Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 828374, 14 pages
http://dx.doi.org/10.1155/2015/828374

Research Article

A CDT-Based Heuristic Zone Design Approach for Economic Census Investigators

Changixu Cheng,[1,2] Xiaomei Song,[2,3] Jing Yang,[1] Xiatian Hu,[2] Shi Shen,[1] and Lijun Wang[4]

[1]*ADREM, Beijing Normal University, Beijing 100875, China*
[2]*LREIS, Institute of Geographic Science and Resources Research, Beijing 100101, China*
[3]*School of Software, Tsinghua University, Beijing 100084, China*
[4]*China Internet Network Information Center, Beijing 100101, China*

Correspondence should be addressed to Changixu Cheng; chengcx@bnu.edu.cn

Received 30 July 2015; Accepted 19 October 2015

Academic Editor: Xiaobo Qu

This paper addresses a special zone design problem for economic census investigators that is motivated by a real-world application. This paper presented a heuristic multikernel growth approach via Constrained Delaunay Triangulation (CDT). This approach not only solved the barriers problem but also dealt with the polygon data in zoning procedure. In addition, it uses a new heuristic method to speed up the zoning process greatly on the premise of the required quality of zoning. At last, two special instances for economic census were performed, highlighting the performance of this approach.

1. Introduction

Zone design is widely used as a method of spatial partitioning in geospatial sciences. It is similar to districting problem, which is a classical topic in optimization research. Both of the zone design and the districting problem use a certain of optimize operations divide a geographical region into some zones (or districts). The difference between them is to meet different criteria or constraints, such as balance, compactness, and contiguity. However, zone design pays more attention to how to build a spatial auxiliary graph. Sometimes, the zone design, also called district design or territory design, is usually considered a strategic activity [1]. The zone design has a broad range of applications, such as in political redistricting [2–7], sales territory alignment [8–10], school redistricting [11], logistics districting [12–15], and delimitating zones for land-use allocation and/or land acquisition and apportionment [16–18].

Traditional methods for zone design mainly focus on the complex geometries and spatial relationships between them. However, there are still some problems to be solved, such as nodes representation problem and geometrical barriers problem. Solving these problems from a GIScience perspective may lead to a tractable solution, highlighting the importance of geographical insights [19, 20]. The use of ordinary and nonordinary Voronoi diagrams to solve districting problems has been reported in many literatures [21–23]. Novaes et al. [22] took advantage of the Voronoi diagram to detect the spatial relationship and solved geometrical barriers problem successfully. However, all above approaches are based on point data but are not suitable for polygon data. Murray and Tong [19] have raised the issue that only point-based representations are too simplistic to represent more complex vector objects like polygons in GIS. On the other hand, traditional methods make the shape of zones tend to be a circle or a square [21, 24]. That no longer applies with the presence of geometrical barriers problem. Another remarkable feature of traditional districting methods is that running procedures for optimizing the districting results usually cost too much time, which is in seconds or minutes [25–27].

So it is hard to use the previous studies or existing optimization software platform (such as CPLEX or MOSEK) to find an effective solution. To solve problems in this paper, we have to find a way different from the traditional ones. This paper presented a heuristic multikernel growth approach via Constrained Delaunay Triangulation (CDT) and introduced

how to divide work zones for economic census investigators. Section 2 reviewed some relative work. Section 3 detailed how to construct an auxiliary graph for polygon data and barrier data and how to heuristically grow and generate spatial partitions. Section 4 gave some test to show functionality and performance of this method. Section 5 made some conclusions and some further work.

2. Problems Definition

This paper discussed a special zoning problem for economic census application. The objective of the problem is to partition a region into some census zones for investigators. Every investigator takes charge of surveying the financial situation of all companies in one zone. The partitioned zones should be contiguous, compact, and balanced. The main constraints presented are as follows: (1) contiguity makes sure that units in every zone are geographically adjacent; (2) compactness requires total travelling time in one zone as small as possible; (3) balance ensures that the workloads of every investigator are as equal as possible. The workload mainly includes investigating time and travelling time. In our application, data sources are polygon data and line obstacle data. Polygon data stands for the shape of buildings, where companies located. Some polygons are touching, but some of them are not touching, while line obstacle data stands for fences or busy roads which cannot be passed by. Additionally, it is difficult for ordinary users to give some upper and lower limits on a scalar attribute (e.g., investigating time and travelling time), because they hope to use it as simple as possible. To solve problems in this paper, we have to find a way different from the traditional ones.

