Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Spatial Data Mining for Optimized Selection of Facility Locations in Field-based Services

A. Zarnani, M. Rahgozar, C. Lucas, and F. Taghiyareh

Abstract—**Spatial data mining has been developed as the effective technique in many applications that involve large amounts of geo-spatial data. Many organizations provide field-based services such as delivery, field-services and emergency to their customers. Considering the geographical distribution of the customer request points, the location of facilities will have noticeable impact on the overall efficiency of the company's operations. The closer the facilities are to the customers, the sooner and cheaper will be the service provision transaction. In this paper, we empirically study the role of spatial clustering methods in such context. We have implemented and tuned some of the main spatial clustering algorithms to discover the best locations for facility establishment. A new spatial clustering algorithm is proposed that does not require the number of facilities as input. The new algorithm will determine the optimal number of facilities along with their locations based on the business context trade-offs. Many experiments are conducted to study the performance of the studied algorithms on real world and synthetic data sets. The results reveal valuable distinctions between the different methods and confirm the higher efficiency of the proposed algorithm.**

I. Introduction

SPTIAL data mining has become a new and powerful tool for efficient and complex analysis of very large geo-spatial databases. The location dimension involved in spatial data is an important aspect in many applications [3]. Such spatial information is now available by the advent of new geo-spatial technologies such as Remote Sensing, GPS and Geo-Coding.

One of the main examples of such applications is in the organizations that provide field-based services such as delivery, emergency and field-services to their customers [1]. Clearly, the location of the facilities that house the technicians, parts and tools play a significant role in the overall field service logistics performance. Therefore, selection of optimal locations for the establishment of new facilities is a critical decision [9]. The total establishment cost, logistics cost and response time are of the main criteria that contribute to this procedure.

More accurately, the customer request points are modeled in the geographic space in addition to other information such as the drive-time distances, candidate locations for facility establishment, regions with different costs for facility establishment; etc. The problem we face is to decide to establish how many facilities and in which locations to minimize the overall cost of covering the customer points. This cost is generally measured by the total amount of distances from the location of each customer request point to its nearest facility.

A similar optimization problem to the above problem is the *discrete p-median* or the facility location problem known to the operations research community [2], [16]. Integer programming is the common approach used in this community and lagrangian relaxation along with other heuristics have been successfully applied for small versions of this problem (involving hundreds of points) [2], [12], [16]. However, as discussed in [5] and [18] our concern is the scalability of such approaches with respect to the large databases encountered in today's applications that contain large number of points (perhaps thousands or more). Also in the p-median problem the number of facilities that are to be established has to be initially given to the algorithm by the user [2]. In addition, the candidate locations are to be manually supplied too. Thus, another concern (which is generally ignored) is the selection of appropriate number of facilities and the candidate locations. This process can degrade the quality of the final solution if performed poorly.

The concern of huge amount of data is the challenge that the knowledge discovery approaches are typically engaged in [3], [19], [20]. As mentioned before, spatial data mining has become a popular and powerful solution for such complex spatial problems. Spatial data mining is defined as the automatic process of discovering interesting and implicit knowledge from large amounts of spatial data [10]. The common high volume of geo-spatial databases has turned the aspects of efficiency and scalability into the main concerns in the design and development of spatial data mining algorithms. Many data mining tasks have been studied in the context of geo-spatial databases and many algorithms have been developed for extraction of interesting patterns and relationships from these databases. In spatial classification,

Manuscript received October 31, 2006. This work was supported in part by the TAKFA Grant Program (Iran's Supreme Council of ICT).

Ashkan Zarnani is with the Database Research Group, Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: a.zarnani@ece.ut.ac.ir).

Masoud Rahgozar is with the Database Research Group of the Control and Intelligent Processing Center of Excellence (CIPCE), Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: rahgozar@ut.ac.ir).

Caro Lucas is with the Database Research Group of the Control and Intelligent Processing Center of Excellence (CIPCE), Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: lucas@ipm.ir).

Fattaneh Taghiyareh is with the Faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: ftaghiyar@ut.ac.ir).

models are extracted from the spatial database to predict a spatial phenomenon. An efficient spatial classification algorithm was developed in [11]. Shekhar *et al.* [17] proposed an improved classification method by considering the spatial autocorrelation concept. Spatial association rules were first defined by Koperski *et al.* [10] who also developed an algorithm for spatial association rule mining. In this algorithm, spatial relationship hierarchies were exploited for higher efficiency. Spatial association rule mining was further improved in [19] and also co-location patterns were introduced in [7]. In [3], [22] algorithms are proposed for efficient discovery of spatial trends which are patterns presenting regular change of some non-spatial attribute in the neighborhood of an object. In spatial clustering which is one of the main areas of spatial data mining [5], [13], we aim to identify subsets of spatial objects having similar characteristics. Spatial clustering algorithms search for a set of representative points that will determine the groupings by assigning each object to its nearest representative. The quality of a set of representatives is generally evaluated by the sum of distances from each point to its nearest representative [5], [13], [20]. This is the same objective that we face in facility establishment i.e. finding locations that are totally closer to customers. Here, the distance has the natural notion of the spatial Euclidean distance or the drive-time distance.

