J Ind Eng Int (2014) 10:54
DOI 10.1007/s40092-014-0054-x

TECHNICAL ARTICLE

New spatial clustering-based models for optimal urban facility location considering geographical obstacles

Maryam Javadi · Jamal Shahrabi

Received: 25 February 2013 / Accepted: 6 March 2014 / Published online: 25 March 2014

Abstract The problems of facility location and the allocation of demand points to facilities are crucial research issues in spatial data analysis and urban planning. It is very important for an organization or governments to best locate its resources and facilities and efficiently manage resources to ensure that all demand points are covered and all the needs are met. Most of the recent studies, which focused on solving facility location problems by performing spatial clustering, have used the Euclidean distance between two points as the dissimilarity function. Natural obstacles, such as mountains and rivers, can have drastic impacts on the distance that needs to be traveled between two geographical locations. While calculating the distance between various supply chain entities (including facilities and demand points), it is necessary to take such obstacles into account to obtain better and more realistic results regarding location-allocation. In this article, new models were presented for location of urban facilities while considering geographical obstacles at the same time. In these models, three new distance functions were proposed. The first function was based on the analysis of shortest path in linear network, which was called SPD function. The other two functions, namely PD and P2D, were based on the algorithms that deal with robot geometry and route-based robot navigation in the presence of obstacles. The models were implemented in ArcGIS Desktop 9.2 software using the visual basic programming language. These models were evaluated using synthetic and real data sets. The overall performance was evaluated based on the sum of distance from demand points to their corresponding facilities. Because of the distance between the demand points and facilities becoming more realistic in the proposed functions, results indicated desired quality of the proposed models in terms of quality of allocating points to centers and logistic cost. Obtained results show promising improvements of the allocation, the logistics costs and the response time. It can also be inferred from this study that the P2D-based model and the SPD-based model yield similar results in terms of the facility location and the demand allocation. It is noted that the P2D-based model showed better execution time than the SPD-based model. Considering logistic costs, facility location and response time, the P2D-based model was appropriate choice for urban facility location problem considering the geographical obstacles.

Keywords Facility location · Spatial clustering · Geographical obstacles · Distance function · Geographic information system

Introduction and literature review

The facility location problem is an important research topic in spatial data analysis which aims to investigate the challenging problems of matching the supply and demand by exploiting sets of objectives and constraints (Koperski et al. 2001).

The objective is to determine a set of locations for the facilities in such a way that the total supply and assignment cost is minimized. For example, city planners are interested in the best feasible way of allocating facilities (hospitals,

M. Javadi (✉)
Isfahan Municipality Information & Communication Technology, Department of Software Engineering, University of Payam Noor, Tehran, Iran
e-mail: m.javadi@isfahan.ir; maryamjvd@gmail.com

J. Shahrabi
Industrial Engineering Department, Amirkabir University of Technology, Tehran, Iran

fire stations, etc.) to new residence areas. The decision is made according to the local population and constraints.

During the past 30 years, geographical information system (GIS) has evolved into a considerable research and application area. GIS is playing a significant role in location model development and application due to the ability of supporting a wide range of spatial queries and analyses. GIS can constructively support the decisions involving facility establishment through effective spatial analysis (Church 2002).

To be more specific, the demand points are modeled in the geographic space along with other information such as the distances, candidate locations for facility establishment and regions with different location costs. The objective is to locate facilities in the city so as to minimize the overall cost incurred to satisfy the demand points. This cost is usually measured by the sum of distances from a demand point to its nearest facility. This optimization problem is well known to the operations research community as the discrete *p*-median or the facility location problem (Berman and Krass 2002).

Integer programming, Lagrangian relaxation and other heuristic methods are common approaches to deal with this problem (Galvao et al. 2000). However, the *p*-median problem is a NP-hard problem. The scalability of such approaches is an important issue due to the large databases encountered in today's applications, which involve a large number of points, i.e., thousands or more (Tung et al. 2001; Estivill-Castro and Houle 2001).

Given a data set, clustering detects specific number of clusters that are internally homogeneous and members of different clusters have maximum dissimilarity. Many algorithms have been designed to perform spatial clustering with respect to spatial dimensions of the objects. In many geographical knowledge discovery tasks, it is preferred to apply spatial cluster analysis because of its ability to extract structures directly from the data without employing any priori known spatial concept hierarchies (Estivill-Castro and Houle 2001; Ng and Han 1994). As a matter of fact, a spatial clustering algorithm seeks a specified number of representative points in the spatial space. These points lead to the clusters and their members.

To evaluate the quality of a set of representatives, it is common to use the sum of distances from each point to its nearest representative (Kaufman and Rousseeuw 1990). The same objective (in the facility establishment problem) is faced when demand points and facilities are considered as spatial points and representative points, respectively. Here, the distance refers to the spatial distance such as Euclidean distance.

Consequently, we discuss that the spatial clustering algorithms can be modified to solve facility establishment problem effectively. Some researchers have customized clustering algorithms to locate capacitated facilities (Liao and Guo 2008; Geetha et al. 2009; Kaveh et al. 2010). Several methods have been proposed for the static and transportation facility location problem (Wei and Xin 2010). Various clustering algorithms have been studied and compared in the context of solving facility establishment problem (Zarnani et al. 2007). Most of the researchers use the Euclidean distance as the dissimilarity function.

A city may contain obstacles such as rivers and mountains. These obstacles can have drastic effects on the distances between demand points and facilities.