First problem is about how to generate auxiliary graph for these touching or nontouching building polygons. Most of literatures often use a point to represent a polygon and then generate auxiliary graph based on these simplified points. However, in most of cases, the auxiliary graph mentioned above may not truly represent buildings' connection and touching relationships, because sometimes a polygon with complex shape may distort the connection and touching relationships of neighbor polygons. The partitions based on simple point-representation may result in some errors. Although Chou and Li [28, 29] and Ríos-Mercado and Fernández [27] discussed how to generate auxiliary graph directly based on polygons, they only considered touching polygons rather than nontouching polygons. Therefore, it is necessary to generate an auxiliary graph based on touching or nontouching polygons. In this paper, we used Constrained Delaunay Triangulation (CDT) to generate an *Auxiliary Graph for nontouching Polygons* (AGP). The shortest distance between discrete polygons in the auxiliary graph is related to the shapes of touching polygons, which would be described in Section 3.1.

The second problem is how to adjust AGP when line obstacles are involved. Novaes et al. [22] discussed some geometrical barriers problem, but they still focused on point data based on Voronoi diagram. Their method is not suitable for our polygon data based on CDT. In our application, geographical barriers may be polylines or polygons; they will affect the connection of building polygons. When a barrier crosses some edges of an auxiliary graph, these edges (paths) will be cut. The adjusted graph is called *auxiliary graph for polygons and barriers* (AGPB).

The third problem is how to speed up the zoning process. In fact, spatial partition is a classical NP-hard problem. Most of literatures adopt the multikernel growth approach. The approach firstly selects a certain number of basic nodes as "seeds" (centers) for some zones and then successively adds every other node to its neighboring center until all have been assigned [23]. In the optimization phase, they need to constantly swap two adjacent nodes of zones in order to achieve optimized zones. When it comes to too many nodes, it is exhausted. It is feasible to consider at heuristics [6, 8, 30, 31], such as Tabu Search [32, 33] and simulated annealing [17, 34, 35]. The problem of traditional solutions is that bad seeds selection would generate bad partitions initiation. That means they often took more time to get optimization partitions. The procedure usually spends a lot of time after 20,000~80,000 iterations [7]. In the paper, we use some heuristic strategies about spatial structure information to participate in the location of nodes and then avoided some unnecessary iterations.

3. A Heuristic Zone Design Approach Based on Constrained Delaunay Triangulation

There are two major phases in our method. Phase 1 is to construct an auxiliary graph for polygons and barriers and at last write them into XML files. Phase 2 is to use a spatial heuristic strategy to realize multikernel growth. The heuristic strategy continues to run as a simple optimization after all nodes are allocated. A flow chart of the general procedure is presented in Figure 1. Details of the steps are provided below.

3.1. Phase I: Auxiliary Graph Construction. The left part of Figure 1 shows the flow chart of Phase I. The AGPB construction is an optional operation. Users can input polygon data and barriers data based on actual situations. Finally, an auxiliary graph, which reflects the connectivity between nodes, is written into XML files.

3.1.1. Construction of AGP. When polygon data is input, topology checks and polygon-touching checks should be made first of all. This can avoid polygons being overlapped and can record nodes relation of touching. Then the AGP construction should be made and it can be depicted as follows: Firstly, DT is generated based on the vertexes of polygons and then is adjusted as Constrained Delaunay Triangulation (CDT) by counter lines of polygons. Secondly, all the edges contained in the polygons are deleted. Thirdly, shortest paths between the polygons are generated based on these triangulations. The construction of AGP is generated on CDT because a graph (DT) needs to be constructed by using the vertexes of the polygon; in most cases, the counter line of a polygon will intersect with the edges of DT. The special CDT is generated as follows: counter lines of polygons are seen as barrier lines and also as edges of triangulations that are forcibly inserted. Then, triangulations in polygons

