

## Research Article

# A Novel Artificial Immune Algorithm for Spatial Clustering with Obstacle Constraint and Its Applications

Liping Sun,<sup>1,2</sup> Yonglong Luo,<sup>1,2</sup> Xintao Ding,<sup>2</sup> and Ji Zhang<sup>3</sup>

<sup>1</sup> College of National Territorial Resources and Tourism, Anhui Normal University, China

<sup>2</sup> Engineering Technology Research Center of Network and Information Security, Anhui Normal University, China

<sup>3</sup> Faculty of Health, Engineering and Sciences, University of Southern Queensland, Australia

Correspondence should be addressed to Yonglong Luo; ylluo@ustc.edu.cn

Received 18 July 2014; Accepted 23 August 2014; Published 4 November 2014

Academic Editor: Jianjun Yang

Copyright © 2014 Liping Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An important component of a spatial clustering algorithm is the distance measure between sample points in object space. In this paper, the traditional Euclidean distance measure is replaced with innovative obstacle distance measure for spatial clustering under obstacle constraints. Firstly, we present a path searching algorithm to approximate the obstacle distance between two points for dealing with obstacles and facilitators. Taking obstacle distance as similarity metric, we subsequently propose the artificial immune clustering with obstacle entity (AICOE) algorithm for clustering spatial point data in the presence of obstacles and facilitators. Finally, the paper presents a comparative analysis of AICOE algorithm and the classical clustering algorithms. Our clustering model based on artificial immune system is also applied to the case of public facility location problem in order to establish the practical applicability of our approach. By using the clone selection principle and updating the cluster centers based on the elite antibodies, the AICOE algorithm is able to achieve the global optimum and better clustering effect.

## 1. Introduction

Spatial clustering analysis is an important research problem in data mining and knowledge discovery, the aim of which is to group spatial data points into clusters. Based on the similarity or spatial proximity of spatial entities, the spatial dataset is divided into a series of meaningful clusters [1]. Due to the spatial data cluster rule, clustering algorithms can be divided into spatial clustering algorithm based on partition [2, 3], spatial clustering algorithm based on hierarchy [4, 5], spatial clustering algorithm based on density [6], and spatial clustering algorithm based on grid [7].

The distance measure between sample points in object space is an important component of a spatial clustering algorithm. The above traditional clustering algorithms assume that two spatial entities are directly reachable and use a variety of straight-line distance metrics to measure the degree of similarity between spatial entities. However physical barriers often exist in the realistic region. If these obstacles and facilitators are not considered during the clustering process, the clustering results are often not realistic. Taking the simulated

dataset in Figure 1(a) as an example, where the points represent the location of consumers, the clustering result shown in Figure 1(b) can be obtained, when the rivers and hill as obstacles are not considered. If the obstacles are taken into account and bridges as facilitators are not considered, the clustering result in Figure 1(c) can be gained. Considering both the obstacles and facilitators, Figure 1(d) demonstrates the more efficient clustering patterns.

At present, only a few clustering algorithms consider obstacles and/or facilitators in the spatial clustering process. COE-CLARANS algorithm [8] is the first spatial clustering algorithm with obstacles constraints in a spatial database, which is an extension of classic partition clustering algorithm. It has similar limitations to the CLARANS algorithm [9], which has sensitive density variation and poor efficiency. DBCluC [10] extends the concepts of DBSCAN algorithm [11], utilizing obstruction lines to fill the visible space of obstacles. However, it cannot discover clusters of different densities. DBRS+ is the extension of DBRS algorithm [12], considering the continuity in a neighborhood. Global parameters used by DBRS+ algorithm make it suffer from

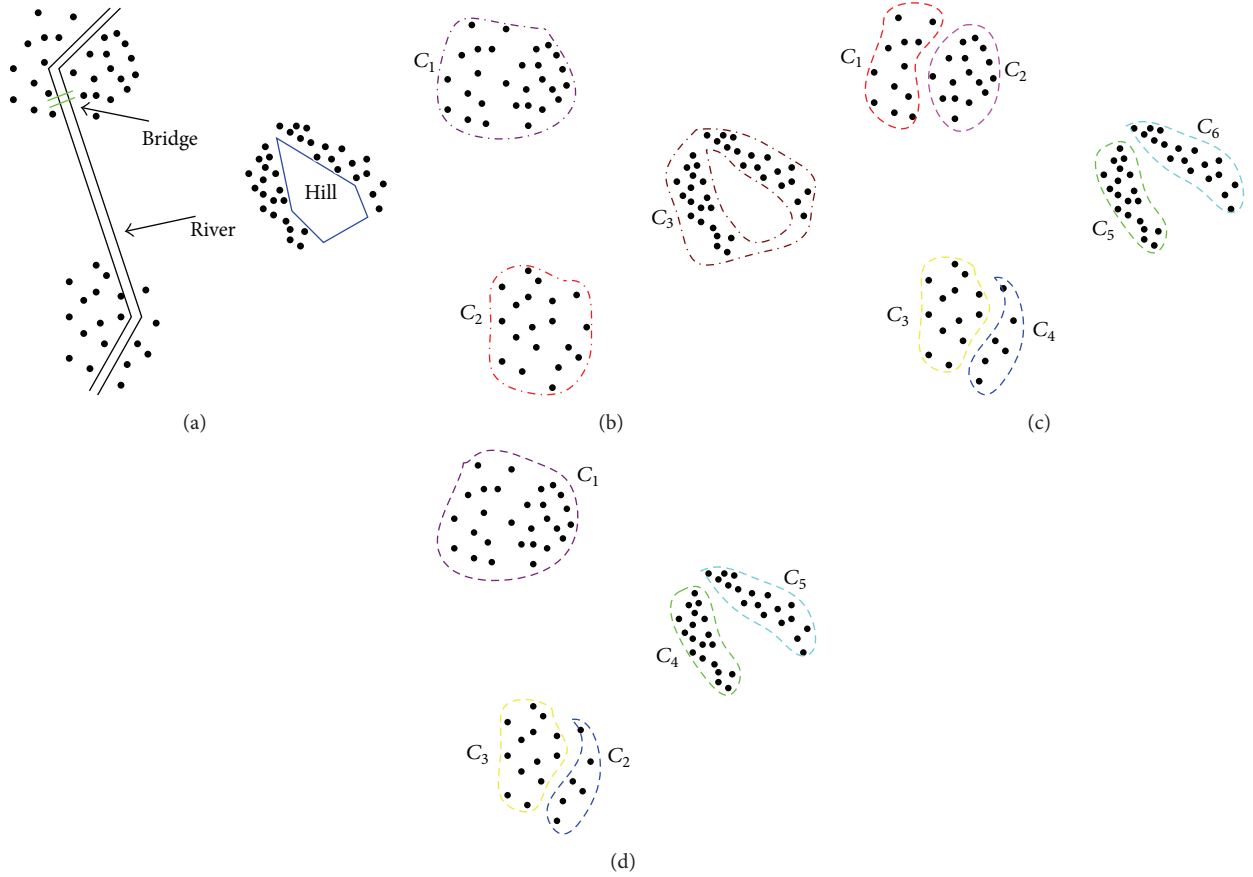


FIGURE 1: Spatial clustering with obstacle and facilitator constraints: (a) spatial dataset with obstacles; (b) spatial clustering result ignoring obstacles; (c) spatial clustering result considering obstacles; (d) spatial clustering result considering both obstacles and facilitators.

the problem of uneven density. AUTOCLUST+ is a graph-based clustering algorithm, which is based on AUTOCLUST clustering algorithm [13]. For the statistical indicators used by AUTOCLUST+ algorithm, it could not deal with planar obstacles. Liu et al. presented an adaptive spatial clustering algorithm [14] in the presence of obstacles and facilitators, which has the same defect as AUTOCLUST+ algorithm.