COD-CLARANS (clustering with obstructed distance based on CLARANS) (Tung et al. 2001) was the first clustering algorithm that takes into consideration the presence of obstacle entities. Although COD-CLARANS generates good clustering results, there are several major problems with this algorithm. Since COD-CLARANS is an extension of CLARANS algorithm, it suffers from similar drawbacks as CLARANS. In addition, COD-CLARANS cannot handle outliers. Also the overall efficiency of the algorithm is very low because the model used in preprocessing for determining visibility and building the spatial join index would need to be significantly changed. Third, if the dataset has varying densities, COD-CLARANS's micro-clustering approach may not be suitable for the sparse clusters.

In this paper, we propose new models for the facility location problem while considering geographical obstacles. To compute distances between demand points and facilities in the presence of obstacles, three new distance functions are introduced which are based on the DIJKSTRA's shortest path algorithm, Bug1 and Bug2 algorithms for robot navigation (Choset et al. 2007).

The proposed distance functions are compared in terms of facility location, allocation of demand points to facilities, logistics cost and response time. We evaluate the models on synthetic and real data sets. The real data set is based on the regional maps of Isfahan city. In this paper, the center of each urban population area is considered as a weighted demand point. The weight of each center is equal to its population.

The rest of the paper is organized as follows. In "Problem definition and proposed models", the problem is defined in detail and the new models are presented. We evaluate the proposed models on synthetic and real data sets in section "Evaluate the proposed models". Finally, section "Conclusion" concludes the paper.

Problem definition and proposed models

The problem focused on in this study was to find the best locations for the establishment of the facilities so that it could be covered the customers' demands with the least logistics cost. The logistics cost was measured by the distance that needs to be traveled from a facility to customer demand points.

Problem definition

The significance of each demand point was determined by the weight assigned to it. The logistics cost was measured by the sum of distances (in km) between the facilities and demand points. In fact, there is a spatial point r_i for the location of each demand point, where the set of all demand points is $R = \{r_0, r_1, \ldots, r_{n-1}\}$ and n is the total number of demand points. It was considered a spatial point f_i for each facility. The following criteria must be optimized (minimized) to obtain improved logistics performance (Estivill-Castro and Houle 2001):

$$M(F) = \sum_{i=0}^{n-1} w_i \times d(r_i, \mathrm{fac}[r_i, F]) \tag{1}$$

where $F = \{f_0, f_1, \ldots, f_{K-1}\}$ is the set of K facilities in the two-dimensional space R^2. w_i is an optional factor that shows the significance of a demand point r_i.

$d(p, q)$ is the distance between two points. $d(p, q)$ is the Euclidean spatial distance by Eq. (2) in most researches.

$$d(p, q) = \left(|p_x - q_x|^2 + |p_y - q_y|^2\right)^{\frac{1}{2}} \tag{2}$$

Natural obstacles, such as mountains and rivers, can have drastic impacts on the distance that needs to be traveled between two geographical locations. While calculating the distance between various supply chain entities (including facilities and demand points), it is necessary to take such obstacles into account to obtain better and more realistic results regarding location-allocation. In this paper, some modified form of the Euclidean distance was used as a dissimilarity function.

In this study, three new distance functions have been proposed to consider these obstacles. These functions were defined in section "Proposed distance functions".

$\mathrm{fac}[r_i, F]$ is the nearest facility (in F) to a demand point r_i, which can be defined by Eq. (3) as follows:

$$d(r_i, \mathrm{fac}[r_i, F]) = \min_{j \in \{0, \ldots, (K-1)\}} d(r_i, f_j) \tag{3}$$

The overall distance from the current set of facilities F to the satisfied demand points is represented by Eq. (1). Based on this value considering and a basic cost for a specific amount of logistics distance, the logistics cost can be obtained. Thus, the value of LogCost (F) is computed which is the total logistics cost with respect to the set of facilities F,

$$\mathrm{LogCost}(F) = p.M(F)/d \tag{4}$$

where p is the corresponding financial cost of the logistics distance d.

In brief, a system was studied with two types of inputs that were spatial points representing demand locations and models to determine the total logistics cost. The output of the system was the optimal location of facilities to cover the demand points with the minimum cost.

In this paper, three new distance functions were proposed to take geographical obstacles into consideration. The first method was based on DIJKSTRA shortest path algorithm (Zhang et al. 2005).

The other methods have been developed such as robot navigation in the presence of obstacles (Choset et al. 2007). The new models were incorporated such as Bug1 and Bug2 algorithms into each method.

In following, Bug1, Bug2 algorithms are introduced.

Bug1 algorithm

In this algorithm, a robot begins movement at the start and proceeds towards the goal. It arrives at either the goal or an obstacle (hit point). Once an obstacle is encountered, the robot will completely circumnavigate the obstacle before proceeding forward from the point on the perimeter that has the shortest distance to the goal. This point is called a leave point. From leave point, the robot continues to move directly toward the goal again. Perhaps the most straightforward path planning approach is to move toward the goal, unless an obstacle is encountered, in which case, circumnavigate the obstacle until motion toward the goal is once again allowable. Essentially, the Bug1 algorithm formalizes the "common sense" idea of moving toward the goal and going around obstacles (Choset et al. 2007).

Robot path when encountering the obstacle in Bug1 algorithm is shown in Fig. 1.