Consequently, we study the application of effective spatial clustering methods as solutions for finding the best spatial location of facilities in field-based services. As scalability is the main challenge and consideration in the developments of clustering algorithms, we think that such algorithms can be used as efficient solutions to the large versions of the facility location selection problem. Hence, the empirical study of these algorithms in such context is an attractive and helpful research.

In addition to comparison between different clustering methods, we propose a new algorithm named Fac-means that can automatically find the optimal number of facilities along with their locations. This is done by considering the trade-off between paying more facility establishment cost and getting closer to the customer points. The experiments compare the different algorithms with each other. The results reveal many advantages and disadvantages between the algorithms when they are applied on the facility location problem.

II. FIELD-BAESED SERVICE AND FACILITY LOCATION SELECTION

A. Field-based Services

The importance of the service support provided after the sail of the product is vivid for any competitive company [1], [9]. There are two main categories of services: facility-based and field-based. In facility-based service customers access the service facility while in field-based service it is the responsibility of the service provider to provide services to customers and/or their possessions, located at the customer's presence site [1]. In field-based service the field worker has to be dispatched to the customer's site with the needed parts and tools. Delivery, emergency and after-sales are examples of field-based services. The logistics is the most important aspect in improving efficiency and effectiveness of such services. Clearly, the location of the facilities that house the field workers, parts and tools play a significant role in the overall logistics performance.

B. Facility Location Selection Problem

The problem we focus on is finding the best locations for the establishment of the facilities so that we can cover the customers with the least logistics cost. The logistics cost is measured by the distance that is to be traveled from a facility location to the customer request points. In fact, we have a spatial point s_i for each customer request point where the set of all customer request points is $S = \{s_0, s_1, ..., s_{n-1}\}$ and n is the total number of customer request points. For each facility there is a spatial point f_i. To improve the logistics performance of the field-based services, we intend to optimize the following criteria:

$$\text{minimize } M(F) = \sum_{i=0}^{n-1} w_i d(s_i, fac[s_i, F]) \quad (1)$$

Where $F = \{f_0, f_1, ..., f_{K-1}\}$ is the set of facilities and *K* is the total number of facilities in the 2-dimentional spatial space $\mathcal{R}^2$. w_i is an optional parameter that shows the number of requests in different times of a customer at point s_i, and

$$d(\vec{p}, \vec{q}) = (|p_x - q_x|^2 + |p_y - q_y|^2)^{1/2} \quad (2)$$

is the Euclidean spatial distance. $fac[s_i, F]$ is the closest facility in *F* to the customer in s_i, that is,

$$d(s_i, fac[s_i, F]) = \min_{j \in \{0,...,k-1\}} d(s_i, f_j) \quad (3)$$

The value formulated in equation 1 is the total amount of distances from the current set of facilities to the covered customer requests. The logistics cost can be obtained from this value considering a basic cost for a specific amount of logistics distance. Thus, we obtain a value $LCost(F)$ which is the total logistics cost incurred with the set of facilities *F*,

$$LCost(F) = c.M(F)/d \quad (4)$$

Where *c* is the financial cost incurred for the logistics distance *d*. There is also a value $ECost(F)$ that formulates the cost needed for the establishment of facilities in *F*,

$$ECost(F) = \sum_{j=0}^{k-1} e_j \quad (5)$$

Where the establishment cost of a facility f_j is e_j .This value can be defined for different geographical regions.

It can also be dependent on the total number of customer requests that the facility is to serve. Assuming that the revenue gained from the customers is constant for different sets of *F*, our purpose is to minimize the cost of covering the customers ($CCost(F)$), that is the sum of $LCost(F)$ and $ECost(F)$ as shown in equation 6.

$$\text{minimize } CCost(F) = LCost(F) + ECost(F). \quad (6)$$

It is clear that the two values in the right side of this equation are inversely dependent on each other. That is, there will be less logistics cost when establishing more facilities by paying more establishment cost.

C. Spatial Clustering Approaches

Clustering is a process that divides a set of objects into several groups (clusters) such that the similarity between the members of the clusters is maximized. In data clustering, the formulation of the problem is the same as the formulation provided in the previous section with the exception that the data points have *m* dimensions. Because having different costs for centroids is meaningless in general clustering problems, Equation 1 is generally used as the final objective [5], [8], [14]. This objective is a measure of dissimilarity in a clustering result. Spatial clustering methods on the other hand focus on the points with 2-dimensions and incorporate the proximity information of the spatial points [5], [14].