Recently, the artificial immune system (AIS) inspired by biological evolution provides a new idea for clustering analysis. Due to the adaptability and self-organising behaviour of the artificial immune system, it has gradually become a research hotspot in the domain of smart computing [15–20]. Bereta and Burczyński performed the clustering analysis by means of an effective and stable immune  $K$ -means algorithm for both unsupervised and supervised learning [21]. Gou et al. proposed the multielitist immune clonal quantum clustering algorithm by embedding a potential evolution formula into affinity function calculation of multielitist immune clonal optimization and updating the cluster center based on the distance matrix [22]. Liu et al. put forward a novel immune clustering algorithm based on clonal selection method and immunodominance theory [23].

In this paper, a path searching algorithm is firstly proposed for the approximate optimal path between two points among obstacles to achieve the corresponding obstacle distance. It does not need preprocessing and can deal with both

linear and planar obstacles. Based on the path searching algorithm, a spatial clustering algorithm is proposed to the spatial data clustering in the presence of both obstacles and facilitators. A case study is also carried out to apply our method to the problem of public facility optimization.

The remainder of this paper is organized as follows. Section 2 at first presents the path searching algorithm and then elaborates the details of AICOE algorithm, including analysis of population partition, the design of affinity function, and immune operators. Section 3 shows the experimental results. Section 4 presents the conclusions and main findings.

## 2. Theoretical Framework

**2.1. Obstacles Representation.** Physical obstacles in the real world can generally be divided into linear obstacles (e.g., river, highway) and planar obstacles (e.g., lake). Facilitators (e.g., bridge) are physical objects which can strengthen straight reachability among objects. In processing geospatial data, representation of the spatial entities needs to be firstly determined [14]. In this paper, the vector data structure is used to represent spatial data. Obstacles entities are approximated as polylines and polygons. A facilitator is abstracted as a vertex on an obstacle.

Relevant definitions are provided as follows.

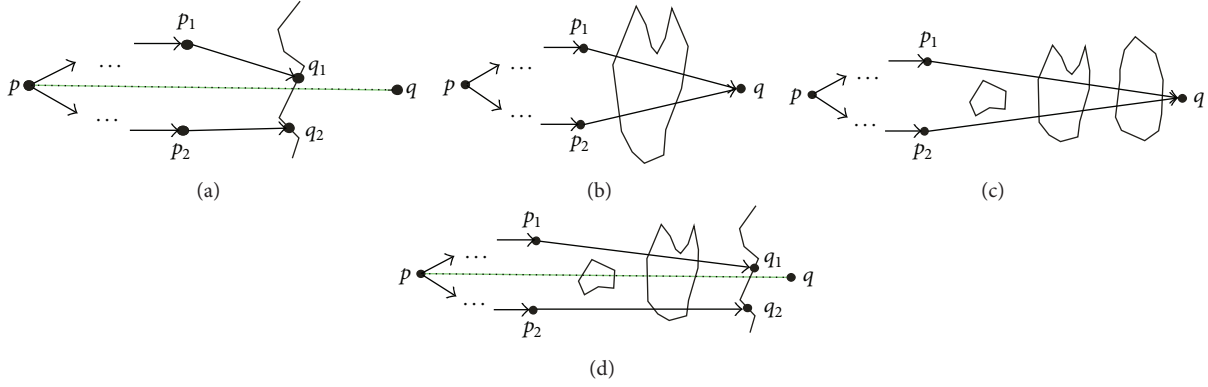


FIGURE 2: Construction of approximate optimal path between two points with obstacle constraints: (a) intersect with a linear obstacle; (b) intersect with the last planar obstacle; (c) intersect with a planar obstacle and obstacles behind it are all planar; (d) intersect with a planar obstacle and the obstacle behind it is linear.

**Definition 1** (linear obstacles). Let  $L = \{L_i \mid L_i = (V_i^{(L)}, E_i^{(L)})$ ,  $i \in \mathbb{Z}^+\}$  be polyline obstacles set, where  $V_i^{(L)}$  is the set of vertices of  $L_i$ ;  $E_i^{(L)} = \{(v_k, v_{k+1}) \mid v_k, v_{k+1} \in V_i^{(L)}, v_k \text{ is the adjacent vertex of } v_{k+1}, k = 1, \dots, M_i - 1, M_i \text{ is the number of } V_i^{(L)}\}$ .

**Definition 2** (planar obstacles). Let  $S = \{S_i \mid S_i = (V_i^{(S)}, E_i^{(S)})$ ,  $i \in \mathbb{Z}^+\}$  be polygon obstacles set, where  $V_i^{(S)}$  is the set of vertices of  $S_i$ ;  $E_i^{(S)} = \{(v_k, v_{(k+1) \bmod N_i}) \mid v_k, v_{(k+1) \bmod N_i} \in V_i^{(S)}, v_k \text{ is the adjacent vertex of } v_{(k+1) \bmod N_i}, k = 1, \dots, N_i, N_i \text{ is the number of } V_i^{(S)}\}$ .

**Definition 3** (facilitators). Let  $V_c = \{V_i^{(C)} \mid V_i^{(C)} \text{ is the set of facilitators on the } i\text{th obstacle}\}$ .

**Definition 4** (direct reachability). For any two points  $p, q$  in a two-dimensional space,  $p$  is called directly reachable from  $q$ , if segment  $pq$  does not intersect with any obstacle; otherwise,  $p$  is called indirectly reachable from  $q$ .

**2.2. The Obstacle Distance between the Spatial Entities.** Currently, the method of distance calculation often computes Euclidean distance between two clustering points. When physical obstacles exist in the real space, obstacles constraints should be taken into account to solve the distance between the two entities in the space. The algorithm handles linear obstacles and planar obstacles, respectively. When traversing linear obstacles, facilitators are also taken into account for path construction. Figure 2(a) illustrates the process of constructing approximate optimal path for linear obstacle, which presents a schematic view of Step 4 of the algorithm. When traversing planar obstacles, path is generated by the method to construct the minimum convex hull. In the case of no more than 100,000 two-dimensional space data samples, the calculation of the minimum convex hull can be finished within a few seconds [24]. Here Graham algorithm is used to produce the minimum convex hull [25]. Figures 2(b) and 2(c) and Figure 2(d), respectively, illustrate the construction process of the approximate optimal path for planar obstacles.

