Knowl Inf Syst (2012) 31:153–170
DOI 10.1007/s10115-011-0402-8

REGULAR PAPER

Clustering spatial data with a geographic constraint: exploring local search

Zhung-Xun Liao · Wen-Chih Peng

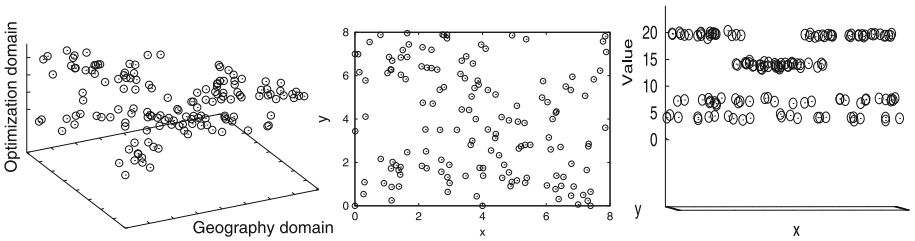
Received: 28 July 2010 / Revised: 29 January 2011 / Accepted: 20 March 2011 /
Published online: 26 April 2011

Abstract Spatial data objects that possess attributes in the optimization domain and the geographic domain are now widely available. For example, sensor data are one kind of spatial data objects. The location of a sensor is an attribute in the geographic domain, while its reading is an attribute in the optimization domain. Previous studies discuss dual clustering problems that attempt to partition spatial data objects into several groups, such that objects in the same group have similar values in their optimization attributes and form a compact region in the geographic domain. However, previous studies do not clearly define compact regions. Therefore, this paper formulates a connective dual clustering problem with an explicit connected constraint given. Objects with a geographic distance smaller than or equal to the connected constraint are connected. The goal of the connective dual clustering problem is to derive clusters that contain objects with similar values in the optimization domain and are connected in the geographic domain. This study further proposes an algorithm CLS (Clustering with Local Search) to efficiently derive clusters. This algorithm consists of two phases: the ConGraph (standing for Connective Graph) transformation phase and the clustering phase. In the ConGraph transformation phase, CLS first transforms the data objects into a ConGraph that captures geographic constraints among data objects and selects initial seeds for clustering. Then, the initial seeds selected nearby data objects and formed coarse clusters by exploring local search in the clustering phase. Moreover, coarse clusters are merged and finely turned. Experiments show that CLS algorithm is more efficient and scalable than existing methods.

Keywords Dual clustering · Spatial clustering · Spatial data mining

Z.-X. Liao · W.-C. Peng (✉)
Department of Computer Science, National Chiao Tung University,
Hsinchu, 30010, Taiwan, ROC
e-mail: wcpeng@cs.nctu.edu.tw

Z.-X. Liao
e-mail: zxliao@cs.nctu.edu.tw



(a) Dual domain **(b)** Geographic domain **(c)** Optimization domain

Fig. 1 An example of the dual clustering problem, where objects have optimization and geographic attributes

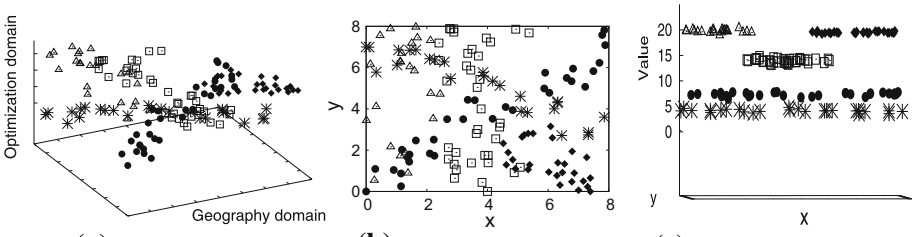
1 Introduction

Spatial data, such as sensor network data and object movement logs, have recently become more widely available [4,10,14,15,17,19,21,26,28]. Spatial data usually have two kinds of attributes: optimization attributes and geographic attributes [16,18,23]. For example, sensors deployed along freeways are utilized to collect readings (e.g., speed readings of vehicles) and monitor traffic status. Clearly, sensor readings are an attribute in the optimization domain, while the location of sensors represents an attribute in the geographic domain. Using spatial data, clustering techniques can find sensors with similar readings. Specifically, given a set of sensors with their locations and sensor readings, traditional clustering approaches partition sensors according to their similarity [5,9,11,25]. Most clustering algorithms calculate similarity as a distance function and determine the dissimilarity between any two sensors based on their optimization attributes [8,12]. Therefore, this kind of clustering algorithm aims to minimize the dissimilarity between each cluster.[1] In the aforementioned example (i.e., sensors for monitoring traffic status of freeways), sensors with similar sensing readings are grouped together. However, this approach may group two sensors whose locations are far away due to their similar sensing readings. Clearly, it is also necessary to consider geographic attributes. As such, this study proposes a dual clustering problem for clustering objects over optimization and geographic domains. Intuitively, optimization (respectively, geographic) attributes belong to the optimization (respectively, geographic) domain. The goal of dual clustering over the optimization and geographic domains is to group objects if their optimization attributes are similar. Objects in the same cluster form a compact region in terms of geographic attributes [6,7,16,18,20,23,24,27].

Consider the example in Fig. 1a, where the value distribution of objects' geographic (respectively, optimization) attribute appears in Fig. 1b (respectively, Fig. 1c). With the set of objects in Figs. 1a, 2a shows the result of dual clustering. As Fig. 2a shows, there are five clusters and objects in the same cluster are marked with the same symbol. Figure 2b, c shows that objects in the same cluster are connective and have similar values in the optimization domain.

Previous research [16,23] explores the dual clustering problem and proposes efficient methods. However, previous approaches fail to specify geographic constraint clearly. In the dual clustering problem discussed in the study by Lin et al. [16,23], objects in the same cluster form a compact region in the geographic domain. Another paper by Lin et al. [16] proposes an Interleaved Clustering-Classification (ICC) algorithm that interlaces clustering

[1] Attributes used to measure the dissimilarity are called optimization attributes because the optimization objective of clustering is to minimize the dissimilarity between each cluster.



(a) Dual domain **(b)** Geographic domain **(c)** Optimization domain

Fig. 2 Clustering results for the example in Fig. 1

Table 1 Comparison of ICC, BINGO, and CLS

Algorithms	ICC	BINGO	CLS
Geographic domain	Complete-link	T-region (grid)	Connected constraint
Optimization domain	SVM	Expanding T-regions	Local search
Data transformation	None	NeiGraph (O^3)	ConGraph (O^2)
Seed selection	None	Random selection	Heuristic selection