Clustering algorithms can be generally categorized into partitioning methods [5], [14], hierarchical methods [23], density-based methods [4] and grid-based methods [20]. In this work we concentrate on the partitioning methods. There are many motivations behind selecting partition-based methods for this problem. First, the hierarchical methods suffer from poor scalability with increasing the number of points. In fact the computational cost incurred is $O(n^2)$ for *n* data points [5]. Second, the main advantage of density-based methods is their ability to find elongated and non-convex clusters [4]. This is a valuable capability in spatial data mining applications. Nonetheless, this is not useful in the problem of finding best locations for facilities since here the objective is to minimize the customers covering cost. Third, grid-based approaches [20] also suffer from some shortcomings as a possible solution to our problem. The performance of these algorithms relies on many parameters such as the granularity of the lowest level of the grid structure and assumptions on data distribution [14]. Also the resulting clusters are bounded horizontally or vertically, but never diagonally.

The above explanations are the main motivations that most of the spatial clustering methods use partitioned-based approaches [5], [12], [13]. Our proposed algorithm, Fac-Means is a partitioning method too. In the following, we review the spatial clustering algorithms that are focused in our study.

III. Spatial Clustering Algorithms

A. K-means

K-means [8] is one of the most basic and widely used algorithms in clustering analysis and can be easily applied to cluster spatial objects. The attractiveness of this algorithm lies in its simplicity. In this algorithm the data points are partitioned into *K* different subsets by assigning each point to the nearest center (equation 3). The number of desired clusters (*K*) and the set of points *S* are provided as the inputs. The steps of the K-means clustering algorithm are shown in Figure 1:

0. Initialize the centroids $f_{0..k-1}$ to random values.
1. Associate each point s_i with the nearest centroid.
2. Recalculate the new centroids for each cluster by taking a weighted average of its member points.
3. If any centroid is changed repeat from step (1) else terminate.

Figure 1. K-means Algorithm

K-means is a deterministic approach that heuristically solves the optimization problem of equation 6 (known to be the clustering error) by finding a local minimum.

$$\text{minimize } M(F) = \sum_{i=0}^{n-1} w_i d^2(s_i, fac[s_i, F]) \quad (7)$$

It has been argued that the consideration of un-squared Euclidean distance as in equation 1 is statistically more robust and preferred in spatial clustering applications [5], compared to the squared error used in equation 6.

The initialization step is crucial since the algorithm converges to the final centroids based on the initial values of the centers. Some methods have been proposed for the fine initialization of the centroids. One study showed that repeating the execution of the algorithm with randomly selected points leads to better results in terms of error and robustness [8]. We use this approach for finding the optimal locations of facilities.

K-means has high computational efficiency as a solution to the facility location problem. In fact, the complexity of an execution of K-means is $O(tkn)$, where K is the given number of facilities of which the optimal locations are to be found, t is the number of iteration and n is the total number of customer request points.

B. PAM

A large proportion of partitioning clustering algorithm is based on K-medoid. In medoid based approaches, the cluster representatives are selected from the data points [8]. This means that $F \subset S$ in equation 1. Once a set of *K* representative points (medoids) are determined, the other (non-medoid) points are grouped based on their distance to the medoids similar to the K-means method. PAM

(Partitioning Around Medoids) [8] is the basic version of K-medoid algorithms.

In each iteration of this algorithm the cost for swapping each non-medoid point (S_p) with each medoid point (S_m) is computed. The cost of a replacement ($Cost_{p,m}$) states the change in the total Euclidean dissimilarity of the clusters ($M(F)$) if the swap is applied. Hence, negative exchange cost means that the objective is improved and vice versa. We refer to [8], [14] for details of calculating the swap cost. In the next step, the best possible replacement is made, leading us to a new set of representatives that are passed to the next iteration. The process is terminated when there is no improving (negative cost) possible swap. Figure 2 shows the detailed steps of this algorithm.

0. Initialize $f_{0..k-1}$ with arbitrary selected points from *S*.
1. Compute $Cost_{p,m}$ for all possible pairs of S_p and S_m where S_m is currently a member of F and S_p is not.
2. If the minimum of $Cost_{p,m}$ is negative, replace S_p with S_m and go to step(1).
3. Else output F as the final set of representatives.

Figure 2. PAM Algorithm

The computational cost of Step (1) alone in PAM is $O(n^2)$. This makes PAM applicable only for small data sets (e.g. 100 points). For this reason we will not study the application of this algorithm in our experimental study. Many algorithms are proposed based on PAM to alleviate its poor efficiency. The best-known algorithms among these are CLARA and CLARANS which will be reviewed in the next subsection.