Figure 2(b) shows a schematic view of the first case of Step 5. Figures 2(c) and 2(d) demonstrate a schematic view of the second case of Step 5.

For the sake of easy presentation of the path searching algorithm, the relevant symbols are defined as follows. Let  $o_i \in L \cup S$  be an obstacle, and  $V_i^{(l)}(\vec{pq}) \subset V_c$  is the vertex subset of  $o_i$  on your left hand when you walk along vector  $\vec{pq}$  from point  $p$  to  $q$ . Similarly,  $V_i^{(r)}(\vec{pq}) \subset V_c$  is the vertex subset of  $o_i$  on the right hand.  $Gra(U, p, q)$  is the smallest convex hull which is constructed from the start point  $p$  to the end point  $q$  containing all the points of the vertex set  $U$ .  $Path^{(c)}(Gra(U, p, q))$  denotes the path from the start point  $p$  to the end point  $q$ , which is constructed by the adjacent edges of  $Gra(U, p, q)$  in the clockwise direction;  $Path^{(cc)}(Gra(U, p, q))$  denotes the path from the start point  $p$  to the end point  $q$ , which is constructed by the adjacent edges of  $Gra(U, p, q)$  in the counterclockwise direction.  $path1$  and  $path2$ , respectively, are the obstacle paths on the left and right hand of  $\vec{pq}$ . When new segments are added to  $path1$  and  $path2$ , the start points of the added segments are denoted by  $p1$  and  $p2$ , respectively. Similarly, the end points are denoted by  $q1$  and  $q2$ .  $d_o(p, q)$  represents the obstacle distance between two spatial entities. If  $p$  is directly reachable from  $q$ ,  $d_o(p, q)$  is Euclidean distance between the two points, denoted by  $d(p, q)$ ; if  $p$  is indirectly reachable from  $q$ , path is configured to bypass the obstacles while  $p, q$ , respectively, are taken as the start and end points.

The path searching algorithm for the approximate optimal path between two points among obstacles can be elaborated as follows.

**Step 1.** If  $p$  is directly reachable from  $q$ , then  $d_o(p, q) = d(p, q)$ , and the algorithm is terminated; otherwise, go to Step 2.

**Step 2.** Find the obstacles intersect with  $\vec{pq}$ , which in turn are represented as  $o_1, o_2, \dots, o_m \in L \cup S$ , where  $m$  is the number of the obstacles.

**Step 3.** Consider  $path1 = \phi$ ,  $path2 = \phi$ ,  $p1 = p2 = p$ , and  $i = 0$ .

Step 4. If  $o_i \in L$ , execute the following steps.

- (i) Select the vertex  $u \in V_i^{(l)}(\vec{pq})$  which has the smallest distance to  $\vec{pq}$ .
- (ii) Select the vertex  $v \in V_i^{(r)}(\vec{pq})$  which has the smallest distance to  $\vec{pq}$ .
- (iii) Consider  $q_1 = u$ ,  $q_2 = v$ ,  $path1 = path1 \cup \vec{p_1q_1}$ , and  $path2 = path2 \cup \vec{p_2q_2}$ .
- (iv) Consider  $i++$ ,  $p_1 = q_1$ , and  $p_2 = q_2$ .
- (v) Go to Step 6.

Step 5. If  $o_i \in S$ , there are the following two cases.

(I) If  $i == m$ , execute the following steps.

- (i) If  $\vec{p_1q}$  intersects with  $o_i$ , add  $V_i^{(l)}(\vec{p_1q})$  to  $U_1$ ,  $path1 = path1 \cup Path^{(c)}(Gra(U_1, p_1, q))$ .
- (ii) If  $\vec{p_2q}$  intersects with  $o_i$ , add  $V_i^{(r)}(\vec{p_2q})$  to  $U_2$ ,  $path2 = path2 \cup Path^{(cc)}(Gra(U_2, p_2, q))$ .
- (iii) Consider  $i++$ ,  $p_1 = q$ , and  $p_2 = q$ .
- (iv) Go to Step 6.

(II) If  $i < m$ , execute the following steps.

- (i) If  $o_k (k = i, i+1, \dots, m) \in S$ , execute the following steps.
  - (a) Add  $V_k^{(l)}(\vec{p_1q})$  to  $U_1$ ,  $path1 = path1 \cup Path^{(c)}(Gra(U_1, p_1, q))$ .
  - (b) Add  $V_k^{(r)}(\vec{p_2q})$  to  $U_2$ ,  $path2 = path2 \cup Path^{(cc)}(Gra(U_2, p_2, q))$ .
  - (c) Consider  $i = m$ ,  $p_1 = q$ , and  $p_2 = q$ .
- (ii) If  $o_i, o_{i+1}, \dots, o_k (k < m) \in S$  and  $o_{k+1} \in L$ , execute the following steps.
  - (a) Select the vertex  $u \in V_{k+1}^{(l)}(\vec{pq})$  which has the smallest distance to  $\vec{pq}$ .
  - (b) Select the vertex  $v \in V_{k+1}^{(r)}(\vec{pq})$  which has the smallest distance to  $\vec{pq}$ .
  - (c) Consider  $q_1 = u$  and  $q_2 = v$ .
  - (d) Add  $V_i^{(l)}(\vec{p_1q_1}), V_{i+1}^{(l)}(\vec{p_1q_1}), \dots, V_k^{(l)}(\vec{p_1q_1})$  to  $U_1$ . Consider  $path1 = path1 \cup Path^{(c)}(Gra(U_1, p_1, q_1))$ .
  - (e) Add  $V_i^{(r)}(\vec{p_2q_2}), V_{i+1}^{(r)}(\vec{p_2q_2}), \dots, V_k^{(r)}(\vec{p_2q_2})$  to  $U_2$ . Consider  $path2 = path2 \cup Path^{(cc)}(Gra(U_2, p_2, q_2))$ .
  - (f) Consider  $i = k+1$ ,  $p_1 = q_1$ , and  $p_2 = q_2$ .

Step 6. If  $i < m$ , go to Step 4; otherwise if  $p_1! = q$  and  $p_2! = q$ , then  $path1 = path1 \cup \vec{p_1q}$ ,  $path2 = path2 \cup \vec{p_2q} \cdot d_o(p, q) = \min(\text{length}(path1), \text{length}(path2))$ .

**2.3. Spatial Clustering Algorithm with Obstacle Constraints Based on Artificial Immune System.** Computational intelligence techniques have been widely applied to data engineering research, including classification, clustering, deviation, or outlier detection [19]. Artificial immune system (AIS) is an intelligent method, which mimics natural biological function of the immune system. For its promising performance in immune recognition, the ability of immune learning and immune memory, AIS gradually becomes an important branch of intelligent computing [26–29]. In order to solve the problems of the traditional cluster algorithm in sensitivity to the initial value and the tendency to fall into local optimum, while maintaining its advantages of fast convergence speed, a novel spatial clustering algorithm with obstacle constraints is proposed in this paper.