and classification processes to discover clusters from spatial data objects. This approach applies a support vector machine (SVM) [1] to the optimization domain in the classification phase and uses a complete-link algorithm [13] to the geographic domain in the clustering phase. However, SVM parameters are difficult to determine, and the time complexity caused by the complete-link method is tremendous. Previous authors [23] proposed an improvement to the ICC, called BINGO, which is a three-phase algorithm: binding information, generating clusters, and tuning borders. The binding-information phase first partitions the geographic domain into several grids in a top-down manner until each grid is a T-region. Objects in the same grid are within a distance threshold T in the optimization domain. The first phase further constructs a NeiGraph by representing each T-region as a node ,and an edge means the two nodes are next to each other. However, although two nodes are next to each other, they may not be in close proximity, due to the nature of the grid. The generating-clusters phase selects representative nodes in NeiGraph as seeds and then expands the seeds through their edges to form clusters. Finally, after the objects change, the tuning borders phase updates the cluster borders. Although the method in the study by Tai et al. [23] is more efficient than that in Lin et al. [16], the NeiGraph transformation is still inefficient because the time complexity of transformation is $O(n^3)$, where n is the number of objects. This approach also poorly defines the compact region, which is strongly related to the size of the grid instead of the distances in the geographic domain. For example, suppose that two sensors with similar sensing readings are deployed in different cities. The BINGO algorithm would still cluster them as a compact region if the grid size was large. This paper presents a connective dual clustering problem with an explicit connected constraint in the geographic domain to guarantee the closeness of objects in a cluster. The requirement of clustering results considered in this paper is that objects in the same cluster should have a geographic distance, satisfies the connected constraint, and minimizes the dissimilarity of each cluster. Table 1 compares the different features of ICC, BINGO, and CLS algorithms.

To the best of our knowledge, this is the first study to define geographic constraints in the dual clustering problem clearly. By defining an explicit geographic constraint, the proposed dual clustering problem is more meaningful. This paper also proposes a CLS

Table 2 Notations and symbols used in this paper

Notation	Description
o_i	Spatial object i
S_i	Optimization domain of o_i
L_i	Geographic domain of o_i
s_i^j	The jth attribute in S_i
l_i^j	The jth attribute in L_i
d_S	The number of dimensions in the optimization domain
d_L	The number of dimensions in the geographic domain

(clustering with local search) algorithm, to discover connective dual clusters using a local search mechanism. The proposed CLS algorithm consists of two phases: the ConGraph transformation phase and the clustering phase. The first phase uses an efficient transformation mechanism to capture the geographic information of spatial objects into pairwise relationships between objects if the geographic distances between spatial objects are smaller than or equal to the geographic constraint given. Unlike BINGO [23], the proposed method only considers geographic attributes in this phase. Furthermore, several objects are selected as cluster seeds for clustering. In the second phase, each seed searches for its member locally, deriving coarse clusters via ConGraph. These coarse clusters are then merged and finely tuned to produce the final clusters. Because seeds should be judiciously selected, this study also proposes a seed selection method. The extensive experiments in this study evaluate the performance of the proposed algorithm. Experimental results show that the CLS algorithm is efficient and scalable compared with existing methods.

The rest of the paper is organized as follows. Section 2 presents the preliminaries. Section 3 presents the proposed algorithm. Section 4 conducts a performance evaluation. Finally, Sect. 5 offers conclusions.

2 Preliminaries

As in the study by Lin et al. [16], spatial data objects in this study possess two kinds of attributes. One is the attribute in the optimization domain, and the other is the attribute in the geographic domain. Table 2 summarizes the symbols and notations used in this paper.

As in prior works [16,23], the distance between two spatial objects serves as the dissimilarity measurement. Among a variety of distance functions, the Euclidean distance is the most widely employed. Thus, we have the following two distance functions in the optimization domain and the geographic domain.

Definition 2.1 (*Distance functions*) For two objects o_i and o_j, the distance measurement in the geographic domain is formulated as

$$dist_{\text{geo}}(o_i, o_j) = \sqrt{\sum_{k=1}^{d_L} (l_i^k - l_j^k)^2}$$

and the distance measurement in the optimization domain is formulated as

$$dist_{\text{opt}}(o_i, o_j) = \sqrt{\sum_{k=1}^{d_S} (s_i^k - s_j^k)^2}.$$

Based on the definition of distance functions earlier, it is possible to determine the cost of a cluster in the optimization domain. Assume that a cluster C_j has a set of objects (e.g., $(o_1, o_2, \ldots, o_{|C_j|})$), where $|C_j|$ is the number of objects in C_j. The cost of cluster C_j in the optimization domain is then $g(C_j) = \frac{1}{|C_j|} \sum_{i=1}^{|C_j|} dist_{\text{opt}}(o_i, cen_j)$, where cen_j is the centroid of C_j and is derived as $(\frac{1}{|C_j|} \sum_{o_i \in C_j} s_i^1, \frac{1}{|C_j|} \sum_{o_i \in C_j} s_i^2, \ldots, \frac{1}{|C_j|} \sum_{o_i \in C_j} s_i^{d_S})$. Consequently, the average cost of a set of clusters is defined as follows:

Definition 2.2 (*Average cost of clusters*) Let $SC = C_1, C_2, \ldots, C_k$ be a set of clusters. The cost of SC is defined as $f(SC) = \sum_{i=1}^{k} \frac{|C_i|}{n} g(C_i)$, where $n = \sum_{i=1}^{k} |C_i|$.

The constraint in the geographic domain is used to cluster objects such that their distance in the geographic domain is within a threshold. If objects are in the same cluster, they are *connected*. The definition of the connected constraint is as follows.

Definition 2.3 (*Connected constraint*) Given a clusters C_t, where $|C_t| > 1$, and a threshold $r, \forall o_i, o_j \in C_t \wedge o_i \neq o_j, dist_{\text{geo}}(o_i, o_j) \leq r$ or there is a sequence of objects $o_{u1}, o_{u2}, \ldots, o_{un} \in C_t$ such that $dist_{\text{geo}}(o_i, o_{u1}) \leq r, dist_{\text{geo}}(o_{u1}, o_{u2}) \leq r, \ldots$, and $dist_{\text{geo}}(o_{un}, o_j) \leq r$.

From the definitions earlier, the problem addressed in this paper is that, given the number of clusters k, a distance threshold r and n spatial objects $o_1, o_2, \ldots, o_n$ with their attributes in the optimization domain and the geographic domain derive a set of clusters, denoted as $SC = (C_1, C_2, \ldots, C_k)$, such that (1) each object o_i belongs to only one cluster C_j, (2) objects in the same cluster are connected, and (3) the average cost (i.e., $f(SC)$) is minimized.