Bug2 algorithm

Like its Bug1 sibling, the Bug2 algorithm exhibits two behaviors:

Motion-to-goal and boundary following. During motion-to-goal, the robot moves toward the goal on the m-line; however, in Bug2 the m-line connects start point and goal point, and thus remains fixed. The boundary-following behavior is invoked if the robot encounters an obstacle, but this behavior is different from that of Bug1. The robot circumnavigates the obstacle until it reaches a new point on the m-line closer to the goal than the initial point of contact with the obstacle, for Bug2. Then, the robot proceeds toward the goal, repeating this process until it encounters

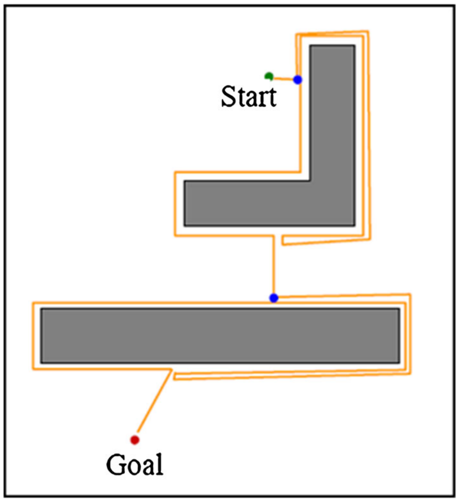


Fig. 1 Robot path in Bug1 algorithm

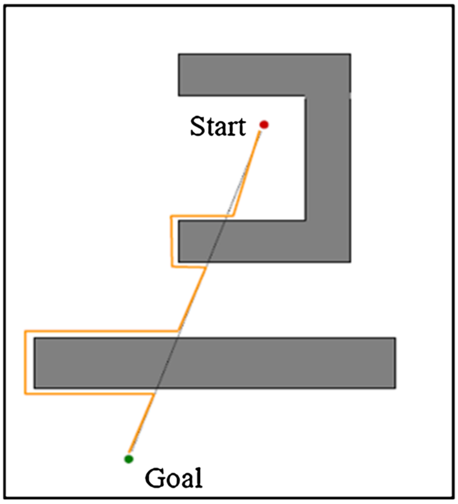


Fig. 2 Robot path in Bug2 algorithm

an object. If the robot re-encounters the original departure point from the *m*-line, then the robot concludes there is no path to the goal.

In fact, the start and the goal are connected with an imaginary straight line (i.e., *M*-line) and the robot follows this line, in this algorithm. If the robot hits an obstacle it will circumnavigate the obstacle until it reaches the *M*-line. Then, the robot starts moving towards the goal. Here, the robot does not have to entirely circumnavigate the obstacles (Choset et al. 2007).

Robot path when encountering the obstacle in Bug2 algorithm is shown in Fig. 2.

Spatial clustering methods for facility location problem

Spatial data mining has become a popular and powerful means for complex analysis of huge amounts of geo-referenced data. Spatial data mining is defined as the automatic process of discovering interesting and implicit knowledge from large amounts of spatial data. 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.

Spatial clustering is known to be one of the main spatial data mining tasks. Many algorithms have been developed for the task of spatial clustering focusing on the spatial dimensions of the objects.

Clustering is a process that divides a set of objects into several groups (clusters) such that the similarity between the members of each cluster is maximized. In general data clustering, the formulation of the problem is the same as the formulation provided in the “Problem definition” section with the exception that the data points have *m* dimensions (Estivill-Castro and Houle 2001; Kaufman and Rousseeuw 1990). Spatial clustering methods on the other hand focus on the points with two dimensions and incorporate the proximity information of the spatial points.

In many geographical knowledge discovery tasks, the attractiveness of spatial cluster analysis is its ability to find structures directly from the data without relying on any a priori known spatial concept hierarchies. (Estivill-Castro and Houle 2001; Ng and Han 1994) Actually, a spatial clustering algorithm searches for a specified number of representative points in the spatial space. These points determine the clusters and their members.

Clustering algorithms can be generally categorized into partitioning methods (Kaufman and Rousseeuw 1990; Ng and Han 1994), hierarchical methods (Guha and Rastogi 1998), density-based methods (Ester et al. 1996) and grid-based methods (Wang et al. 1997). In this work, we concentrate on the partitioning methods because of the following motivations:

- The main advantage of hierarchical methods is their ability to extract a hierarchy of clusters (dendrogram) which is not helpful in our target problem. Also the hierarchical methods suffer from poor scalability with the increasing number of points. In fact the computational cost incurred is *O* (*n*2) for *n* data points.
- The main advantage of density-based methods is their ability to find elongated and non-convex clusters. This is a valuable capability in spatial data mining applications. Nonetheless, this is not useful in the problem of finding the best locations for facilities. Also, density-based approaches are robust towards noise and outliers. However, based on our definition of the problem all of the demand points have to be covered and served by the facilities. Hence, the notion of outlier is actually of no importance in this context. The interested reader can

refer to for solutions that do not necessarily cover all of the customer points.

- Grid-based approaches also suffer from some shortcomings as a possible solution to our problem. First, the performance of these algorithms relies on many user-given parameters such as the granularity of the lowest level of the grid structure and data distribution. Second, the resulting clusters are bounded horizontally or vertically, but never diagonally. Finally, the same case about noise tolerance in density approaches holds in grid-based methods.
- Considering that the overall distance must be minimized, the most appropriate algorithms are partitioning methods.
- In addition, embedding the objective functions for optimal facility establishment in partitioning methods is much less complicated compared to the other methods as will be shown.

The above explanations are the main motivations for using partitioned-based approaches in most of the spatial clustering methods. So, we have used partitioning method for optimal facility location problem.

K-means algorithm uses the average of cluster objects as a center. The initial centers can selected arbitrarily and at each point of Euclidean space. But, in *k*-medoid algorithms, the initial centers must be selected from the demand points. These algorithms cannot handle outliers. Given that all points must be considered in our problem, we have selected *K*-means algorithm.

K-means algorithm

K-means is one of the most basic and widely used partitioning based clustering algorithms due to its simplicity and ease of use (Kaufman and Rousseeuw 1990). In this algorithm, the data points are partitioned into *K* different subsets by assigning each point to the nearest center [Eq. (3)]. The number of desired clusters *K* and the set of points *S* are provided as the inputs.

K-means is a deterministic approach that heuristically solves the optimization problem of Eq. (1) (known as clustering error) by finding a local minimum.