**2.3.1. The Clustering Problem.** Given  $V$ , the goal of a clustering algorithm is to obtain a partition  $I = \{I_1, I_2, \dots, I_k\}$  (i.e.,  $I_i \neq \emptyset$ , for all  $i$ ;  $\bigcup_{i=1}^k I_i = V$ ;  $I_i \cap I_j = \emptyset$ , for all  $i \neq j$ ) which satisfies that objects classified as the same cluster are as similar to each other as possible, whereas objects classified as the different clusters are as dissimilar as possible.

**2.3.2. Antibody Encoding.** Let  $V = \{v_1, v_2, \dots, v_M\}$  be a set of  $M$  sample points, corresponding to the antigen set  $Ags = \{ag_1, ag_2, \dots, ag_M\}$ . The antibody set  $Abs = \{ab_1, ab_2, \dots, ab_N\}$ , where  $N$  is the number of antibodies. Each antibody  $ab_i$  consists of  $k$  cluster centers, and each cluster center can be expressed as a real-value  $d$ -dimensional profile vector which is represented as  $ab_i = \{\underbrace{a_{i1}a_{i2} \dots a_{id}}_{c_1} \dots \underbrace{a_{i1}a_{i2} \dots a_{id}}_{c_i} \dots \underbrace{a_{i1}a_{i2} \dots a_{id}}_{c_k}\}$ , where  $c_i$  corresponds to the center of the  $i$ th cluster.

**2.3.3. Affinity Function Design and Immune Operators.** In most occasions, the most used similarity metric in a clustering algorithm is distance metric. The total within-cluster variance or the total mean-square quantization error (MSE) [30] is calculated as follows:

$$\text{Perf}(V, C) = \sum_{i=1}^m \left\{ \|v_i - c_j\|^2, j = 1, \dots, k \right\}, \quad (1)$$

where  $\|v_i - c_j\|$  denotes the similarity between sample point  $v_i$  and clustering center  $c_j$  and the obstacle distance is used as a distance metric in this paper. Obstacles constraints should be taken into account for clustering algorithms in the paper. On this basis, cluster centers set  $C = \{c_1, c_2, \dots, c_k\}$  and the corresponding partition  $I = \{I_1, I_2, \dots, I_k\}$  are achieved by applying the rule that the nearer sample points are apart from a cluster center in obstacle distance.

Bearing in mind the measurement of the MSE in (1), we design an affinity function  $f_{i,j}$  in (2), which represents the affinity of the antibody of  $i$  with antigen  $j$ . Let  $D_{\text{in-cluster}} = \sum_{j=1}^k \sum_{v_i \in V \cap I_j} d_o(v_i, c_j)$ ; then

$$f_{i,j} = \frac{1}{D_{\text{in-cluster}} + \varepsilon_0}, \quad (2)$$



where  $\varepsilon_0$  is a small positive number to avoid illness (i.e., denominator equals zero).  $fmeans$  denotes the average value of population affinity, which can be calculated as

$$fmeans = \frac{\sum_{i=1}^k \sum_{j=1}^m f_{i,j}}{k}. \quad (3)$$

$M \subseteq Abs$  is memory cell subset. Threshold value of immunosuppression is calculated as

$$\alpha = \frac{1}{k^2} \sum_{i=1}^{k-1} \sum_{j=i+1}^k f'_{i,j}, \quad (4)$$

where  $f'_{i,j} = d_o(c_i, c_j)$ , which represents the affinity of the antibody of  $i$  with antibody of  $j$ .

The antibody selection operations, cloning operations, and mutation operations of AICOE algorithm were defined in the literature [31].

**2.3.4. Artificial Immune Clustering with Obstacle Entity (AICOE) Algorithm.**  $F Ags = \{ag_1, ag_2, \dots, ag_M\}$ , the algorithm is described as follows.

**Step 1.** Initialize antibody set  $Abs(0) = \{ab_1, ab_2, \dots, ab_N\}$ , where  $N$  is the number of antibodies. Consider  $t = 0$ .

**Step 2.** For all  $ag_i \in I_k (1 \leq i \leq M, 1 \leq k \leq N)$ , calculate the value of  $f_{i,k}$  according to (2).

**Step 3.** According to the affinity calculations by Step 2, optimal antibody subset  $bstAS$  is composed of top  $K (K \leq N)$  affinity antibodies where  $bstAS \subseteq Abs(t)$ . Add  $bstAS$  to  $M$ .

**Step 4.** Generation of the next generation antibody set is elaborated as follows.

- (I) Obtain  $bstAS1$  via performing clone operation on  $bstAS$ .
- (II) Obtain  $bstAS2$  via performing mutation operation on  $bstAS1$ . Add  $bstAS2$  to  $M$ .
- (III) Implement the immunosuppression operation on  $M$ . Calculate the value of  $\alpha$  according to (4). For all  $ab_i, ab_j \in M$ , if the value of  $f'_{i,j}$  is less than  $\alpha$ , randomly delete one of the two antibodies.
- (IV) Randomly generate antibody subset to update the next generation antibody set, denoted by  $rdmAS$ .
- (V) Add  $M$  and  $rdmAS$  to  $Abs(t+1)$ . Consider  $t = t + 1$ .

**Step 5.** Calculate the value of the  $fmeans$  of contemporary population by using (3). If the difference  $fmeans$  in certain continual iterations does not exceed  $\varepsilon$ , stop the algorithm; otherwise go to Step 2.

### 3. Case Implementation and Results

This paper presents two sets of experiments to prove the effectiveness of the AICOE algorithm. The first experiment

uses a set of simulated data, which are generated by the simulation of ArcGIS 9.3. Experimental results are compared with  $K$ -means clustering algorithm [2, 3]. The second experiment is carried out on a case study on Wuhu city and compares the results with the COE-CLARANS algorithm [8]. All algorithms are implemented in C# language and executed on *Pentium* 4.3 HZ, 2 GB RAM computers. The main parameters of the algorithm are defined as follows: mutation rate  $p_m = 0.35$ , inhibition threshold  $\alpha = 0.05$ , and the iterative stopping criteria parameter  $\varepsilon = 1.0e - 4$ .

**3.1. Simulation Experimental Results.** The classical  $K$ -means clustering algorithm has been widely used for its simplicity and feasibility. The AICOE algorithm uses obstacle distance defined in this paper for clustering analysis, and  $K$ -means algorithm uses Euclidean distance as similarity measure of samples. Simulated dataset of the first experiment is shown in Figure 3(a). When cluster number  $k = 6$ , the clustering results of  $K$ -means clustering algorithm and AICOE algorithm are shown in Figures 3(b) and 3(c), respectively. Experimental results show that the clustering results of the AICOE algorithm considering obstacles and facilitators are more efficient than  $K$ -means algorithm.