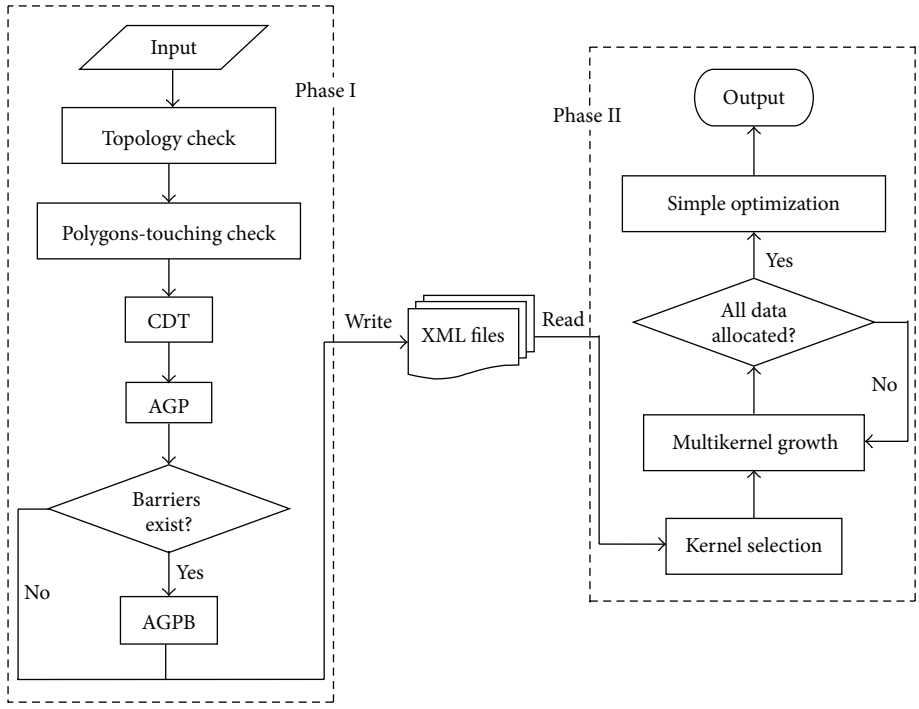


Figure 1: Diagram of general procedure flow.

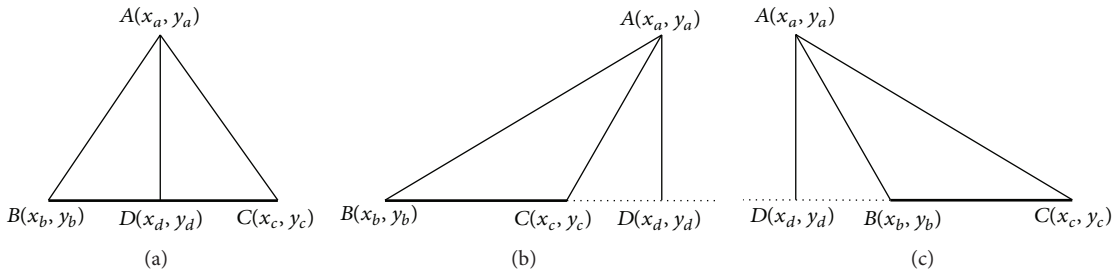


Figure 2: Search the short path.

need to be deleted. After the deletion of all the triangulations contained in the polygons, the polygons are attached in the graph as nodes. The relationship between adjacent nodes can be obtained by recursions.

In the third step above, the shortest paths between polygons are obtained by taking advantage of Delaunay Triangles. In fact, after the second step of CDT construction, triangles must connect two or three polygons. If three points of a triangle belong to three different polygons, its three edges are all likely to be the shortest paths between two polygons. Therefore, all edges in this type of triangles should be stored as candidates for the shortest path. If points of a triangulation belong to two polygons, it must be such a situation that a point belongs to a polygon and the other two points belong to another polygon. In those cases, the shortest path may not be the one with the edges existing in the triangle but rather the vertical distance from a point to a line. In Figure 2, we assume that there is a Delaunay Triangle connected with two polygons, and vertex $A(x_a, y_a)$ belongs to one triangle and vertexes $B(x_b, y_b)$ and $C(x_c, y_c)$ belong to the other one. Assume that the edge AD stands for the shortest edge which is from vertex A to the straight line determined by vertexes B and C. The location of vertex D has two situations: either location inside the edge BC or location outside the edge BC. We assume that the slope of straight line BC is k. If point D moves from vertex B along x-axis by dx units, it will move along y-axis by $dy = x_b + k * dx$ units. Straight line AD is perpendicular to straight line BC and we obtain the formula as follows:

$$\frac{y_a - y_d}{x_a - x_d} = \frac{y_a - (y_b + k * dx)}{x_a - (x_b + dx)} = -\frac{1}{k}. \tag{1}$$

Equation (1) can be further rearranged as follows:

$$dx = \frac{k * (y_a - y_b) + x_a - x_b}{1 + k^2}. \tag{2}$$

If $0 \le dx \le x_c - x_b$, vertex D is located between vertex B and vertex C and the shortest path in the triangulation is the segment AD.