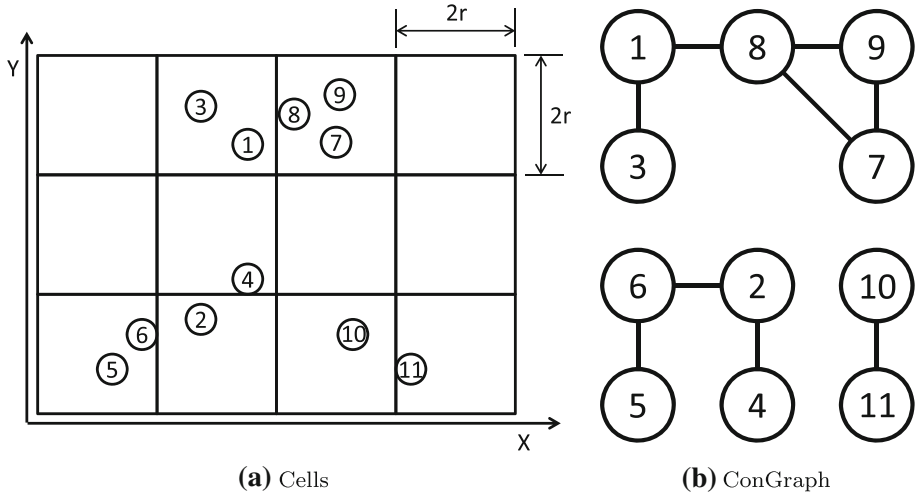
3 The CLS algorithm: clustering with local search

This study proposes the CLS algorithm to cluster the spatial objects from both geographic and optimization domains with connected constraint. The CLS algorithm includes two phases: the ConGraph (standing for connected graph) transformation phase and the clustering phase. First, a ConGraph is constructed according to the geographic attributes of spatial objects, where an edge exists two spatial objects if they satisfy the connected constraint. Once constructing a ConGraph is constructed, the CLS algorithm selects some nodes as seeds to discover possible clusters. Thus, the clustering phase uses a local search mechanism to discover connective dual clusters from seeds in the ConGraph. The main idea of local search is to locally search nearby spatial objects. However, several issues must be dealt with. For example, it is necessary to determine which cluster can be extended and which object should be selected as a cluster member. The following sections present the details of each phase.

3.1 Phase 1: ConGraph transformation

This phase transform an organized ConGraph together with several seeds. To avoid checking the connected constraint in clustering phase, use the ConGraph (connected graph) to represent the connection between data objects. Therefore, the pairwise relations of ConGraph make it possible to determine the connected constraint of each pair of data objects efficiently in the geographic domain. A ConGraph is defined as follows:

Definition 3.1 (*ConGraph*) Given a set of spatial objects $O = \{o_1, \ldots, o_n\}$ and a threshold r, a ConGraph is a graph $G = (O, E)$, where a vertex is an object o_i and an edge $e(o_i, o_j)$ between o_i and o_j exists if $dist_{\text{geo}}(o_i, o_j) \leq r$.



(a) Cells **(b)** ConGraph

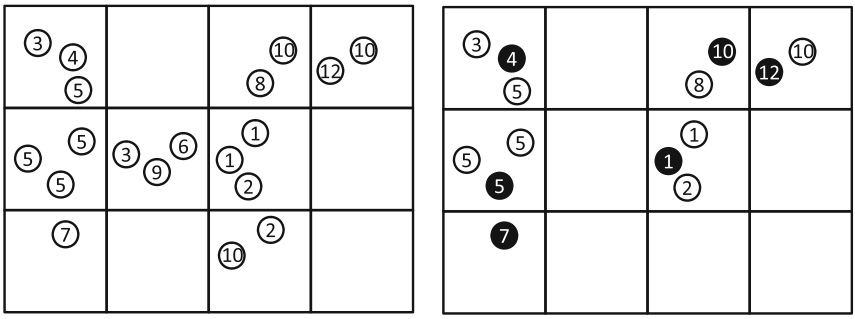
Fig. 3 An example of the ConGraph transformation phase

However, obtaining a ConGraph is expensive if a huge amount of objects is given. Intuitively, it is necessary to compute all distances of each object pair and then verify whether the corresponding distance is within r. Therefore, this study proposes proposed an efficient transformation method that partitions the geographic space into a grid space and assigns each object into the cell in which it resides in geographic location [2]. The advantage of this transformation is that it can avoid the pairwise computation of all objects. In the example shown in Fig. 3a, since the connected constraint is r, the size of cells is set to $2r$-by-$2r$, and only 4 cells must be explored to find the neighbors of a vertex. For example, there are 11 objects in Fig. 3a labeled with their ID, and the transformed ConGraph is shown in Fig. 3b. For the purpose of a concisely representing, the nodes in ConGraph is arranged as grid. The information of optimization domain is omitted here. As Fig. 3b shows, the geographic information of objects is transformed into pairwise relationships.

3.1.1 Time and space analysis of transforming ConGraph

Assume that there are n objects. Let m be the maximum size of cells. The time complexity of constructing the hash table is $O(n)$, and the space complexity is also $O(n)$. Finding all edges for a vertex requires $O(2^{d_L} \cdot m)$ which is smaller or equivalent to $O(n)$. Thus, the total time complexity is $O(2^{d_L} \cdot m \cdot n)$ or $O(n^2)$. When the density of each cell and the dimensions of the geographic domain are low, d_L and m can be viewed as constants. Therefore, finding the edges for a vertex requires only $O(1)$, and thus, finding all edges is $O(n)$. In this case, the total time complexity is $O(n)$.

Once the ConGraph is constructed, pick some vertices as seeds and perform the clustering phase. Note that the selection of initial seeds significantly affects on the cluster results [3,16,23,25]. Consider the example in Fig. 3a, where o_1, o_3, o_8, and o_7 are selected as seeds. Two problems arise here. First, the upper connected graph in Fig. 3b is broken into four parts. Second, the objects in the lower two groups will not be labeled as a cluster because there is no connectivity via seeds. Therefore, the seeds should be as far apart as possible in the geographic domain. On the other hand, seeds should be much different from each other in the optimization domain to fit the objective function $f(SC)$, defined in Definition 2.2.



(a) Before filtering **(b)** After filtering

Fig. 4 An example of filtering during seed selection

To select seeds as far apart as possible, this study uses the advantage of cells in the ConGraph transformation. Since the objects have been placed into cells in the geographic domain, it is possible to treat these cells as temporary clusters because they are inherently far apart in the geographic domain. Therefore, the medoids for each cell on optimization domain are regarded as the candidate seeds. However, the number of candidate seeds may be large when the size of cell is small. To reduce the number of candidate seeds, iteratively filter out half candidate seeds in the cell with higher variance. Define variance of a cell as $\sum_{o \in cell} dist_{\text{opt}}(o, cen)^2$, where cen is the centroid of the cell. Stop filtering process when the number of cells is smaller than k. In addition, to consider the optimization domain, randomly select one candidate as the first seed after deciding the candidate seeds. Then, greedily select the farthest candidate as the next seed until the number of seeds is k. Choose the minimum distance since this operation can reveal the most difference between seeds. Let S and U be the set of selected seeds and the unselected candidates, respectively. The best selection of the next seed is $\arg\max_{u \in U} \min_{s \in S} dist_{\text{opt}}(s, u)$, which is the most different from all selected seeds compared with the other unselected candidates.

Figure 4 shows an example of the seeds selection. The number in each circle indicates its attribute in the optimization domain, and the locations of data objects are their geographic attributes. Figure 4a shows the objects before filtering. After filtering the high variance cells, the results are shown in Fig. 4b, where black nodes are the medoids of cells. The following discussion uses their attribute in the optimization domain as their identification numbers. Assume that o_{10} is the first seed. Then, o_1 would be selected since it is the farthest candidate in the optimization domain compared with o_{10}. Using the sum of distance to select seeds would select o_{12}. However, o_{12} has a similar attribute in the optimization domain to o_{10}. On the other hand, with minimum distance, o_5 would be selected as the third seed.