#### 3.2. A Case Study on Wuhu City

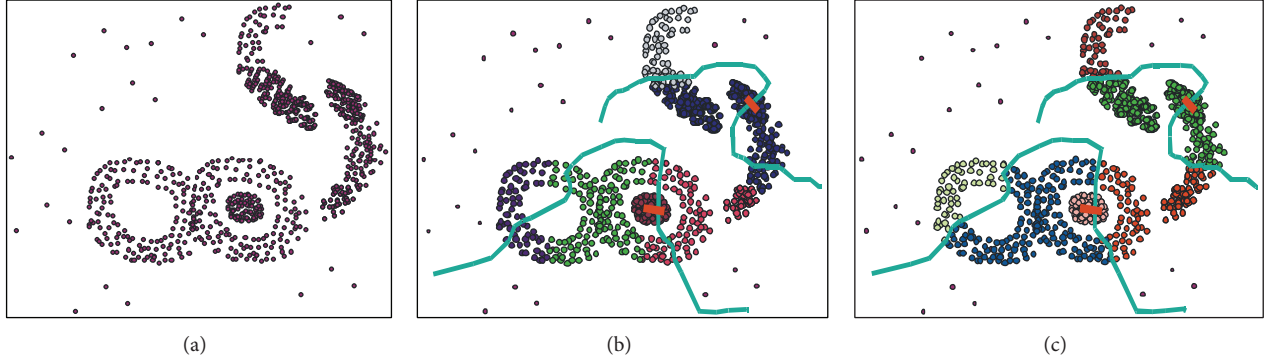
**3.2.1. Study Area and Data.** In this test, the AICOE algorithm is applied to an urban spatial dataset of the city of Wuhu in China (Figure 4). This paper takes 994 residential communities as two-dimensional points, where the points are represented as  $(x, y)$ . In this case study, each residential community is treated as cluster sample point, with its population being an attribute. The highways, rivers, and lakes in the territory are regarded as spatial obstacles, as defined in Definitions 1 and 2, respectively. Pedestrian bridge and underpass on a highway and the bridge on the water body serve as connected points, and the remaining vertices are unconnected points. Digital map of Chinese Wuhu stored in ArcGIS 9.3 was used. And automatic programming has been devised to generate spatial points as cluster points to the address of the residential communities. The purpose of this paper is to find the suitable centers (medoids) and their corresponding clusters.

**3.2.2. Clustering Algorithm Application and Contrastive Analysis.** The COE-CLARANS algorithm [8] and the AICOE algorithm are compared by simulation experiment. The AICOE algorithm uses obstacle distance defined in this paper for clustering analysis. The comparison results of clustering analysis using COE-CLARANS algorithm and AICOE algorithm are shown in Figure 5, and the comparison results of clustering analysis using COE-CLARANS algorithm and AICOE algorithm considering clustering centers are shown in Figure 6.

Given the covered range of different types of public facilities, a clustering simulation is carried out to generate 5, 10, and 15 subclasses, respectively, in this paper. Because Yangtze River is the main obstacle of Wuhu territory, the

TABLE 1: Run time comparison of COE-CLARANS and AICOE (seconds).

Number of points	5 obstacles (50 vertices)		10 obstacles (100 vertices)		20 obstacles (200 vertices)	
	COE-CLARANS	AICOE	COE-CLARANS	AICOE	COE-CLARANS	AICOE
25 k	25.86	20.58	31.54	25.69	36.92	26.87
50 k	43.16	31.59	46.85	35.67	49.36	36.98
75 k	58.22	39.56	61.23	41.23	63.33	42.36
100 k	82.63	64.55	83.79	65.32	83.94	65.87

FIGURE 3: Clustering spatial points in the presence of obstacles and facilitators: (a) simulated dataset; (b) clustering results of  $K$ -means algorithm with obstacles and facilitators; (c) clustering results of AICOE algorithm with obstacles and facilitators.

clustering result of its surrounding regions can demonstrate the validity of the algorithm. Setting cluster number  $k = 5$ , the clustering results of the AICOE algorithm show that only one clustered region 2 has been passed through by Yangtze River where Wuhu Yangtze River Bridge plays a role as a facilitator. While the clustering results of the COE-CLARANS algorithm show that Yangtze River has passed through two clusters, the clustered region 2 does not have any facilitators. Setting cluster number  $k = 10$ , the clustering results of the COE-CLARANS algorithm show that Yangtze River has passed through three subclass regions and the clustered regions 3 and 4 do not have any facilitators. Setting cluster number  $k = 15$ , there does not exist any facilitator in the clustered region 11 obtained by the COE-CLARANS algorithm. In comparison, the clustering results of the AICOE algorithm show that only one clustering region has been passed through by Yangtze River where the facilitator exists. The simulation results demonstrate that the impacts of obstacles on clustering results correspondingly reduce along with the increase in the number of cluster regions.

Figure 7 demonstrates that the COE-CLARANS algorithm is sensitive to initial value, while the AICOE algorithm avoids this flaw effectively. Meanwhile, the AICOE algorithm can get global optimal solution in fewer iterations.

Table 1 shows the results of scalability experiments for the comparison of the COE-CLARANS algorithm and the AICOE algorithm. The synthetic dataset in the following experiments is generated from a Gaussian distribution. The size of dataset varies from 25,000 to 100,000 points. The obstacles and facilitators are generated manually. The number of the obstacles varies from 5 to 20, and the number of vertices

of each obstacle is 10. The number of the facilitators accounts for 20% of the number of the obstacles. Table 1 illustrates that the AICOE algorithm is faster than the COE-CLARANS algorithm.

By comparison of the COE-CLARANS algorithm and the AICOE algorithm for handling spatial clustering with physical constraints, the experimental results show that the COE-CLARANS algorithm causes grouping biases due to its micro-clustering approach. Correspondingly, the AICOE algorithm operates with all the data with less prior preprocessing. The quality of clustering results achieved by the AICOE algorithm surpasses the results of the COE-CLARANS algorithm. Next, the simulation results also indicate that the AICOE algorithm overcomes the COE-CLARANS shortcoming of sensitivity to initial value. The reason for this drawback is that COE-CLARANS algorithm selects the optimum set of representatives for clusters with a two-phase heuristic method. Last, the results of scalability experiments illuminate that the COE-CLARANS algorithm which is affected by the low efficiency of preprocessing runs slower than the AICOE algorithm.

#### 4. Conclusions

Artificial immune clustering with obstacle entity algorithm (i.e., AICOE) has been presented in this paper. By means of experiments on both synthetic and real world datasets, the AICOE algorithm has the following advantages. First, through the path searching algorithm, obstacles and facilitators can be effectively considered with less prior preprocessing compared to the related algorithm (e.g., COE-CLARANS). Then, by embedding the obstacle distance

