. Tran, R. Wehrens, and L. M. C. Buydens, "SpaRef: a clustering algorithm for multispectral images," *Analytica Chimica Acta*, vol. 490, no. 1-2, pp. 303–312, 2003.

[7] G. Ayala, I. Epifanio, A. Simó, and V. Zapater, "Clustering of spatial point patterns," *Computational Statistics and Data Analysis*, vol. 50, no. 4, pp. 1016–1032, 2006.

[8] J. Domingo, G. Ayala, and M. E. Díaz, "Morphometric analysis of human corneal endothelium by means of spatial point patterns," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 2, pp. 127–143, 2002.

[9] E. Demir, C. Aykanat, and B. Barla Cambazoglu, "Clustering spatial networks for aggregate query processing: a hypergraph approach," *Information Systems*, vol. 33, no. 1, pp. 1–17, 2008.

[10] T. Hu and S. Y. Sung, "A hybrid EM approach to spatial clustering," *Computational Statistics and Data Analysis*, vol. 50, no. 5, pp. 1188–1205, 2006.

[11] G. Lin, "Comparing spatial clustering tests based on rare to common spatial events," *Computers, Environment and Urban Systems*, vol. 28, no. 6, pp. 691–699, 2004.

[12] M. Ester and H.-P. Kriegel, "Clustering for mining in large spatial databases [Special Issue on Data Mining]," *KI-Journal*, vol. 1, pp. 332–338, 1998.

[13] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining: a survey," Tech. Rep., Computer Science, Simon Fraster University, 2000.

[14] H.-D. Yang and F.-Q. Deng, "The study on immune spatial clustering model based on obstacle," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 1214–1219, August 2004.

[15] T.-S. Chen, T.-H. Tsai, Y.-T. Chen et al., "A combined K-means and hierarchical clustering method for improving the clustering efficiency of microarray," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '05)*, pp. 405–408, HongKong, China, December 2005.

[16] M. Srinivas and C. K. Mohan, "Efficient clustering approach using incremental and hierarchical clustering methods," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–7, July 2010.

[17] P. Bajcsy and N. Ahuja, "Location- and density-based hierarchical clustering using similarity analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 1011–1015, 1998.

[18] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proceedings of the International Conference Knowledge Discovery and Data Mining*, pp. 58–65, 1998.

[19] K. Elangovan, *GIS: Fundamentals, Applications and Implementations*, 2006.

[20] S. Chawla and S. Shekhar, "Modeling spatial dependencies for mining geospatial data: an introduction," *Geographic Data Mining and Knowledge Discovery*, vol. 75, no. 6, pp. 112–120, 1999.

[21] M.-Y. Cheng and G.-L. Chang, "Automating utility route design and planning through GIS," *Automation in Construction*, vol. 10, no. 4, pp. 507–516, 2001.

[22] Q. Cao, B. Bouqata, P. D. Mackenzie, D. Messier, and J. J. Salvo, "A grid-based clustering method for mining frequent trips from large-scale, event-based telematics datasets," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 2996–3001, San Antonio, Tex, USA, October 2009.

[23] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 29, no. 3, pp. 433–439, 1999.

[24] D. Pelleg and A. W. Moore, "X-means: extending KM with efficient estimation of the number of clusters," in *Proceedings of the 70th International Conference on Machine Learning*, pp. 727–734, 2000.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[26] P. Bajcsy and N. Ahuja, "Location- and density-based hierarchical clustering using similarity analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 1011–1015, 1998.

[27] J. H. Ward Jr., "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.

[28] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '06)*, pp. 281–286, Pisa, Italy, September 2006.