The *K*-means algorithm consists of three main steps: (1) randomly choosing *k* cluster centers within the data space; (2) assigning each data item to the closest cluster center; and (3) recalculating the cluster centers using the points assigned to each cluster. Steps 2 and 3 are then repeated until the result converges. In pseudocode shown in Fig. 3, step 3 is separated into three steps.

The steps of the *K*-means clustering algorithm were shown in Fig. 3.

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 presented better results in terms of clustering error and robustness (Pena et al. 1999). Many other variants of *K*-means have also been developed. The same approach was used for finding the optimal locations of facilities in this study. It is noted that many other variants of *K*-means have also been developed.

Likas et al. (2003) has been proposed a modified version of *K*-means with the improved convergence properties and independence from initialization. However, the computational cost of this algorithm is a big concern as it runs *K*-means once for each node and each value of $k = 2, \ldots, K$.

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.

Proposed models with obstacle consideration

K-means has appropriate computational efficiency as the 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 should be found, *t* is the number of iteration and *n* is the total number of customer request points (Anderberg 1973).

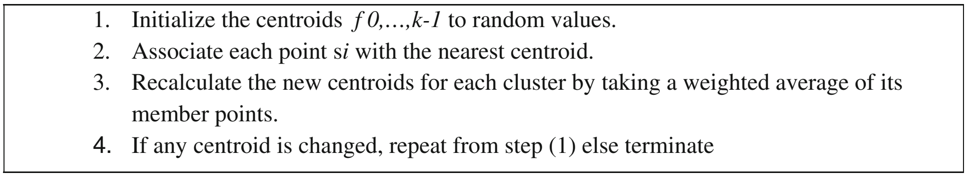
An obstacle is the physical object that obstructs the reachability among the data objects. Natural obstacles, such as mountains and rivers, can have drastic impacts on the distance. They need to be traveled between two geographical locations. While calculating the distance between various supply chain entities (including facilities and demand points), it is necessary to take such obstacles into account to obtain better and more realistic results regarding location-allocation.

The base algorithm is the *K*-means which is implemented via three different distance functions, in this model.

The simulated codes of the proposed algorithm are listed in Fig. 4.

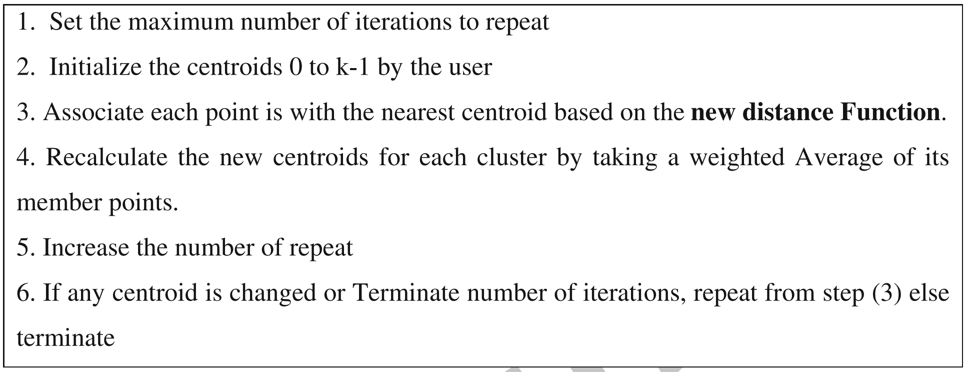
In the first step, the algorithm is begun by taking a maximum repeat value. The next step is crucial since the algorithm converges to the final centroids based on the initial values of the centers. In this step, some centers are selected by the user in the GIS map layer. It is noted that, each demand point is assigned to the nearest center

Fig. 3 K-means algorithm



1. Initialize the centroids *f 0,…,k-1* to random values.
2. Associate each point s*i* with the nearest centroid.
3. Recalculate the new centroids for each cluster by taking a weighted average of its member points.
4. If any centroid is changed, repeat from step (1) else terminate

Fig. 4 The proposed algorithm



1. Set the maximum number of iterations to repeat
2. Initialize the centroids 0 to k-1 by the user
3. Associate each point is with the nearest centroid based on the **new distance Function**.
4. Recalculate the new centroids for each cluster by taking a weighted Average of its member points.
5. Increase the number of repeat
6. If any centroid is changed or Terminate number of iterations, repeat from step (3) else terminate

according to the distance function in the next step. In the fourth step, new centers' locations are calculated according to average of the points that assigned to each center and one unit is added to the number of loop iterations, in the fifth step. The new centers are compared to their previous positions in the last step. In this model, if the deviation was less than the threshold value or the maximum number of iterations was reached, the algorithm would be terminated.

The proposed distance functions would be presented as follows.

Proposed distance functions

In this section, three new distance functions are defined. These functions were based on Euclidean distance function. These functions have three different behaviors in presence of obstacles.

In this section, the SPD, the P2D and the PD functions proposed for the consideration of obstacles in locating the facility are introduced.

The SPD distance function

In this section, the method has proposed that defined the obstructed distance between two points as the length of the shortest path connects them without crossing any obstacles. Three steps are prepared as below:

1. Drawing the line connected the start point to the target point.
2. If the line hits the obstacles, for each obstacle, the obtained line is drawn from the start and target points to obstacle vertices that do not interrupt the obstacle. Also the edges of the obstacle are drawn. Then the shortest path from the start to the target in this linear network is considered as the distance.
3. If the line do not hit the obstacle, the Euclidean distance of the two points would be used as output distance function.