3.1.2 Time analysis of seed selection

For n objects, b cells, and k clusters, computing the variance of all cells requires $O(n)$, filtering the cells requires $O(b)$, and finding k most different medoids from $max(k, b/2)$ cells requires $O(b \cdot k^2)$. Therefore, the total time complexity is $O(n + b + b \cdot k^2)$. In the worst case, each object forms an unique cell (i.e., $b = n$), and the time complexity is $O(n \cdot k^2)$.

3.2 Phase 2: clustering phase

This phase expands seeds according to the proposed local search mechanism until the clusters remains stable, which is called coarse clustering. The clusters obtained from coarse clustering are called coarse clusters. Then, fine clustering merges coarse clusters according to the connected constraint. The following paragraphs give detailed descriptions, examples, and algorithms.

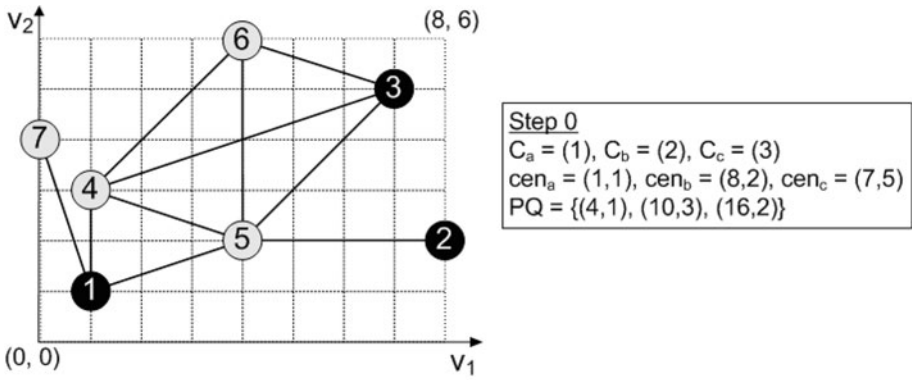
Initially, each seed can be viewed as a cluster and each cluster selects one data object as its member until no more unclustered data objects can be assigned to any clusters. However, there are two challenges held for each cluster: selecting a proper new member and changing the representative in each iteration. In coarse clustering, the new member is selected only from the neighbors of representative in a cluster. Then, the neighbor with minimum distance to the cluster center is selected as the new member. Next, the new member becomes the representative to reduce the cost of finding a new representative and calculating distances between representative and unclustered objects. Only one cluster can expand in each iteration to guarantee that each data object is assigned to the nearest cluster.

Algorithm 3.1 is the coarse clustering algorithm. This algorithm first selects k seeds from the results in the previous phase as initial representatives. Indeed, the centroid of each cluster is also the representative (from line 3 to 7). Adapting the concept of local search [22], those neighbors of these representative objects are extracted. In lines 9 to 15, the distances of these neighbors to the corresponding centroids in the optimization domain are calculated and pushed into a priority queue. Then, only the neighbor with the smallest distance value is selected for the nearest cluster, and the centroid of the corresponding cluster will be updated. Moreover, the representative object for the corresponding cluster is replaced by the new member. This procedure is repeated until no unclustered neighbors remain. After this procedure, the unclustered objects are assigned to the nearest cluster if the connected constraint is observed, as shown in lines 17 to 20.

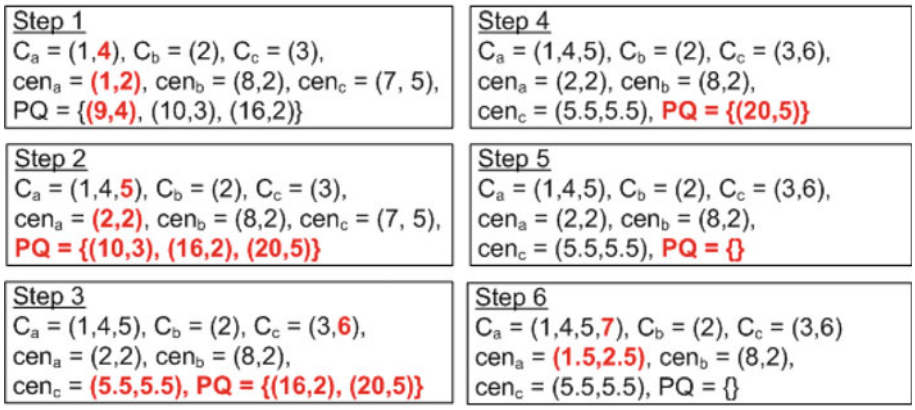
Algorithm 3.1 *Coarse clustering*

Input: *An integer k, a graph* $G = (O, E)$ *and attributes of* O
Output: *A set of clusters SC, where* $|SC| \geq k$

let $N_u(o_i)$ (respectively, $N_c(o_i)$) be the unclustered (respectively, clustered) neighbors of o_i
/* initialization */
select k vertices as the initial seeds and each seed forms a cluster C_t
foreach seed o_i **do**
  let d be the smallest distance $dist_{\text{opt}}(o_i, o_j)$, $o_j \in N_u(o_i)$
  add pair (d, o_i) into a priority queue PQ
end
/* local search */
while $PQ.not\_empty$ **do**
  remove (d, o_i) from PQ with the smallest d
  let C_t be the cluster of o_i, and cen_t be the centroid of C_t
  let d, u be the smallest distance $dist_{\text{opt}}(cen_t, o_j)$ and
    the corresponding vertex, where $o_j \in N_u(o_i)$
  add u into cluster C_t and update cen_t
  add pair (d, u) into PQ
end



(a) Setting



(b) Execution process

Fig. 5 An illustrative example for the coarse clustering

/* fix unclustered objects */
foreach unclustered $o_i \in O$ **do**
  if $|N_c(o_i)| > 0$
  then add o_i to the nearest cluster of $o_j \in N_c(o_i)$ and
    update the corresponding centroid
  else o_i forms a new cluster
end
return the union of all clusters SC

Figure 5 shows an example of the coarse clustering. Figure 5a depicts the ConGraph. There are seven objects, and their coordinates are the attributes in the optimization domain. Assume that we would like to cluster the seven objects into three groups. The procedure of CLS is shown in Fig. 5b. The initial seeds are o_1, o_2 and o_3, and each seed forms a cluster (i.e., C_a, C_b and C_c) in Step 0. Then, the centroid of each cluster is calculated in the optimization domain. PQ is a priority queue to store data pairs (d^2, o_i), where d is the distance between the centroid and the nearest unclustered neighbor of o_i. Here, we use d^2 for the purpose of clearly representing. In Step 1, o_4 is selected and included into cluster C_a. Then,

the unclustered neighbors of o_4 (i.e., o_5, o_6) are searched in the ConGraph. Since the distance between o_5 and the centroid of C_a is the smallest one, the data pair (9, 5) will be inserted into PQ. The same operation can be used iteratively to achieve the clustering results in Step 5. However, there is no representative object in PQ, and o_7 has not been clustered. It is therefore necessary to check the connected constraint and assign o_7 to the nearest cluster, which is C_a. Therefore, the final result of coarse clustering ends in Step 6.