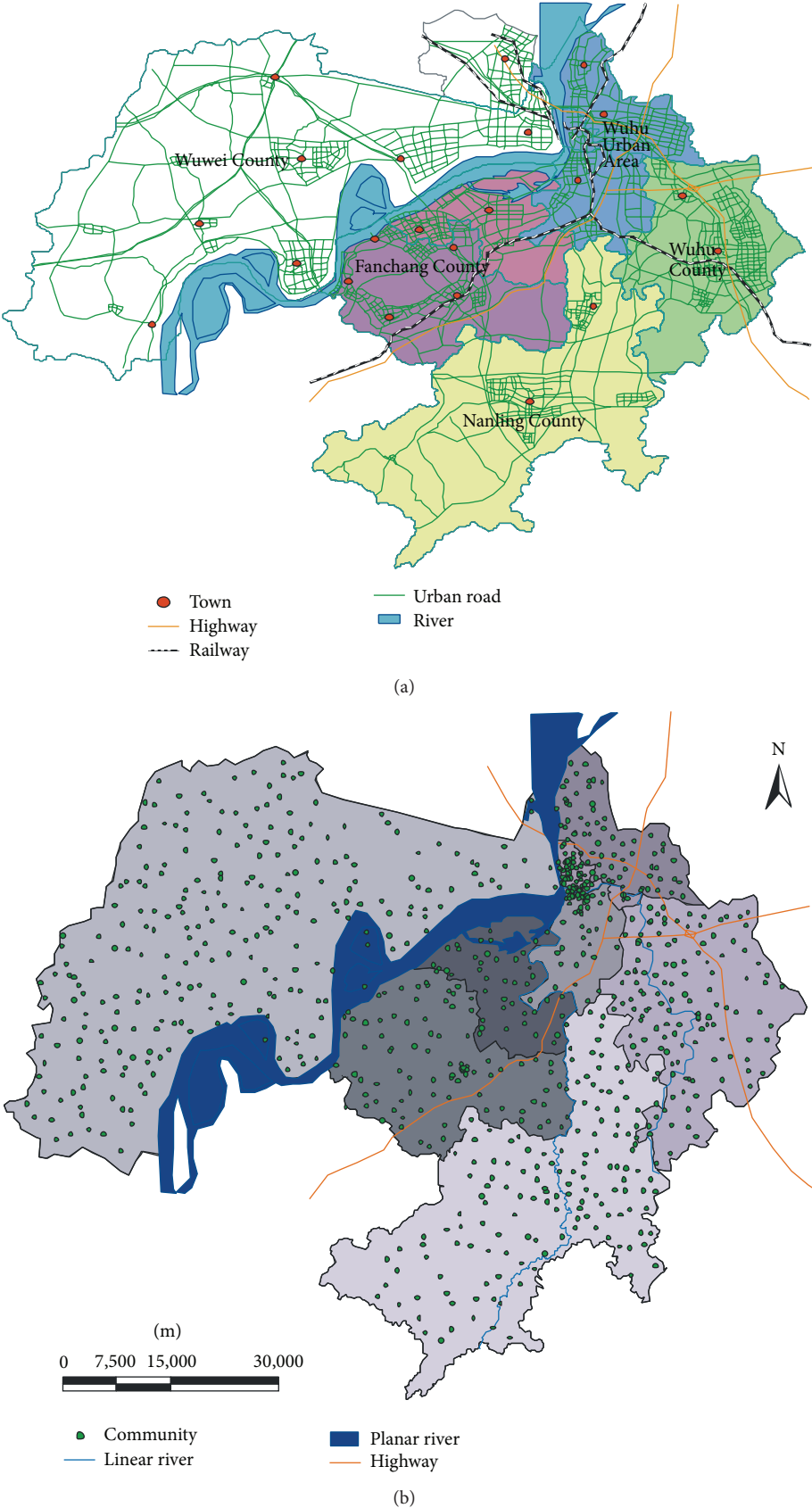


FIGURE 4: The spatial distribution of Wuhu city: (a) administrative map of Wuhu city; (b) the spatial distribution of communities in Wuhu.

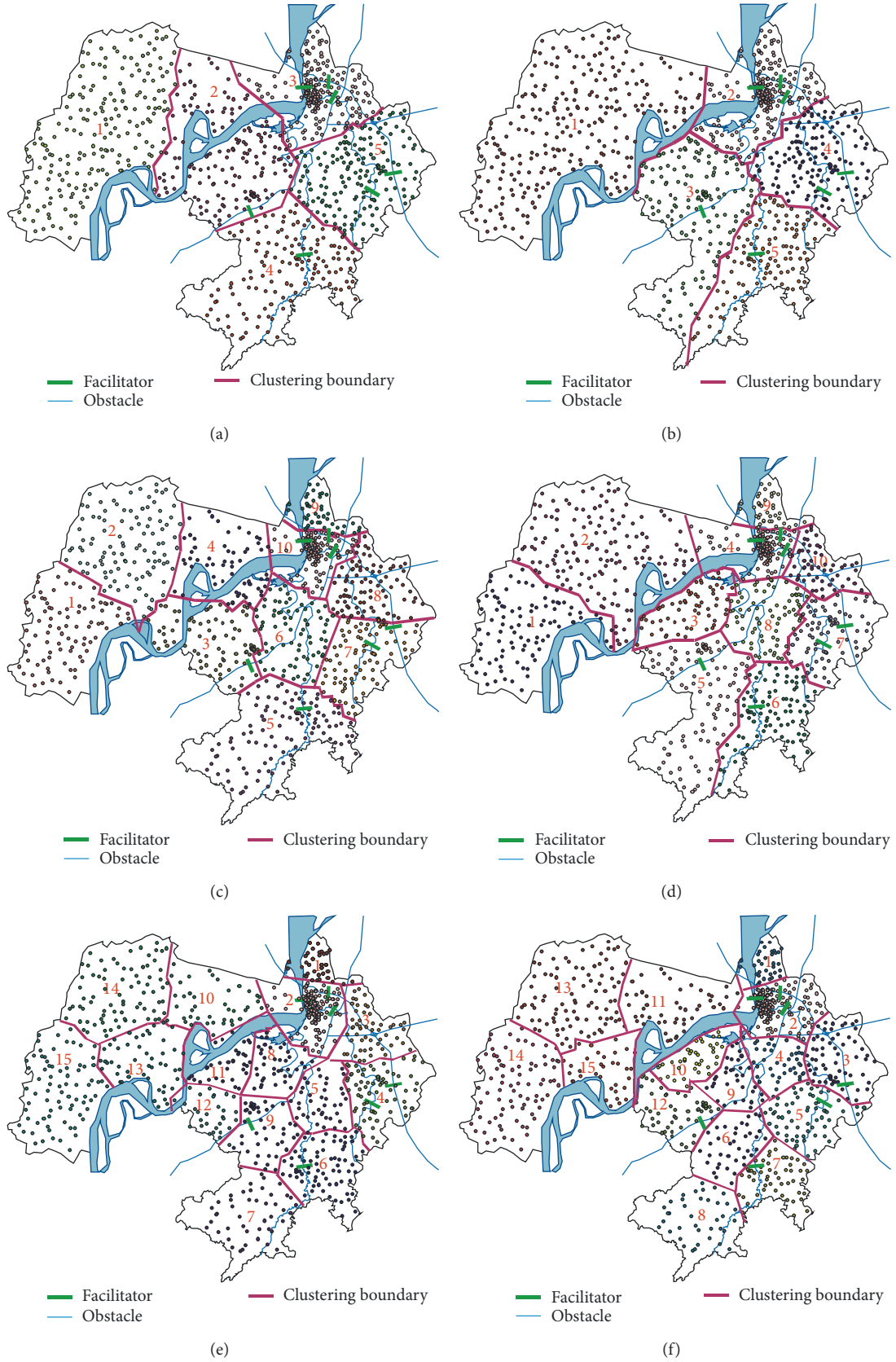


FIGURE 5: Comparison of clustering analysis using the COE-CLARANS algorithm and the AICOE algorithm: (a) 5 subclasses (COE-CLARANS algorithm); (b) 15 subclasses (AICOE algorithm); (c) 10 subclasses (COE-CLARANS algorithm); (d) 15 subclasses (AICOE algorithm); (e) 15 subclasses (COE-CLARANS algorithm); (f) 15 subclasses (AICOE algorithm).