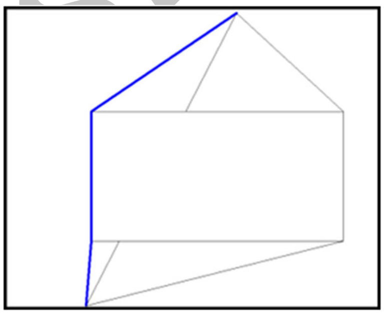


Fig. 5 Computation of distance in the SPD method

Computation of distance in the SPD method is shown in Fig. 5.

The P2D distance function

In this method, the direct distance between two points without crossing the obstacle plus distance on the smaller side of obstacle is considered as the function output to calculate the distance between two points. The steps have been clarified as follows:

1. Drawing the obtained line of two start and target points.
2. If the line hits the obstacle, the obstacle would be broken into some sections encountering the line.

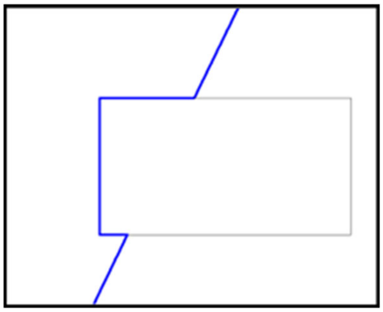


Fig. 6 Computation of distance in the P2D method

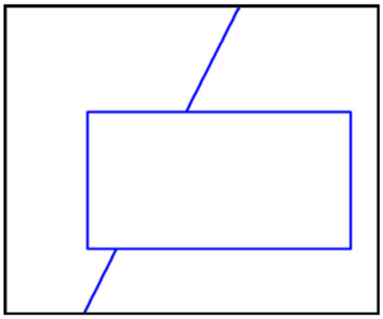


Fig. 7 Computation of distance in the PD method

3. The traveled distance on the smaller section of the obstacle plus the distance of start and target points from the hitting points, is considered as output of the distance function.
4. The Euclidean distance of the two points should be used as output distance function, if the line does not hit the obstacle.

Computation of distance in the P2D method is shown in Fig. 6.

The PD distance function

The direct distance between two points without crossing the obstacle plus obstacle perimeter is considered as the function to calculate the distance between two points in this method. Three steps are planed as follows:

1. Drawing the obtained line between two start and target points.
2. The obstacle perimeter plus the sum of distances of the start and target points from the hit points are considered as the distance function output, if the line hits the obstacle.
3. In case of the line and obstacle do not collide, the Euclidean distance of the two points would be considered. The computation of distance in the PD method is shown in Fig. 7.

Table 1 Characteristics of data sets

Row	Data set	*H* of points	No. of obstacles	No. of centers
1	Zone 1,5-Isf-river	22	11	2
2	Zone 10-Isf-river	117	11	4
3	Test-data-110	110	3	3
4	Test-data-200	200	9	5

Evaluate the proposed models

In this section, first the datasets are introduced and then the results of executing new models on this datasets are discussed and compared.

Characteristics of data sets

The Isfahan city is considered as sample in this study and all the programs are implemented for it. So, the Isfahan should be introduced. Isfahan is the capital of Isfahan Province, that is located about 340 km south of Tehran in Iran. It has a population of 1,583,609 and it is Iran's third largest city (after Tehran and Mashhad). The Isfahan metropolitan area had a population of 3,430,353 in the 2006 Census, the second most populous metropolitan area in Iran (after Tehran). The Zayanderood River starts in the Zagros Mountains, flows from west to east through the heart of Isfahan, and dries up in the Kavir desert.

Maps of Isfahan are selected from the real data sets according to the specifications of the city and crossing the Zayanderood River. So, each area of the river between two bridges is considered as an obstacle.

For synthetic data sets, some demand points are drawn randomly and the accidental weight is assigned to each point. Some areas are plotted as obstacles, in these datasets number of discrete obstacles have been drawn to consider obstacles in synthetic data sets for Test-data-110 data set (third row of Table 1) and the number of continuous and discrete obstacles have been drawn for Test-data-200 data set (fourth row of Table 1).

Characteristics of the synthetic data sets are given in the third and forth rows of Table 1 and characteristics of the real data sets are also presented in the first and second rows of Table 1.

In Table 1, the first row indicates the centers of population regions of 1 and 5 zones of the city of Isfahan and the second row represents the center of the population centers of 10 zones of the city. The third and fourth rows are produced with various numbers of demand points and obstacles.

Table 2 Comparison of proposed functions considering the logistic cost

Row	Data set	Result of SPD function	Result of P2D function	Result of PD function
1	Zone 1,5-Isf-River	8,616.050	8,616.050	8,625.07
2	Zone 10-Isf-River	17,255.777	17,252.171	17,267.48
3	Test-data-110	23,205.946	23,501.008	30,281.105
4	Test-data-200	65,536.871	69,165.398	97,926.996

Table 3 Comparison of proposed functions considering the execution time (in seconds)

Row	Data set	SPD function	P2D function	PD function	Euclidian function
1	Zone 1,5-Isf-River	35	17	9	4
2	Zone 10-Isf-River	79	36	16	7
3	Test-data-110	80	37	18	9
4	Test-Data-200	104	46	27	12

Experimental results

All the algorithms and the functions were implemented in the Arc GIS Desktop 9.2 with the visual basic programming language. These models were executed on system Intel Pentium 4, CPU 3.08 Ghz with 2 Gb of RAM.

The distance between two points was based on the Kilometer unit. Each 1 km was considered as the unit cost to convert this distance to cost.

The results obtained by executing these algorithms were studied on both the synthetic data sets (third and fourth rows of Table 1) and the map of population regions of the city of Isfahan (first and second rows of Table 1) with various cluster numbers and considerable results were obtained.