3.2.1 Time analysis of the coarse clustering

For n objects and k clusters, the initialization costs $O(n \cdot k^2)$. A priority queue can be implemented by a heap which requires $O(k)$ to insert a new object or remove the top object. Therefore, the total time complexity for operating the priority queue is $O(n \log k)$. The best representative object does not appear in each iteration. Instead, the new appended object is chosen as the representative object. Because there is a trade-off between the quality of clusters and running time, the time complexity of the local search is $O(|E| + n \log k)$. Finding unclustered objects requires $O(|E|)$, while generating SC requires only $O(n)$. The overall time complexity of this phase is $O(n \cdot k^2 + |E|)$.

The coarse clusters discovered by local search might be fragments of the connective dual clusters due to the nature of local search. Therefore, merge the coarse clusters belonging to the same connective dual cluster. Fine clustering can obtain the final connective dual clusters from coarse clusters. The connectedness of two clusters is defined in Definition 3.2, which describes whether they can be merged. The final results of connective dual clusters are obtained after recursively merging two connected clusters with the smallest distance between their centroids until the number of clusters is k. To minimize the average cost, use the agglomerative hierarchical clustering with the mean distance [9]. Specifically, if there are more than k disconnected subgraphs in the ConGraph, the number of clusters is also more than k due to the connected constraint. Therefore, there is no suitable solution in this situation.

Definition 3.2 (*Connectedness of clusters*) Two clusters C_i and C_j are connected if and only if $\exists o_t \in C_i$ and $\exists o_u \in C_j$ such that $dist_{\text{geo}}(o_t, o_u) \leq r$.

3.2.2 Time analysis

The agglomerative hierarchical clustering with average link requires $O(k'^3)$, where k' is the number of clusters found by coarse clustering. If sorted lists maintain the distances between each cluster to the other clusters, the time complexity can be reduced to $O(k'^2 \log k')$.

3.3 Overall time and space complexity

Given n objects, the number of clusters k, and the threshold of the connected constraint r, the transformation phase requires $O(n^2)$, while the coarse clustering phase requires $O(n \cdot k^2 + |E|)$. The agglomerative hierarchical clustering with the mean distance requires $O(k'^2 \log k')$, where k' is the number of coarse clusters and $k < k' \ll n$ generally. Thus, the overall time is bounded by the transformation time, $O(n^2)$. The space overheads are cells and graphs, which require $O(E)$ space.

4 Performance study

This section first describes the experimental environment, including the data generator, the suits of test cases, and the evaluation methods in Sect. 4.1. Then, Sect. 4.2 conducts a cost evaluation. All experiments were executed with a 2.4- GHz Intel CPU and 8 GB of memory.

4.1 Experiment settings

The simulation model in this study used the color of objects to represent their optimization attributes. The optimization domain is a three-dimensional attribute in a RGB color model. For example, value (255, 255, 0) in the optimization domain is shown by the color yellow. On the other hand, the geographic domain is the location of objects represented by X-axis and Y-axis. The following discussion uses (x, y) and (cr, cg, cb) to represent attributes in the geographic domain and attributes in the optimization domain, respectively.

4.1.1 Generation of synthetic data

A synthetic data generator was constructed to create ground truth test data. This generator required parameters k and r, where k indicates how many clusters should be generated, and objects with the same clusters are within distance r in the geographic domain. For each cluster, pivot points were generated uniformly from 0 to 799 first, and then, attributes in the geographic domain were generated to pass through these pivot points iteratively. The number of pivot points was generated randomly, and each pivot point has different x or y from the previous one. After the pivot points were generated, the first pivot point was the attribute of the first object, and the next value of x was $x + dir(x, x')w(t)$, where x' is the value of the next pivot point, $dir(x, x') = \frac{x'-x}{|x-x'|}$ and $w(t) = t$ with the probability $\frac{(t+r)}{2r^2}$, $t \in [-r, r]$ and t is an integer. The term $w(t)$ produces larger integers with higher probability, and $dir(x, x')$ is 1 or -1 according to the relative position between x and the destination x'. Therefore, the generated points tend toward to x' with higher probability, but they may be backward locations, too. y is also generated by the same function. Each (x, y) is the attribute in the geographic domain of a new object. After reaching the pivot point (x', y'), the next pivot point is used until no pivot points remain. This approach can generate random points in the form of many intersected "horizontal and vertical lines." To produce denser points, such that objects in each cluster have more neighbors, use the same pivot points again after all pivot points are passed. This approach may create a diagonal line between the last pivot and the first pivot.

After the objects in each cluster have their attributes in the geographic domain, the attributes in the optimization domain are generated according to the normal distribution $N(\mu, \sigma)$. The data generation in the optimization domain is similar to the method in the study by Lin et al. [16]. The terms cr, cg, $andcb$ are generated according to the same process and are therefore explained using only one attribute below. First, μ and σ are generated uniformly from 0 to 255 and from 1 to 32, respectively. Then, each object has the value from $N(\mu, \sigma)$. This value is replaced by 0 (or 255) if it is bellow 0 (or above 255). Therefore, objects in each cluster have similar attributes in the optimization domain, and all clusters obey the connected constraint.

Figure 6 shows an example of synthesized data with $r = 5$ and $k = 5$. In this figure, five clusters overlap with each other and each cluster forms a "closed line" in the geographic domain. Objects in the same cluster are represented by the same symbol and color. The data objects over two domains appear in Fig. 6a, b, which is seen as the ground truth. Figure 6c, d show the results of CLS over two domains. Only a few objects are in the wrong clusters. The following section presents detailed experiments to evaluate CLS.

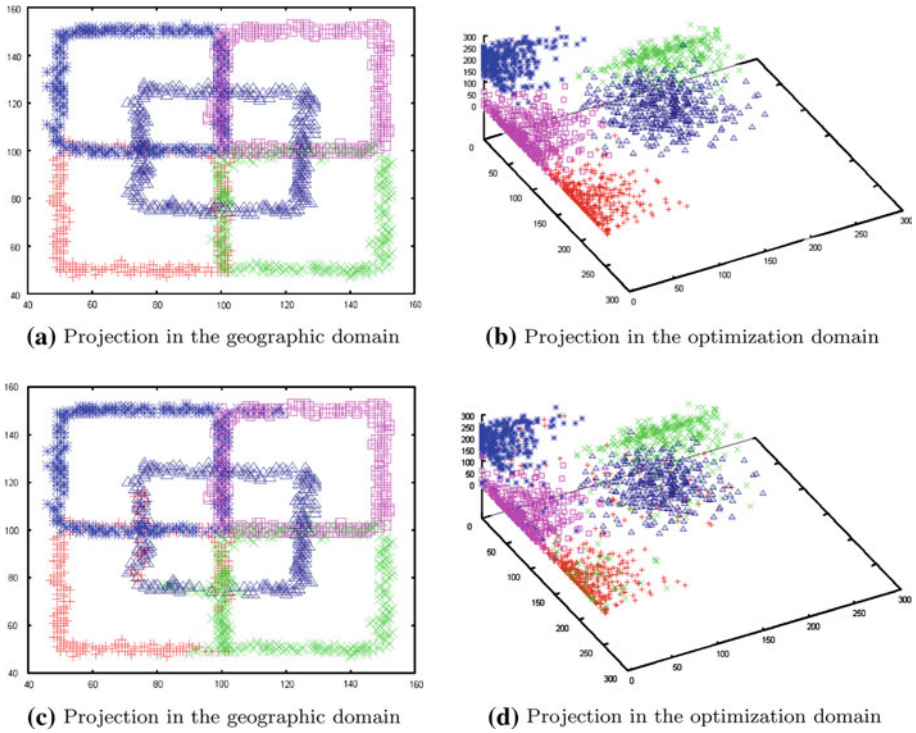
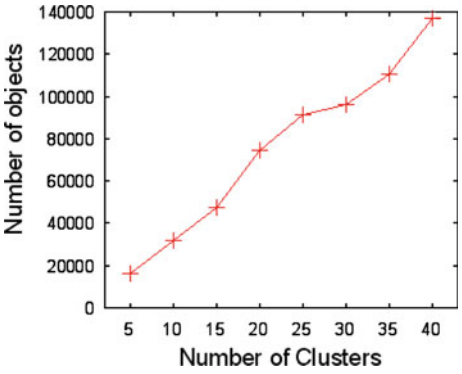