C. CLARA

A random sampling approach is used in CLARA (Clustering LARge Applications) [8] to handle the large number of points in recent applications such as data mining. The key point is that the appropriate sample sizes can effectively maintain the important geometrical properties of the entire data set. To improve the efficiency of PAM, CLARA, applies PAM to find the representative medoids only in a randomly drawn sample from the data set. For better approximation, CLARA repeats this process with multiple samples and the set *F* which leads to the least error (with regards to *all* of the data points) is given as the output. Experiments in [8], [14] show that 5 samples of size $40+2k$ lead to satisfactory efficiency and effectiveness.

0. For *i*=1 to 5:
1. Run PAM on a random sample of $40+2k$ points.
2. Associate each point s_i with he nearest medoid and calculate $M(F)$.
3. If $M(F)$ is less than $\min M(F)$ set minimum to this value and retain *F* as the best set of medoids obtained so far.

Figure 3. CLARA Algorithm

The sampling technique used in CLARA improves the efficiency of PAM as its computational cost becomes $O(n)$. However, there is a trade off in losing cluster quality by sampling fewer points from the whole data set. This trade off has to be adjusted based on the application which CLARA is to be applied.

D. CLARANS

CLARANS (Clustering Large Applications based on RANdomized Search) [14] applies a randomized search on a graph representation of the clustering problem named $G_{n,k}$. Each node of this graph is a set of *K* possible medoids named F_i. The nodes whose sets differ in only one medoid (i.e. $|F_i \cap F_j| = K-1$.) are considered as neighbors. Thus, each node of this graph has $k(n-k)$ neighbors. The quality of a node which is a possible clustering is evaluated by the objective in equation 1 called the cost function.

In fact, PAM performs an exhaustive search on this graph by calculating the difference of the cost function between the current node and all of its neighbors in Step (1) of this algorithm. CLARANS does not apply a sampling approach like CLARA and does not perform an exhaustive search like PAM. Instead, it uses a randomized search so that the sampling approach is applied on the neighbors of the current node in each step rather than the entire data set once in the beginning of the search. The steps performed in CLARANS are depicted in Figure 4.

0. Initialize numlocal and maxneighbor to user given values and set i to 1.
1. Set $F_{current}$ to a node F_i randomly chosen from $G_{n,k}$.
2. Set j to 1.
3. Arbitrarily select a node F_j from the neighbors of $F_{current}$ and calculate the change in cost similar to PAM.
4. If F_j has a lower cost (better clustering quality) set $F_{current}$ to F_j and go to Step (2).
5. Else increment j by 1.
6. If $j \leq maxneighbor$ go to Step (3).
7. Else when $j \succ maxneighbor$, If $M(F_{current})$ is less than $\min M(F)$ set minimum to this value and retain $F_{current}$ as the best set of medoids obtained so far.
8. Increment i by 1. If $i \succ numlocal$ terminate and output the best set of medoids. Otherwise, go to Step(2).

Figure 4. CLARANS Algorithm

After selecting an arbitrary node in the graph as the current node, one of its neighbors is selected randomly. If the value of the cost function for the selected node is less than the current node, the search continues by setting the selected node as the current node. Otherwise, the process of randomly selecting a neighbor for cost comparison is repeated for a pre-determined maximum number of times (*maxneighbor*). If a neighbor with less cost couldn't be achieved after this amount of comparisons the algorithm will terminate giving the current node as a local minimum. *Numlocal* is another input parameter of this algorithm. To achieve a better effectiveness CLARANS repeats the whole process for *numlocal* times by starting from a different initial node and returns the node with the minimum cost in which the algorithm terminated.

The performance of CLARANS depends on the two input parameters namely *numlocal* and *maxneighbor*. With values of *maxneighbor* nearer to $k(n-k)$ the quality of the results will become closer to that of PAM. Experiments have shown that %1.25 of n as a value for *maxneighbor* leads to clustering results with acceptable quality in a reasonable time [14], [21]. Also, it has been experimentally realized that the values 1 or 2 are appropriate for *numlocal* [14], [21] considering the trade off in achieving better results by paying more computational cost.

Generally, using the medoid based approach and the applicability of CLARANS for large size applications such as spatial data mining are its major advantages over the other clustering methods.

E. Fac-means

K-means is one of the most popular clustering algorithms that has been employed in many applications. In spite of its popularity for general purpose clustering, this algorithm suffers from some shortcomings. Fist, the performance of the algorithm degrades when applied on huge size data sets. Second, the number of clusters K has to be initially given to the algorithm [15]. These two shortcomings are of high concern in the problem of facility establishment optimization. Performance is a major concern as we focus on the large sizes of this problem with thousands of customer points. Also, the business objective is to minimize the covering cost of the customers formulated in Equation 6 regardless of the number of new facilities. By restricting the search to find the location of an exact given count of facilities the algorithm will not be able to optimize the objective criterion. The reason is that we may achieve better values of the objective function by increasing or decreasing the value of K. In other words, the cost of covering the customers is directly depended on the number of facilities that are to be established. On top of that, the user has no idea about the optimal number of facilities in advance. Consequently, it will be much more effective and realistic if the algorithm searches for the optimal number of facilities simultaneously. The most simple and naïve way to find the optimal count of facilities would be to run the K-means algorithm with many different values of K and select the solution which results in the least customer covering cost. However, the computational cost of running K-means for a specific value of K is so high that makes this algorithm inefficient for large data sets. Hence, repeating the execution of this algorithm for every possible value of K will be impractical.