The results of the P2D method were very similar to results of computing the shortest path (function SPD) in terms of allocating points to clusters and location of cluster centers. Also, in terms of the logistic cost, the results were equal for the real datasets. Quantitative differences of the results were observed in the synthetic data sets. In the procedure of computing the shortest path (SPD), the execution time due to drawing lines, network formation and running DIJKSTRA algorithm were more than in P2D procedures, but the P2D function results in terms of the center's location, allocating demand points to centers and the logistic cost were similar to it.

A comparison of running algorithms on data sets of Table 1 was presented considering the logistic costs in Table 2.

As shown in Table 2, the logistic cost using both the SPD and P2D distance functions is very similar in the real data sets. It is noted that the difference in synthetic datasets results is low.

The two river sides were separately clustered considering the Zayanderood River as a widespread obstacle. In fact, one center was allocated to each side of the river in the first row of Table 1 data set and two centers were allocated to each side of the river in the second row of the same table data set. In this case, the logistic cost, equal to the logistic cost of clusters on both sides of the river, is indicated separately which in addition to spending more resources, required determining some centers for each side of the river, which was done manually and not optimized. In this case, the logistic cost of locating a center at each side of the river (first row of Table 1) was equal to 8,625,075.22 = 8,714.44 + 4,316,360.78. This is more than the two SPD and P2D methods.

The algorithm execution time (in seconds) with four distance functions on four data sets (Table 1) by four facilities is compared in Table 3.

The execution time using the P2D distance function was less than SPD function as shown in Table 3. Also the execution time using the PD distance function was less than P2D distance function. Euclidean distance function had less run time than the others. But according to minimize of logistics costs in Eq. (1), the P2D and SPD functions were the best. On the other hand, the execution time of the P2D function was much less than the SPD function. So, it was concluded that the P2D distance function was the suitable choice for optimal urban facility location using spatial clustering, considering geographical obstacles.

Results of execution models on the data sets

Figures 8, 9 and 10 show the algorithm execution results on Zone 1, 5-Isf-River dataset (the first row of Table 1) using the SPD distance function, using the P2D distance function and using the PD distance function, respectively. The results of algorithm execution by the Euclidean distance function on both sides of the river were separately observable in Fig. 11.

As shown in Figs. 8 and 9, the center of clusters and points allocated to each cluster using the SPD and P2D distance functions were similar. The points on both sides of the river regarding communicating bridges were allocated to one facility which was not separately locatable to each river side as shown in Fig. 11.

The logistic cost using both the SPD and P2D distance functions were similar in Zone 1, 5-Isf-River data set (first row of Table 1) As shown in Table 2, execution time of the P2D distance function was less than the SPD distance (the first row of Table 3).

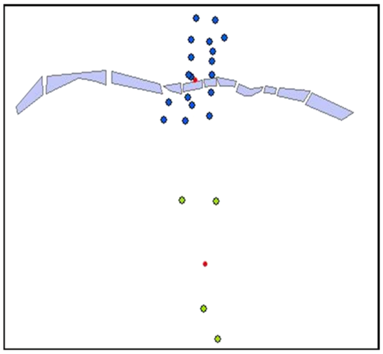


Fig. 8 Algorithm execution results using the SPD distance function on the first row data set

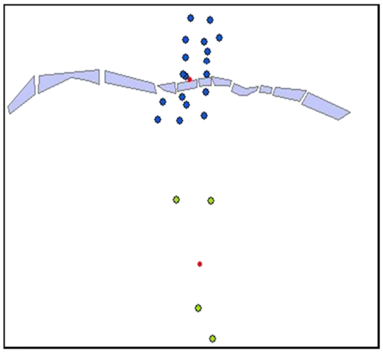


Fig. 9 Algorithm execution results using the P2D distance function on the first row data set

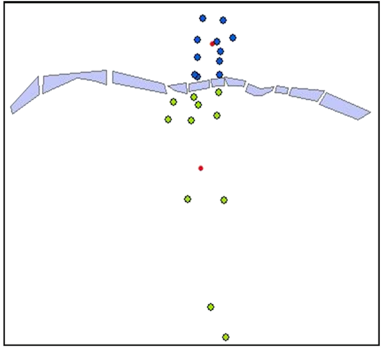


Fig. 10 Algorithm execution results using the PD distance function on the first row data set

The points of both sides of the river did not contain the same cluster using the PD distance function. This was presented in Fig. 10. The logistic cost was more than two P2D and SPD functions that presented in the first row of Table 2. But execution time was less than two P2D and SPD distance functions.

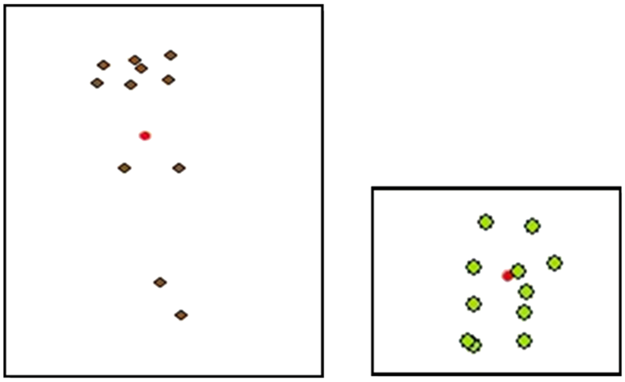


Fig. 11 Algorithm execution results using the Euclidean distance on the two zones separately depicted on both sides of the river

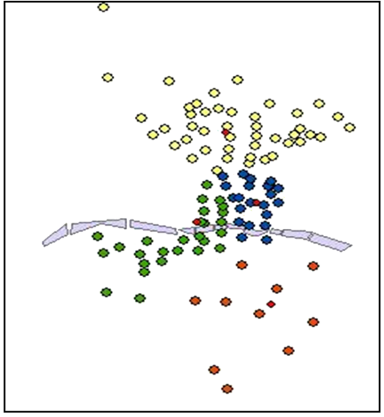


Fig. 12 SPD and P2D functions execution results in the second row of Table 1

The obtained results by executing algorithms in other datasets (rows 2–4 of Table 1) are depicted in Figs. 12, 13, 14, 15, 16, 17, 18 and 19.