Fig. 6 An example of generated data with $r = k = 5$ and the result of CLS: (**a**) and (**b**) show the data over two domains; (**c**) and (**d**) show the result of CLS over two domains

Fig. 7 Setting of test suits



4.1.2 Test cases

To evaluate the scalability and efficiency of the proposed algorithm, this study analyzes 40 test cases with different numbers of data objects. These objects can be divided into eight suits according to their number of clusters, which varied from 5 to 40. Figure 7 shows the number of clusters and the average number of objects for each test suit. As mentioned in Sect. 4.1.1, the ranges of all attributes in the geographic domain and in the optimization domain are [0, 799] and [0, 255], respectively. Let the average distance between objects be

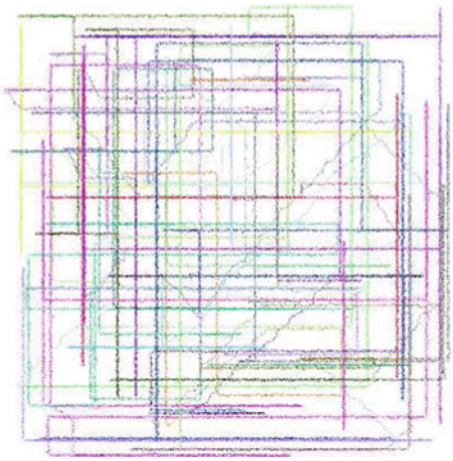


Fig. 8 An example of forty clusters

5 for all test cases to guarantee that the parameters r and k are known when evaluating the performance of each method. Therefore, the data set could be as complicated as Fig. 8, which shows 40 clusters and more than 1,30,000 data objects. Obviously, it is difficult to distinguish all clusters using both their color and location.

4.1.3 Competitor

To evaluate the enhancement of adopting connected constraint and local search, the experiments in this study compare the clusters discovered with BINGO [23] and two traditional clustering algorithms. Since the performance of ICC [16] is worse than that of BINGO, this study does not include ICC as a competitor. In addition, we modified two traditional clustering algorithms, k-means and Jarvis-Patrick clustering, to facilitate the problem of discovering connective dual clusters. The first modified algorithm is called Connected K-means (abbreviated as CK-means), and follows the two phases of CLS. Actually, CK-means adopts K-means instead of local search mechanism for coarse clustering. The second modified algorithm is called Connected Jarvis-Patrick Clustering (abbreviated as CJP). This method derives the coarse clusters by performing conventional Jarvis-Patrick clustering on the transformed ConGraph, where the weight of edges is defined as the distance on optimization domain. Finally, for both CLS and CK-means, the connective dual clusters are acquired by merging the coarse clusters.

4.1.4 Measurement

This study uses four measures to evaluate the accuracy of the discovered clusters: precision, recall, F-measure, and the relative cost. These four measurements are illustrated in detail below. Let $C_1, C_2, \ldots, C_{k'}$ be the clusters found by the algorithm under evaluation, and let $CT_1, CT_2, \ldots, CT_k$ be the true clusters according to the setting of the experiments. The precision is $P = \frac{\sum_{i=1}^{k'} |C_i / capCT_j|}{\sum_{i=1}^{k'} |C_i|}$. Similarly, the recall $R = \frac{\sum_{j=1}^{k} |C_i / capCT_j|}{\sum_{j=1}^{k} |CT_j|}$, and the F-measure [25] is $F = \frac{2PR}{P+R}$. Assuming that there are 1,00,000 objects, the correct result is 100 clusters and the largest cluster contains 2,000 objects. If each object forms unique clusters, then $P = 1.0$ and $R = \frac{100}{100000} = 0.001$. Therefore, $F = 0.002$. On the other hand, if there is only one cluster that contains all objects, then $P = \frac{2000}{100000} = 0.02$ and $R = 1.0$. Therefore,

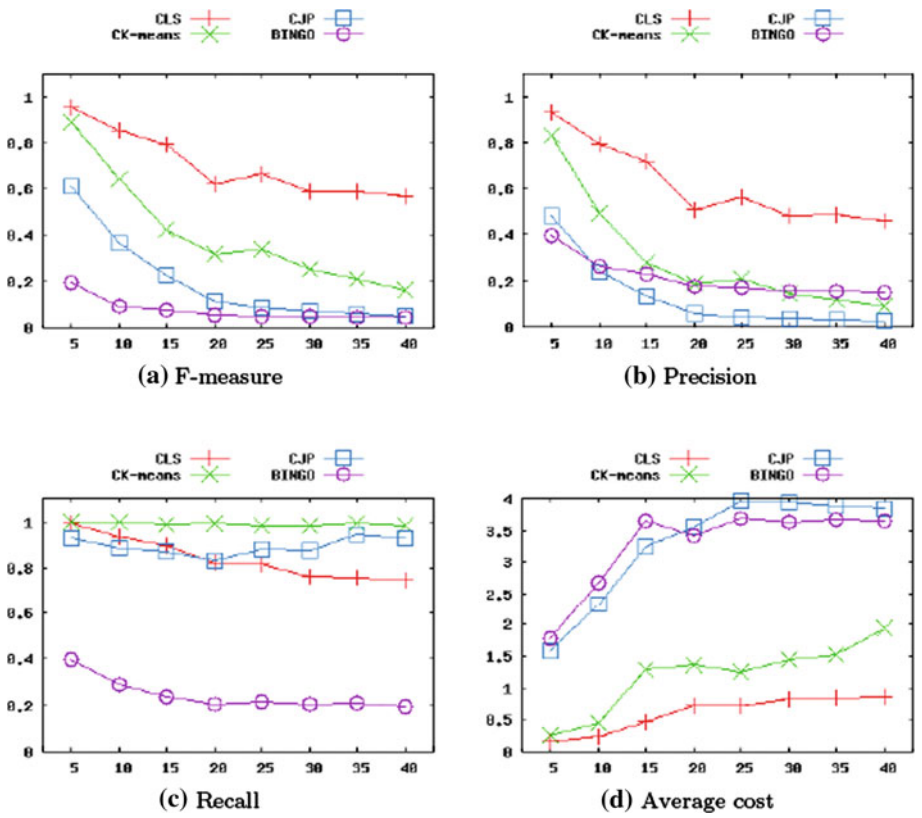


Fig. 9 Overall results of CK-means, CLS, CJP, and BINGO

$F = 0.04$. The last measurement is the relative cost of clusters, which compares the cluster cost SC with the ground truth SC_{true}. The relative cost is $\frac{f(SC)-f(SC_{true})}{f(SC_{true})}$.