We propose Fac-means for efficient integration of the search for the optimal number of facilities along with their optimal location. This algorithm employs the trade-off between the two values in the right side of Equation 6 in an straight forward manner. Fac-means starts by finding the best locations for the establishment of an initial number of facilities and continues by adding new facilities in the most appropriate location if their establishment decreases the customers covering cost. In each iteration, the algorithm tries to further improve the locations by applying K-means. This way, we will be able to search for the optimal number of the facilities beside their optimal locations in a parallel mode. Fac-means is actually a variation of the X-means algorithm proposed in [15]. However, in X-means there is no natural and clear notion for adding the value of K. The algorithm uses the Bayesian Information Criterion as an evaluation criterion to split a cluster [15]. But, in Fac-means we can easily make the most of the cost trade-offs in the covering cost. In fact, the criterion used in Fac-means for adding a new facility is a simple but effective one.

0. Set K to the *minK* and arbitrary initialize $F_{0..k-1}$.
1. Run K-means with K clusters and centroids in F.
2. Run 2-means for each cluster found in Step (1).
3. Replace the facilities for each cluster according to Equation 11 and set the value of *newK* to the count of facilities.
4. If *newK* is equal to the previous value of K terminate. Else, set K to *newK* and pass the updated set F to Step (1)

Figure 5. Fac-means Algorithm

The pseudo-code of the algorithm is given in Figure 5. In the first step of Fac-means, K-means is run on the whole data set to find the best locations for K=*minK* new facilities. The parameter *minK* is an input lower bound for the number of facilities. At this point, there is a fixed total distance from each facility F_j to its customers denoted as $M(F_j)$ or simply M_j where $j=\{0,1,...,K\}$ and

$$M(F_j)=M_j=\sum_{i=0}^{n_j-1} w_i d(s_i,F_j). \quad (8)$$

Where $s_i \in S_j$ and $i=\{0,...,n_j-1\}$. S_j is the set of the customer points assigned to the facility F_j and n_j is the count of these points. As a result there will be a certain logistics cost $LCost(F_j)$ or $LCost_j$ incurred by each facility F_j using Equation 8. Hence, the total cost of covering the customers in S_j denoted as $CCost_j$ will be obtained as follows:

$$CCost_j = LCost_j + ECost_j. \quad (9)$$

Where $ECost_j$ is the establishment cost of facility F_j which is equal to e_j. In the next step, a new search is performed on each cluster found in the previous step. By temporarily excluding F_j, its customer points in S_j will be independently clustered by K-means to two clusters (i.e. *K*=2). By independently, we mean that the 2-means algorithm will only consider the customer points in S_j and not the points assigned to other facilities in F. Assuming that the 2-means algorithm yields the two new facilities F_a and F_b, the new cost of covering the customers in S_j will become:

$$CCost'_j = LCost_a + ECost_a + LCost_b + ECost_b . \quad (10)$$

Now, we have to decide whether to split the cluster of F_j by replacing F_j with the two new facilities F_a and F_b. For this purpose, we use the simple final criterion i.e. the customer covering cost:

$$CCost'_j \prec CCost_j \Rightarrow \text{Replace } F_j \text{ with } F_a \text{ and } F_b . \quad (11)$$

By applying the above process on every cluster, the count of facilities will become *newK* and the new clusters are passed to the next iteration. In the beginning of the next iteration, the K-means algorithm is globally run for all the points in *S* with the value of *K* being equal to *newK*. The algorithm terminates when the number of facilities does not change in an iteration; which means that the establishment of additional facilities will not decrease the covering cost any more. At this point the minimum covering cost has been obtained by finding the optimal number of facilities along with their locations.

IV. Using Clustering For Facility Location Optimization

Having explained some major previously proposed clustering algorithms and proposing our new algorithm, we will investigate the application of these algorithms into the problem of optimal facility establishment. This problem has some particular characteristics that make such investigation important.

Logistics Cost. Considering the final objective of the customer covering cost, the logistics cost has the major contribution to this objective. This is due to the fact that the number of customer points is thousands of times larger than the number of facilities. Moreover, the logistics cost is incurred in every field-based service lifecycle for a long period of time, perhaps for years. In addition, the total distance to the customer points provides an insight to the average amount of time between receiving a request and its fulfillment. Hence, lower values of logistics distance indicate faster response to the customer requests. Therefore, the ability of the clustering solution to find facility locations with lower sum of logistics cost is a key feature. This value is actually equivalent to the clustering error as defined in Equation 1.