Algorithm execution results using the SPD and P2D functions were similar in terms of the centers' location and allocation of demand points to centers. Also points of both sides of the river regarding communicating bridges were allocated to one facility which was not separately locatable to each river side.

As shown in Table 2, the logistic cost using both the SPD and P2D distance functions was similar in Zone10-Isf-River data sets (second row of Table 1) and execution time of the P2D distance function was less than the SPD distance function that was exposed in the second row of Table 3.

As shown in Fig. 13, some points from different sides of the river were placed in one cluster using the PD distance function. It was not possible in the simplified hypothesis

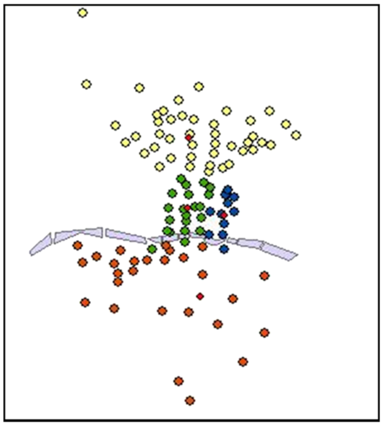


Fig. 13 PD function execution results in the second row data set of Table 1

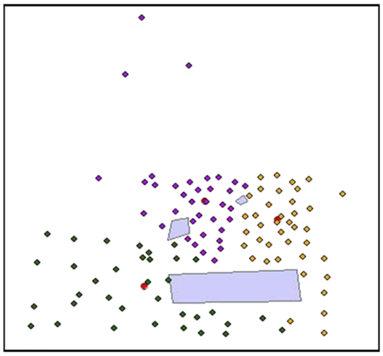


Fig. 14 Algorithm execution result using the SPD function on the third row of the data set

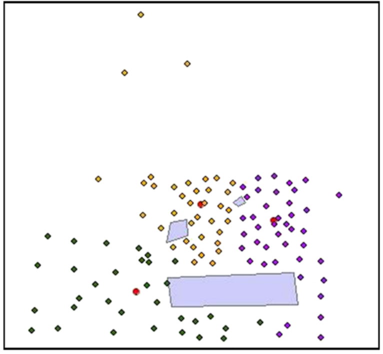


Fig. 15 The results of executing algorithm using the P2D on the third row of the data set

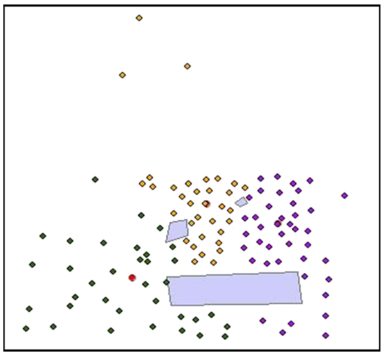


Fig. 16 The results of executing algorithm with the PD on the third row of the data set

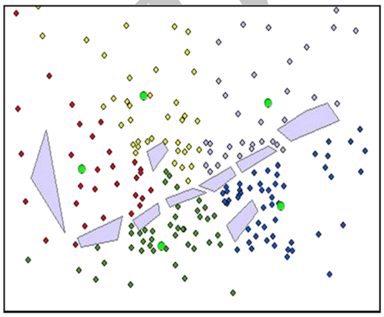


Fig. 17 The results of executing algorithm with the SPD on the fourth row of the data set

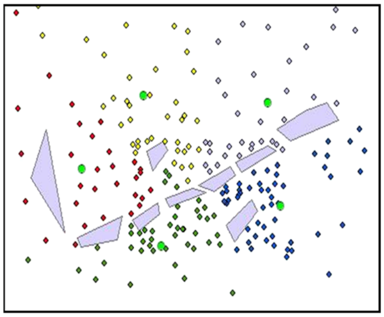


Fig. 18 The results of the executing algorithm with the P2D on the fourth row of the data set

that was allocated separate centers to each side of the river. It is important that the logistics cost in this method was more than two other functions.

The center's locations were similar using the P2D and SPD functions presented in Figs. 14 and 15. It is noted that the allocation of demand points to center was similar. Logistic costs using the P2D function in this dataset were a little more than SPD functions.

The logistic cost using the PD function was more than two SPD and P2D functions in this dataset as given in Table 2.

The center's locations were similar using the P2D and SPD functions that was shown in Figs. 17 and 18.

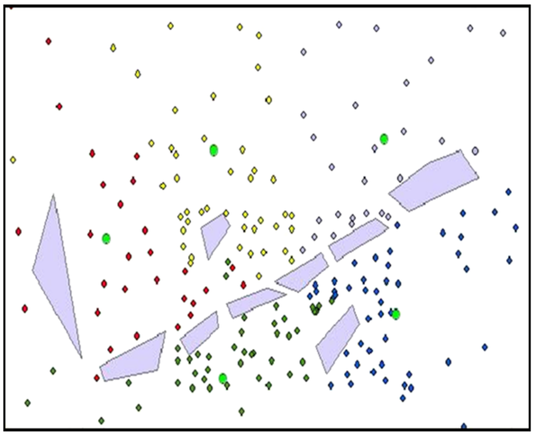


Fig. 19 The results of the executing algorithm with the PD on the fourth row of the data set

Allocation of demand points to center was also similar, but it was a little different using the PD function in Fig. 19.

The results show that logistic costs using the P2D function in this dataset were more than the SPD function and less than PD function.