If $dx > x_c - x_b$, vertex D is located in the extension line of segment BC and the shortest path in the triangulation is the segment AC.

If $dx < 0$, vertex D is located in the extension line of segment CB and the shortest path in the triangulation is the segment AB.

3.1.2. Construction of AGPB. Further, the existing graph should be adjusted when geographical barriers exist. After the construction of AGP, we continue to put the vertexes of geometrical barriers into the graph and construct AGPB. In the process above we should find out all the edges intersecting with counter lines of geometrical barriers. Assume that the edge with the start point A and end point B needs to be adjusted. According to the graph, the geometrical barriers associated with points A and B can be found out and the set of their vertexes is assumed to be noted as A_{aset} and B_{aset}, respectively.

Assume that the line with start point A and end point B is defined as Line$\langle A, X, B\rangle$ and X stands for a point or a line or the collection of both. For example, the Line$\langle C, D, E\rangle$ means points C, D, and E form a line; the line of Line$\langle A, \text{Line}\langle C, D, E\rangle, B\rangle$ is composed of points A, C, D, E, and B. Suppose $\mathbf{O}$ (defined as in (3)) is the set of barriers and is defined as follows where the integer c means the count of geometrical barriers. Consider

$$\mathbf{O} = \{O_1, O_2, \ldots, O_c\} \quad (1 < c < \infty). \tag{3}$$

O_i $(i = 1, \ldots, c)$ represents a polygon or a line. Assume that the line with the minimum length in a set L of lines is noted as $\text{Line}_{\text{minlength}}\{L\}$. The shortest path function (noted as Iter_Line$(\cdot)$) of points A and B can be calculated as follows:

$$\begin{aligned}\text{Iter\_Line}(A, X, B) = \text{Line}_{\text{minlength}}\{&\text{Line\_}\alpha\langle A, X, B\rangle,\\ &\text{Line\_}\beta\langle A, X, B\rangle, \text{Line\_}\gamma\langle A, X, B\rangle,\\ &\text{Line\_}\delta\langle A, X, B\rangle\}.\end{aligned} \tag{4}$$

As can be seen from the formula, the shortest path is calculated by considering four cases. Calculations of four cases are as follows:

(1) The shortest path is composed of three points, which are point A, point X from A_{aset}, and point B. Mathematical expression is as follows:

$$\begin{aligned}\text{Line\_}\alpha\langle A, X, B\rangle = \text{Line}_{\text{minlength}}\Bigg\{&\text{Line}\langle A, X, B\rangle\\ &\in R^2 \mid \text{Line}\langle A, X, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right) = \emptyset,\ X \in A_{\text{aset}}\Bigg\}.\end{aligned} \tag{5}$$

And we set a set α as follows:

$$\alpha = \left\{X \in A_{\text{aset}} \mid \text{Line}\langle A, X, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right) = \emptyset\right\}. \tag{6}$$

(2) The shortest path is composed of three points, which are point A, point X from B_{aset}, and point B. Mathematical expression is as follows:

$$\begin{aligned}\text{Line\_}\beta\langle A, X, B\rangle = \text{Line}_{\text{minlength}}\Bigg\{&\text{Line}\langle A, X, B\rangle\\ &\in R^2 \mid \text{Line}\langle A, X, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right) = \emptyset,\ X \in B_{\text{aset}}\Bigg\}.\end{aligned} \tag{7}$$

And we set a set β as follows:

$$\beta = \left\{X \in B_{\text{aset}} \mid \text{Line}\langle A, X, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right) = \emptyset\right\}. \tag{8}$$