Efficiency and Scalability. Facing the huge size of today's spatial databases in field-based services, the efficiency and scalability of the clustering algorithms in terms of execution time is a significant concern. Indeed, some algorithms that can accomplish better results may be non practical for large data sets because of their inefficiency. Hence, we should consider the trade off between spending more computational cost and achieving less customers covering cost when comparing the algorithms for facility location optimization.

Number of Facilities. Another important aspect is the capability of the algorithm to effectively find the best possible number of facilities in addition to their locations. In typical data clustering, there is no clear interpretation about the best number of clusters. Many criteria have been proposed for this purpose such as silhouette width [13] and various information theoric measures [15]. Here, the clustering results with different number of facilities are compared in terms of the accomplished total cost of covering the customer points formulated in $CCost(F)$. This final objective is much more significant in the current application compared to the conventional applications of data clustering algorithms in which there is no direct concern of financial resources. Thus, the ability of the clustering algorithm to effectively optimize the final cost by searching for the optimal number of facilities is a major feature.

V. Experimental Study

A. Data Sets and Parameters

We use four different data sets to experimentally study the performance of the discussed clustering algorithms. The first data set consists of 1912 spatial points of bank branches in the city of Tehran. The output facilities can be used for applications such as money delivery to the branches and police centers. This data set is shown in Figure 6.

The other three data sets are synthetically generated with 3000, 4000 and 5000 spatial points in each of them. The points in these data sets are generated in the same area of the first data set which has the ranges of [0, 30000] and [0, 25000] for X and Y, respectively. The data generator algorithm inputs the total number of points (*n*), the number of clusters (K'), and the ranges of number of member points and the radius of each cluster. After selecting a random center point for a cluster from the area, number of member points and the radius are selected from the provided ranges at random. The ranges for the number of points and the radius are set to [35, 125] and [500, 4000], respectively. To simulate the real world situations, the generated clusters of this model may have intersections and $\%\varphi$ of the points will be generated randomly to simulate noise and outliers. The value of φ is set to 2.

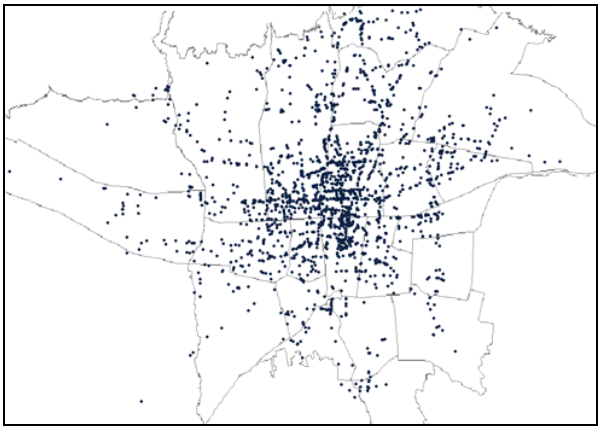

Figure 6. The spatial points of branches of eight financial banks in the city of Tehran.

Appropriate values for the parameters of the algorithms are chosen for each algorithm based on some experiments and also the results reported in previous works. Considering K-means, the algorithm is repeated five times with randomly selected intial points. This also holds for the application of K-means in Fac-means. The selected sample size for CLARA is $2K + 40$ and five samplings are applied [14], [21]. The values of *numlocal* and *maxneighbor* in CLARANS are set to 1 and $1.25n$, respectively [14], [21]. Experiments are run on P4 3GHz CPU with 1G RAM.

B. Logistic Cost

The main objective in facility establishment is to minimize the logistics cost which is equivalent the clustering error in here. In Table 1 the averaged logistics cost incurred by the output set of facilities in each algorithm is given. The values are actually the sum of the distances (in kilometers) from the output facilities to the customers' spatial location. The algorithms are run by setting the number of facilities (K) for each data set to the value that Fac-means found for that data set. This value is shown in the second column of Table. In Fac-means and in all other experiments, the establishment cost for any facility is equal to the logistics cost of 100 kilometers.

Table 1. The logistics cost averaged over 10 independent runs for the four data sets.

Data Set	K	K-means	CLARA	CLARANS	Fac-means
BankPoints	16	3223 ± 14	3922 ± 62	3250 ± 25	3211 ± 6
Ds1	25	3678 ± 71	4374 ± 134	3486 ± 18	3464 ± 10
Ds2	31	4398 ± 72	5184 ± 219	4081 ± 16	4069 ± 14
Ds3	37	4142 ± 138	4966 ± 117	3576 ± 17	3568 ± 7

As shown in the Table 1 Fac-means is the best performing algorithm in terms of the clustering error which is equal to the logistics cost here.