Conclusion

The facilities, resources should be located around the urban to make sure that all demand points are covered and all the needs are met. It is noted that the facility locations are important for the organizations and the governments to control and make the best urban area for a city. So, some searches are done on solving the facility location problems. Performing spatial clustering has used the Euclidean distance between two points as the dissimilarity function.

In this paper, the total logistic cost was considered equal to the sum of distances between demand points to nearest facility was minimized to find the locations of specific numbers of city facilities.

Cities might have such obstacles like rivers and mountains, which influence the real distance between the two points.

These models were presented for locating facilities, considering spatial obstacles based on the spatial clustering. It is noted that geographical obstacles were considered also in calculating distances between the demand points and facilities.

Then, three new distance functions were proposed based on the nearest neighbor analysis and the path-finding algorithms of the robot's movement in the robotic geometry considering the obstacles and the results are executed.

The synthetic data sets and real data set of Isfahan's population region centers as weighted demand points with a different number of clusters were employed to investigate the execution of these algorithms and the results were obtained. It shown that the results of the robotics geometry (P2D function) were similar to the results of computing the shortest path (SPD function) in terms of the facilities location, allocating demand points to facilities, execution time and logistic costs.

The results were quite similar and the logistic cost was equal in both methods by comparing the two methods in the population region (zones 1 and 5 of Isfahan). The execution time due to drawing lines, network formation and executing the DIJKSTRA algorithm of the SPD method was more than the P2D. The P2D results were equivalent to it, in the synthetic data set.

The results shown that the execution times using the P2D distance function were less than the SPD function. Also, the execution time using the PD distance function was less than the P2D function. The time consuming of Euclidean distance function exertion was less than the other functions. But according to minimize of logistics costs, the P2D and SPD functions were the most appropriate. On the other hand, the execution time of the P2D function was much less than the SPD function. So, it is concluded that the P2D distance function was an appropriate choice for optimal urban facility location using spatial clustering considering geographical obstacles. In the next step, a simplification regarding the Zayanderood River as a continuous obstacle of the city was prepared and both sides of the river were clustered. Then, the obtained results were compared with the proposed models. In the proposed models, the comparison revealed that it is not necessary to divide and allocate the facilities to each side of the river in addition to considering the real distance while hitting an obstacle. In conclusion, these models would be identified facilities with high quality. The demand points in the proposed models were allocated to the facilities such that some of the points are placed in one cluster from the two sides of the obstacle. It is noted that, this issue was not considered in the simplified assumption. The proposed model's results are improved as of the logistic cost rather than the simplified model.

References

Anderberg M (1973) Cluster analysis for applications. Academic Press, New York

Berman O, Krass D (2002) The generalized maximal covering location problem. Comput Oper Res 29(6):563–581

Choset H, Lynch K, Hutchinson S, Kantor G, Burgard W, Kavraki L, Thrun S (2007) Principles of robot motion: theory, algorithms, and implementation (chapter 2). The MIT Press, Cambridge, MA, USA

Church R (2002) Geographical information systems and location science. Comput Oper Res 29(6):541–562

Ester M, Kriegel HP, Sander S, Xu S (1996) 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

Estivill-Castro V, Houle M (2001) Robust distance-based clustering with applications to spatial data mining. Algorithmica 30(2):216–242

Galvao R, Espejo L, Boffey B (2000) A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. Eur J Oper Res 124:377–389

Geetha S, Poonthalir G, Vanathi P (2009) Improved *K*-means algorithm for capacitated clustering. InfoComp J 8(4):52–59

Guha S, Rastogi R, Shim K (1998) CURE: an efficient clustering algorithm for large databases. In: Proceedings of the ACM SIGMOD international conference on management of data, Seattle, pp 73–84

Kaufman L, Rousseeuw P (1990) Finding groups in data: an introduction to cluster analysis. Wiley, New York

Kaveh P, Sabzevari Zadeh A, Sahraeian R (2010) Solving capacitated p-median problem by hybrid k-means clustering and FNS algorithm. Int J Innov Manag Technol 1(4):405–410

Koperski K, Adhikary J, Han J (2001) Spatial data mining: progress and challenges. Survey paper

Liao K, Guo D (2008) A clustering-based approach to the capacitated facility location problem. Trans GIS 12(3):323–339

Likas A, Vlassis N, Verbeek J (2003) The global K-means clustering algorithm. Pattern Recognit 36(2):451–461

Ng R, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Proceedings of the 20th Conference on very large data bases, Santiago, pp 144–155

Pena J, Lozano J, Larranaga P (1999) An empirical comparison of four initialization methods for the K-means algorithm. Pattern Recognit Lett 20:1027–1040

Tung A, Ng R, Laksmanan L, Han J (2001) Constraint-based clustering in large databases. In: Proceedings of the International Conference on database theory, London, pp 405–419

Wang W, Yang J, Muntz R (1997) STING: a statistical information grid approach to partial data mining. In: Proceedings of the 23rd International Conference on very large data bases, Athens, pp 186–195

Wei G, Xin W (2010) Studies on the performance of a heuristic algorithm for static and transportation facility location allocation problem. In: Zhang W, Chen Z, Douglas CC, Tong W (eds) High performance computing applications. Lecture notes in computer science, vol 5938. Springer, Heidelberg, pp 27–37

Zarnani A, Rahgozar M, Lucas C, Taghiyareh F (2007) Spatial data mining for optimized selection of facility location s in field-based service. In: Proceedings of the IEEE International symposium of computational intelligence and data mining. IEEE Press, Hawaii, pp 734–741

Zhang J, Papadias D, Mouratidis K, Manli Z (2005) Query processing in spatial databases containing obstacles. Int J Geogr Inf Sci 19(10):1091–1111