4.2 Performance evaluation

This section reports the results of several experiments with three goals in mind. The first one is to evaluate the correctness of the discovered clusters. The second one is to evaluate the effectiveness of fine clustering. The CLS, CK-means, and CJP algorithms, which consist of fine clustering, are compared for their discovered clusters after fine clustering. The third one is to evaluate the accuracy of initial seed selection method. Experimental results show that the proposed method can discover initial seeds that are as good as the average accuracy of random selection. Thus, it saves time and increases accuracy in selecting initial seeds.

4.2.1 Overall comparison

Figure 9 shows the results of the experiments with our algorithms, BINGO and the two modified traditional approaches. According to Fig. 9a, b, CLS performs better than BINGO, CK-means, and CJP, especially when the number of clusters increases. Although BINGO can achieve higher accuracy than CK-means and CJP for larger data sets, it still performs worse than the proposed CLS algorithm. On the other hand, CLS achieves 50% accuracy

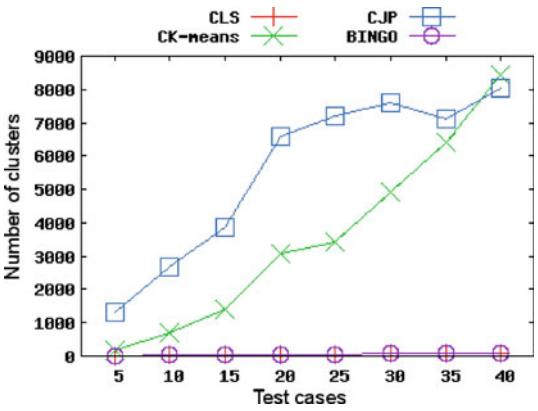


Fig. 10 Number of clusters found

Table 3 Summary of the quality of the coarse and fine clustering

Algorithms	Avg. F	Avg. P	Avg. R	Avg. cost
CLS w/o fine	0.673	0.778	0.598	36.864
CLS	0.705	0.618	0.841	38.967
CK-means w/o fine	0.756	0.806	0.720	22.042
CK-means	0.404	0.294	0.990	53.174
CJP w/o fine	0.381	0.465	0.427	55.615
CJP	0.197	0.132	0.893	102.365

on all of the 40 test cases. In addition, the recall evaluation is depicted in Fig. 9c, where the result of BINGO is much worse than other methods. Although CLS did not have higher recall values than CK-means and CJP in all test cases, it still outperforms other methods in evaluating the F measure score. Figure 9d compares the average relative cost with the true average cost. Finally, Fig. 10 shows the number clusters discovered by each method. Both CLS and BINGO can find the actual number of clusters, while CK-means and CJP find out many more clusters than the ground truth. This is because k-means and JP clustering cannot cluster the data objects that have similar attributes in the optimization domain but dissimilar attributes in the geographic domain, breaking the ConGraph into fragments.

4.2.2 Comparison between coarse and fine clustering

This section evaluates the effectiveness of fine clustering. The idea of fine clustering is to combine two similar clusters that have good precision into a new larger cluster so that it can maintain a high precision and a higher recall. Table 3 shows the average F measure score, precision, recall, and cluster cost for each method, where "CLS w/o fine," "CK-means w/o fine," and "CJP w/o fine" represent the clusters found by CLS, CK-means, and CJP before fine clustering. The fine clustering could improve the recall value but decrease the precision of each method. For CLS, the decrease in precision is insignificant that the F measure score increases after fine clustering. Although fine clustering can improve the recall of CK-means to 0.99, the F measure score and precision become much worse.

4.2.3 Initial seed evaluation

To determine the quality of the proposed method in choosing the initial seeds from ConGraph, this study performed CLS with random seeds 10 times for each test case and compared the

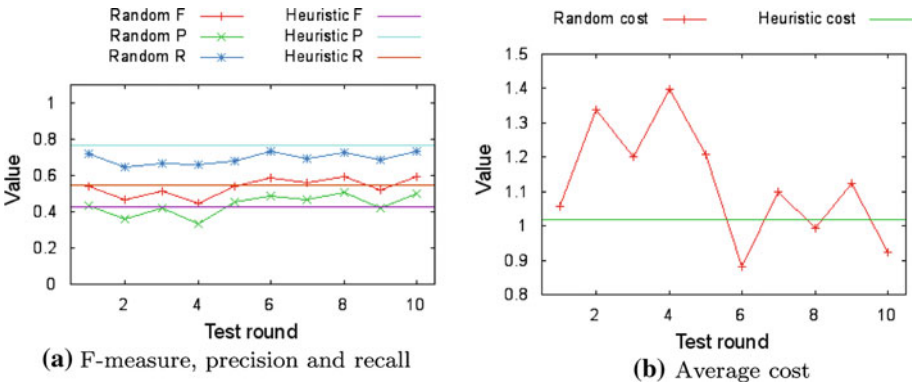


Fig. 11 Detailed results between the proposed seed selection and random selection in test case 38

Table 4 Result of test case 38: comparison between random and proposed seeds selection in CLS algorithm

Seeds selection methods	Avg. F	Avg. P	Avg. R	Avg. cost
Random	0.536	0.438	0.694	1.123
CLS	0.548	0.426	0.770	1.02

results with the CLS with the proposed seed selection methods. Figure 11 shows the result of test case 38 with $k = 40$, which has largest number of objects and clusters, and the average F-measures of random and proposed seed selection are most similar. Figure 11 represents the results of the proposed method as horizontal lines, showing that the results of random seeds would perform better or worse than the proposed method due to different random results. In addition, Table 4 depicts the average values of F measure, precision, recall, and relative cost for both proposed and random seed methods. As Table 4 shows, the results of all values are highly similar, and the difference between the average F measure of eight test suits is only 0.003.

From this experiment, although random seeds outperform the proposed initial seeds in some cases, they often cause very poor results. To achieve similar results with the proposed method, the random seed selection method must try different random results and thus wast time on computing and searching suitable initial seeds. On the other hand, proposed seed selection can generally achieve stable and good results. For large scale data, the usefulness of proposed seed selection becomes more significant since the time cost of one execution is very expensive.

5 Conclusion

This paper presents a dual clustering problem with an explicit geographic constraint, which is a general dual clustering problem. The proposed CLS algorithm consists of two phases: the ConGraph transformation phase and the clustering phase. This algorithm derives a set of clusters in which objects in the same cluster have similar values in the optimization domain and are connected. The ConGraph transformation phase, according to the geographic attributes, builds a ConGraph to capture connectivity relationships. The CLS algorithm also explores cells to construct the ConGraph efficiently and select the initial seeds for clustering. Using the ConGraph, the CLS algorithm could find the coarse clustering result by searching the local nearest data object. Using the advantage of local search, the coarse clusters are generated

efficiently without checking all connections between cluster members and unclustered data objects. Then, fine clustering results are obtained by combining the coarse clusters that satisfy the connected constraint. To prove that the proposed CLS algorithm is efficient, this study presents a complexity analysis of the overall CLS algorithm and each of its phases. Experimental results show that CLS can discover more natural clusters than other methods. Moreover, the efficiency evaluation results indicate that CLS is scalable and efficient.