Fac-means has finds the best set of facilities for the BankPoints data set. In addition, Fac-means is significantly more efficient and can give high quality results in a reasonable time. CLARA looses its effectiveness for large data sets as it applies a sampling approach. On the other side CLARANS leads to facilities with satisfactory logistics cost compared to Fac-means.

C. Efficiency and Scalability

The major challenge in the application of clustering algorithms for facility establishment optimization is the efficiency and scalability of these algorithms when applied on large number of customer spatial points. Figure 7 depicts the scalability of the algorithms when increasing the size of data set.

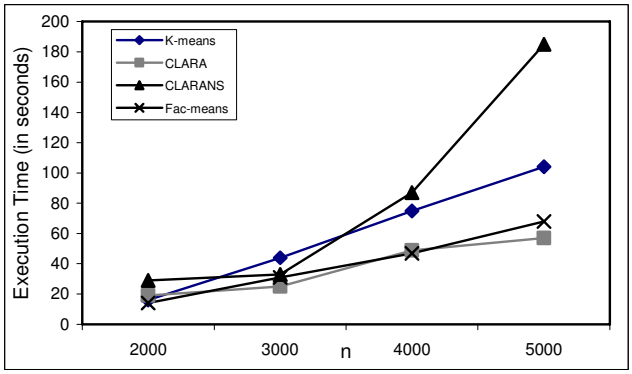

Figure 7. Comparison of scalability with regards to n

As shown in Figure 7, CLARA has the best scalability. This is due to the fact that CLARA applies PAM on a sample data set whose size is independent of n. Fac-means and CLARANS also have good efficiency. Actually, Fac-means has a better efficiency compared to K-means and CLARANS.

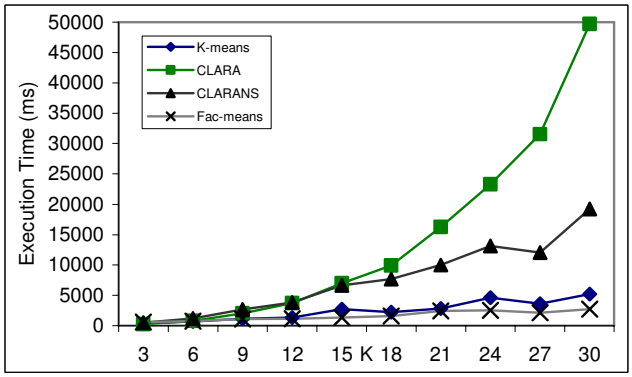

Figure 8. Comparison of scalability with regards to K.

In Figure 8, the scalability of the algorithms are compared when increasing the number of clusters in the BankPoints data set. In contrast to Figure 6, CLARA has the worst scalability in terms of K. Fac-means has the best efficiency while the performance of CLARANS is also satisfactory. Consequently, the superior performance of Fac-means makes it a suitable solution for the optimal facility establishment problem even in large data sets.

D. Number of Facilities

To investigate the capability of the algorithms to find the optimal number of facilities we compare the results of Fac-means with that of CLARANS which is the second best

performing algorithm. The comparison will be based on the final objective which is the total covering cost. In each run for CLARANS, the algorithm is run with different values of K and the value of K which has the best cost is chosen. To compress the search space of K in CLARANS, we use the rang of $[K_{Fac-means} - 3, K_{Fac-means} + 3]$ where $K_{Fac-means}$ is the number of facilities that are mostly found by 30 independent runs of Fac-means. Table 2 summarizes the results of this experiment for the four different data sets.

Table 2. Comparison of final objective and number of facilities found.

Data set	Algorithm	Found K	Total Covering Cost	Execution Time (ms)
BankPoints	CLARANS	16.8 ± 1.2	4808 ± 28	37066
	Fac-means	16.3 ± 0.6	4785 ± 3	2745
Ds1	CLARANS	26.3 ± 1.0	6014 ± 2	206642
	Fac-means	25.0 ± 0.1	5983 ± 0.6	8781
Ds2	CLARANS	29.6 ± 0.8	7195 ± 1	518212
	Fac-means	30.8 ± 0.5	7184 ± 2	7741
Ds3	CLARANS	35.9 ± 0.5	7295 ± 31	1077458
	Fac-means	36.4 ± 0.7	7277 ± 9	9732

The final covering cost averaged in 30 independent runs of each algorithm shows that Fac-means is able to find the best number of facilities along with their locations. In addition Fac-means has much less computational cost compared to CLARANS. Hence, Fac-means is the most effective solution for the problem of optimal facility establishment and has the important advantage of scalability for large size data sets.