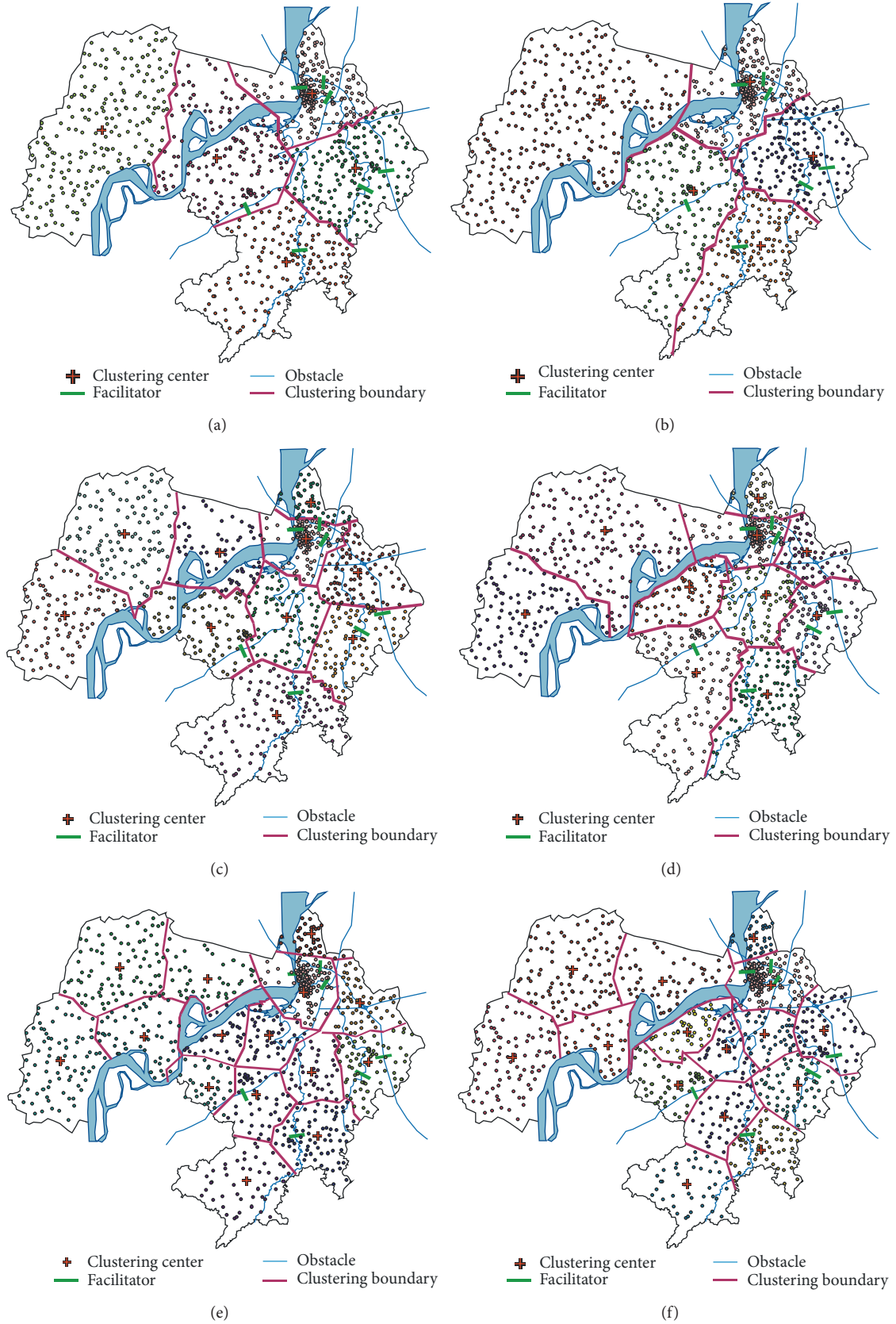


FIGURE 6: Comparison of clustering analysis using the COE-CLARANS algorithm and the AICOE algorithm considering clustering center: (a) 5 subclasses (COE-CLARANS algorithm); (b) 15 subclasses (AICOE algorithm); (c) 10 subclasses (COE-CLARANS algorithm); (d) 15 subclasses (AICOE algorithm); (e) 15 subclasses (COE-CLARANS algorithm); (f) 15 subclasses (AICOE algorithm).