Acknowledgments Wen-Chih Peng was supported in part by the National Science Council, Project No. 97-2221-E-009-053-MY3, by Taiwan MoE ATU Program, by ITRI-JRC, Project No. 100-EC-17-A-05-01-0626, by D-Link and by Microsoft.

References

1. Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. J Mach Learn Res 2:125–137
2. Bentley JL, Friedman J H (1979) Data structures for range searching. ACM Comput Surv 11(4):397–409
3. Bradley PS, Fayyad UM (1998) Refining initial points for k-means clustering. In: Proceedings of the 15th international conference on machine learning (ICML). Madison, Wisconson USA, July 24–27 1998, pp 91–99
4. Chang Y-M, Wei L-Y, Lin C-S, Jung C-H, Chen I-H, Peng W-C (2009) Exploring gps data for traffic status estimation. In: Proceeding of the 10th international conference on mobile data management (MDM). Taipei, Taiwan, 18–20 May 2009, pp 369–370
5. Chaoji V, Hasan MA, Salem S, Zaki MJ (2010) Sparcl: an effective and efficient algorithm for mining arbitrary shape-based clusters. Knowl Inf Syst 21(2):201–229
6. Ester M, Ge R, Gao BJ, Hu Z, Ben-Moshe B (2006) Joint cluster analysis of attribute data and relationship data: the connected k-center problem. In: Proceedings of the 6th SIAM international conference on data mining (SDM), Bethesda, MD, USA, April 20–22
7. Ge R, Ester M, Gao BJ, Hu Z, Bhattacharya BK, Ben-Moshe B (2006) Joint cluster analysis of attribute data and relationship data: the connected k-center problem, algorithms and applications. TKDD 2(2):7:1–7:35
8. Grabmeier J, Rudolph A (2002) Techniques of cluster algorithms in data mining. Data Min Knowl Discov 6(4):303–360
9. Han J, Kamber M (2000) Data mining: concepts and techniques. Morgan Kaufmann, Los Altos
10. Han J, Li Z, Tang LA (2010) Mining moving object, trajectory and traffic data. In: Proceedings of the 15th international conference on database systems for advanced applications (DASFAA). Tsukuba, Japan, April 1–4, 2010, pp 485–486
11. Hasan MA, Salem S, Zaki, MJ (2010) Simclus: an effective algorithm for clustering with a lower bound on similarity. Knowl Inf Syst 1–21
12. Kaufman L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis. Wiley J, New York
13. King B (1967) Step-wise clustering procedures. J Am Stat Assoc 62(317):86–101
14. Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma W-Y (2008) Mining user similarity based on location history. In: Proceeding of the 16th ACM SIGSPATIAL international symposium on advances in geographic information systems (ACM-GIS). Irvine, California, USA, November 5–7, 2008, p 34
15. Li X, Li Z, Han J, Lee J-G (2009) Temporal outlier detection in vehicle traffic data. In: Proceedings of the 25th international conference on data engineering (ICDE). Shanghai, China, March 29 2009–April 2 2009, pp 1319–1322
16. Lin CR, Liu KH, Chen MS (2005) Dual clustering: integrating data clustering over optimization and constraint domains. IEEE Trans Knowl Data Eng 17:628–637
17. Lin S, Arai B, Gunopulos D, Das G (2008) Region sampling: Continuous adaptive sampling on sensor networks. In: Proceedings of the 24th international conference on data engineering (ICDE). Cancun, Mexico, April 7–12, 2008, pp 794–803
18. Lo C-H, Peng W-C (2008) Efficient joint clustering algorithms in optimization and geography domains. In Proceedings of the 12th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD), Osaka, Japan, May 20–23, 2008, pp 945–950
19. Lo C-H, Peng W-C, Chen C-W, Lin T-Y, Lin C-S (2008) Carweb: A traffic data collection platform. In: Proceedings of the 9th international conference on mobile data management (MDM). Beijing, China, April 27–30, 2008, pp 221–222

20. Moser F, Ge R, Ester M (2007) Joint cluster analysis of attribute and relationship data without a-priori specification of the number of clusters. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (SIGKDD). San Jose, California, USA, August 12–15, 2007, pp 510–519
21. Ng RT, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of 20th international conference on very large data bases (VLDB). Santiago de Chile, Chile, September 12–15, 1994, pp 144–155
22. Russell SJ, Norvig P (2002) Articial intelligence: a modern approach, 2nd edn. Prentice Hall, Englewood Cliffs
23. Tai C-H, Dai B-R, Chen M-S (2007) Incremental clustering in geography and optimization spaces. In: Proceedings of the 11th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD). Nanjing, China, May 22–25, 2007, pp 272–283
24. Takacs B, Demiris Y (2010) Spectral clustering in multi-agent systems. Knowl Inf Syst 25(3):607–622
25. Tan PN, Steinbach M, Kumar V (2006) Introduction to data mining. Addison Wesley, Boston
26. Tang M, Zhou Y, Li J, Wang W, Cui P, Hou Y, Luo Z, Li J, Lei F, Yan B (2010) Exploring the wild birds migration data for the disease spread study of h5n1: a clustering and association approach. Knowl Inf Syst
27. Wang F, Ding CHQ, Li T (2009) Integrated kl (k-means - laplacian) clustering: a new clustering approach by combining attribute data and pairwise relations. In: Proceedings of the SIAM international conference on data mining (SDM). Sparks, Nevada, USA, April 30–May 2, 2009, pp 38–48
28. Yoo JS, Shekhar S, Kim S, Celik M (2006) Discovery of co-evoluting spatial co-located event sets. In: Proceedings of the 6th SIAM international conference on data mining (SDM). Bethesda, MD, USA, April 20–22, 2006

Author Biographies

Zhung-Xun Liao received the BS degree from the National Taiwan Normal University, in 2002, and the MS degree in Computer Science from the National Chengchi University, Taiwan, in 2004. Currently, he is a Ph.D. candidate at the department of Computer Science, National Chiao Tung University, Taiwan. His research interests include data mining and databases.

Wen-Chih Peng was born in Hsinchu, Taiwan, R.O.C. in 1973. He received the BS and MS degrees from the National Chiao Tung University, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree in Electrical Engineering from the National Taiwan University, Taiwan, R.O.C. in 2001. Currently, he is an assistant professor at the department of Computer Science, National Chiao Tung University, Taiwan. Prior to joining the department of Computer Science and Information Engineering, National Chiao Tung University, he was mainly involved in the projects related to mobile computing, data broadcasting, and network data management. Dr. Peng serves as PC members in several prestigious conferences, such as IEEE International Conference on Data Engineering (ICDE), Pacic Asia Knowledge Discovering and Mining (PAKDD), and Mobile Data Management (MDM). His research interests include mobile computing, network data management, and data mining. He is a member of IEEE.