VI. Conclusions and Future Works

Selecting optimal locations for the establishment of new facilities is a critical decision in organizations that provide field-based services. With the increasing size of this problem in today's applications, efficiency and scalability of the solution has become a major challenge. In this paper, we have studied the application of efficient and effective spatial clustering algorithms for discovering the optimal locations. We have proposed a new algorithm that is capable of finding the optimal number of facilities along with their locations. The experiments provided a comparative study between the discussed algorithms and confirmed that Fac-means has the best performance and scalability among the algorithms. In our future studies, we plan to study the application and tuning of other efficient spatial clustering algorithms. We will also investigate other business factors in the clustering algorithms such as variable facility establishment costs, covering radius limit or the maximum/minimum number of customers for each facility.

References

[1] S. Agnihothri, N. Sivasubramaniam, and D. Simmons, "Leveraging technology to improve field service," *International Journal of Service Industry Management*, vol.13, no.1, pp. 47-68, 2002.

[2] O. Berman and D. Krass, "The generalized maximal covering location problem," *Computers & Operations Research*, vol. 29, no.6, pp. 563–581, 2002.

[3] M. Ester, A. Frommelt, H.P. Kriegel, and J. Sander, "Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support," *International Journal of Data Mining and Knowledge Discovery*, vol. 4, no.2/3, pp.193-217, 2000.

[4] M. Ester, H. P. Kriegel, S. Sander, and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise," In *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pages 226–231.

[5] V. Estivill-Castro and M.E. Houle,, "Robust Distance-Based Clustering with Applications to Spatial Data Mining," *Algorithmica* Vol. 30, no. 2, pp.216-242. 2001.

[6] R.D. Galvao, Gonzalo Acosta Espejo L., B. Boffey, "A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem," *European Journal of Operational Research*, vol. 124, no. 2, pp.377–89, 2000.

[7] Y. Huang, S. Shekhar, and H. Xiong, "Discovering Spatial Co-location Patterns from Spatial Datasets: A General Approach.," *IEEE Transactions on Knowledge and Data Eng.* vol.17, no.12 , pp. 1472-1485, 2004.

[8] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York, 1990.

[9] R.K. Klimberg and F.C. Van Bennekom, "Aggregate planning models for field service delivery," *International Journal of Location Science*, Vol. 5, no. 3, pp. 181-195, 1997.

[10] K. Koperski and J. Han, "Discovery of Spatial Association Rules in Geographic Information Databases," In *Proc. 4th Int. Symp. on Large Spatial Databases*, 1995, pp. 47-66.

[11] K. Koperski, J. Han, and N. Stefanovic, "An Efficient Two-step Method for Classification of Spatial Data," In *Proc. International Symp. On Spatial Data Handling* , 1998, pp.320-328.

[12] A.T. Murray and R.L. Church, "Applying simulated annealing to location-planning models," *Journal of Heuristics*, Vol. 2. pp. 31-53, 1996.

[13] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," In *Proc 20th Conference on Very Large Data Bases (VLDB)*, 1994, pp. 144–155.

[14] R. Ng and J. Han, "CLARANS: A Method for Clustering Objects for Spatial Data Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp.1003–1017, 2005.

[15] D. Pelleg and A. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," In *Proc 17th International Conf. on Machine Learning*, 2000, pp. 727–734.

[16] K. E. Rosing, C. S. ReVelle, and H. Rosing-Voyelaar, "The p-median and its linear programming relaxation: An approach to large problems," *Journal of the Operational Research Society*, vol.30 pp. 815–823, 1979.

[17] S. Shekhar, P. Schrater, W. R. Vatsavai, W. Wu and S. Chawla, "Spatial Contextual Classification and Prediction Models for Mining Geospatial Data," *IEEE Transactions on Multimedia*, vol. 2, no.4, pp.174-188, 2002.

[18] A. K. H. Tung, R. T. Ng , L. V. S. Laksmanan, and J. Han, "Constraint-based clustering in large databases," In *Proc. Intl. Conf. on Database Theory*, 2001, pp.405-504.

[19] L. Wang, K. Xie, T. Chen, and X. Ma, "Efficient Discovery of Multilevel Spatial Association Rules Using Partitions," *Information and Software Technology*, Vol. 47, no. 13, pp.829-840, 2005.

[20] W.Wang, J. Yang, and R. Muntz. "STING: A statistical information grid approach to spatial data mining," In *Proc. 23rd International Conference on Very Large Data Bases (VLDB)*, 1997, pp.186–195.

[21] C.-P.Wei, Y.-H Lee, and C.-M. Hsu, "Empirical comparison of fast partitioning-based clustering algorithms for large data sets," *Expert Systems with Applications*, Vol. 24, No.4, pp. 351-363, 2003.

[22] A. Zarnani, M. Rahgozar, C. Lucas and A. Memariani, "AntTrend: Stigmergetic Discovery of Spatial Trends," In *Proc. 16th Int. Symp. On Methodologies for Intelligent Systems*, 2006, pp.91-100.

[23] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," In *Proc. International Conference on Management of Data*, 1996, pp.103–114.