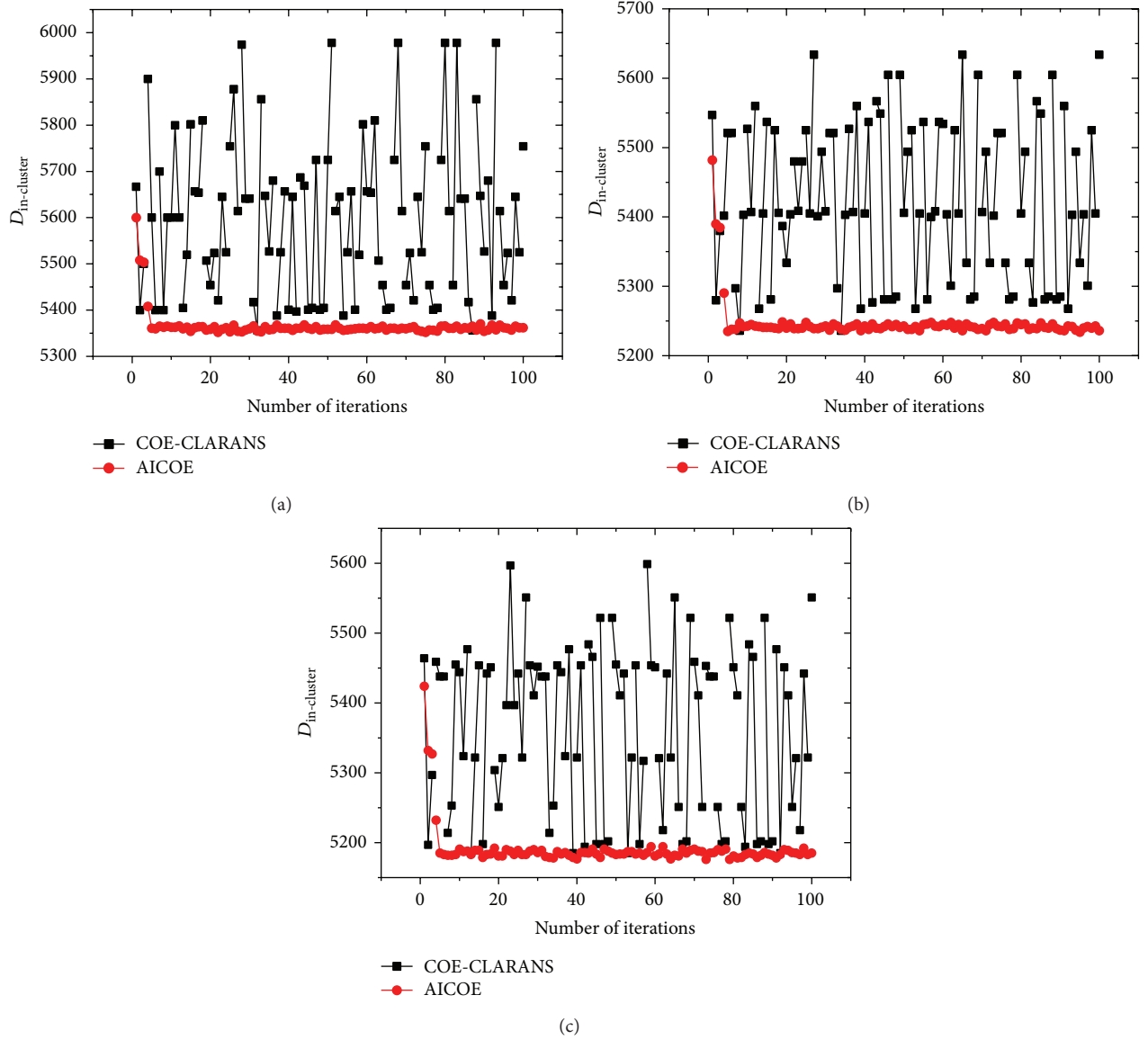


FIGURE 7: Comparison of clustering analysis using the COE-CLARANS algorithm and the AICOE algorithm by intercluster distances: (a) cluster number  $k = 5$ ; (b) cluster number  $k = 10$ ; (c) cluster number  $k = 15$ .

metric into affinity function calculation of immune clonal optimization and updating the cluster centers based on the elite antibodies, the AICOE algorithm effectively solves the shortcomings of the traditional method. The comparative experimental and case study with the classic clustering algorithms has demonstrated the rationality, performance, and practical applicability of the AICOE algorithm.

Due to the complexity of geographic data and the difference of data formats, present researches on spatial clustering with obstacle constraint mainly aim at clustering method for two-dimensional spatial data points [8, 10, 12–14]. There are two directions for future work. One is to extend our approach for conducting comprehensive experiments on more complex databases from real application. The other is to take nonspatial attributes into account for a comprehensive analysis of spatial database.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61370050 and the Natural Science Foundation of Anhui Province under Grant no. 1308085QF118.

## References

- [1] J. Han and M. Kamber, *Data Mining, Concepts and Technologies*, Morgan Kaufmann, Boston, Mass, USA, 2001.

- [2] J. Macqueen, "Some method for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [3] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, NY, USA, 1990.
- [4] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 103–114, Montreal, Canada, 1996.
- [5] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 27, pp. 73–84, 1998.
- [6] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.
- [7] X. Wang, C. Rostoker, and H. Hamilton, "Density-based spatial clustering in the presence of obstacles and facilitators," in *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 446–458, 2004.
- [8] A. Tung, J. Hou, and J. Han, "COE: clustering with obstacles entities, a preliminary study," in *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 165–168, 2000.
- [9] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th Conference on Very Large Databases*, pp. 144–155, Santiago, Chile, 1994.
- [10] O. R. Zaiane and C.-H. Lee, "Clustering spatial data when facing physical constraints," in *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM '02)*, pp. 737–740, IEEE, December 2002.
- [11] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.
- [12] X. Wang, C. Rostoker, and H. J. Hamilton, "A density-based spatial clustering for physical constraints," *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 269–297, 2012.
- [13] V. Estivill-Castro and I. Lee, "Clustering with obstacles for Geographical Data Mining," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 1-2, pp. 21–34, 2004.
- [14] Q. Liu, M. Deng, and Y. Shi, "Adaptive spatial clustering in the presence of obstacles and facilitators," *Computers and Geosciences*, vol. 56, pp. 104–118, 2013.
- [15] W. Ma, L. Jiao, and M. Gong, "Immunodominance and clonal selection inspired multiobjective clustering," *Progress in Natural Science*, vol. 19, no. 6, pp. 751–758, 2009.
- [16] A. Graaff and A. Engelbrecht, "Clustering data in an uncertain environment using an artificial immune system," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 342–351, 2011.
- [17] C.-Y. Chiu, I.-T. Kuo, and C.-H. Lin, "Applying artificial immune system and ant algorithm in air-conditioner market segmentation," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4437–4442, 2009.
- [18] W. Huang and L. Jiao, "Artificial immune kernel clustering network for unsupervised image segmentation," *Progress in Natural Science*, vol. 18, no. 4, pp. 455–461, 2008.
- [19] Q. Cai, H. He, and H. Man, "Spatial outlier detection based on iterative self-organizing learning model," *Neurocomputing*, vol. 117, pp. 161–172, 2013.
- [20] A. J. Graaff and A. P. Engelbrecht, "Using sequential deviation to dynamically determine the number of clusters found by a local network neighbourhood artificial immune system," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2698–2713, 2011.
- [21] M. Bereta and T. Burczyński, "Immune K-means and negative selection algorithms for data analysis," *Information Sciences*, vol. 179, no. 10, pp. 1407–1425, 2009.
- [22] S. Gou, X. Zhuang, Y. Li, C. Xu, and L. C. Jiao, "Multi-elitist immune clonal quantum clustering algorithm," *Neurocomputing*, vol. 101, pp. 275–289, 2013.
- [23] R. Liu, X. Zhang, N. Yang, Q. Lei, and L. Jiao, "Immunodominance based clonal selection clustering algorithm," *Applied Soft Computing*, vol. 12, no. 1, pp. 302–312, 2012.
- [24] J. Wang, "Study of optimizing method for algorithm of minimum convex closure building for 2D spatial data," *Acta Geodaetica et Cartographica Sinica*, vol. 31, no. 1, pp. 82–86, 2002.
- [25] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, vol. 1, no. 4, pp. 132–133, 1972.
- [26] A. Diabat, D. Kannan, M. Kaliyan, and D. Svetinovic, "A optimization model for product returns using genetic algorithms and artificial immune system," *Resources, Conservation and Recycling*, vol. 74, pp. 156–169, 2013.
- [27] O. Er, N. Yumusak, and F. Temurtas, "Diagnosis of chest diseases using artificial immune system," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1862–1868, 2012.
- [28] M. Basu, "Artificial immune system for dynamic economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 1, pp. 131–136, 2011.
- [29] M. M. El-Sherbiny and R. M. Alhamali, "A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem," *Computers and Industrial Engineering*, vol. 64, no. 2, pp. 610–620, 2013.
- [30] Z. Güngör and A. Ünler, "K-harmonic means data clustering with simulated annealing heuristic," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 199–209, 2007.
- [31] L. Castro and F. Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of the Genetic and Evolutionary Computation Conference, Workshop on Artificial Immune Systems and Their Applications (GECCO '00)*, pp. 36–39, 2000.