(3) The shortest path is composed of four points, which are point A, point K from $(A_{\text{aset}}\text{-}\alpha)$, point L from $(B_{\text{aset}}\text{-}\beta)$, and point B. Mathematical expression is as follows:

$$\begin{aligned}\text{Line\_}\gamma\langle A, X, B\rangle = \text{Line}_{\text{minlength}}\Bigg\{&\text{Line}\langle A, K, L, B\rangle\\ &\in R^2 \mid \text{Line}\langle A, X_1, X_2, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right) = \emptyset,\ K\\ &\in (A_{\text{aset}}\text{-}\alpha),\ L \in (B_{\text{aset}}\text{-}\beta)\Bigg\}.\end{aligned} \tag{9}$$

And we set a set γ as follows:

$$\begin{aligned}\gamma = \Bigg\{&X_1 \in (A_{\text{aset}}\text{-}\alpha),\ X_2\\ &\in (B_{\text{aset}}\text{-}\beta) \mid \text{Line}\langle A, X_1, X_2, B\rangle \cap \left(\bigcup_{j=1}^{n} O_j\right)\\ &= \emptyset\Bigg\}.\end{aligned} \tag{10}$$

(4) An iterative process will be taken into account in this case. One point in the $(A_{\text{aset}}\text{-}\alpha\text{-}\gamma)$ and one point in the $(B_{\text{aset}}\text{-}\beta\text{-}\gamma)$ are put into the function (I) and a set of lines (noted as *sublines*) can be obtained. Finally select the shortest one from *sublines*, put points A to B into the first and last position of the shortest subline, and the shortest path would be found out. The line on

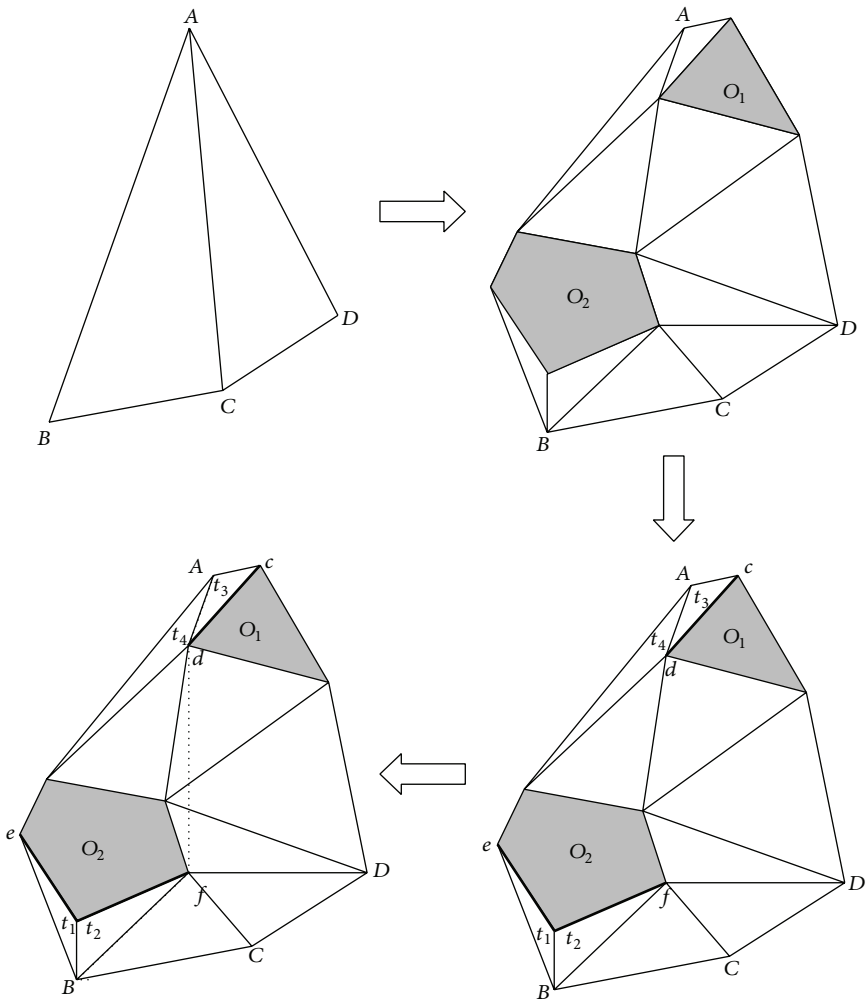


Figure 3: Search the short path around barriers.

behalf of the shortest path is composed of not less than four points. Mathematical expression is as follows:

$$\begin{aligned}
&\text{Line_}\delta\ \langle A, X, B\rangle \\
&= \text{Line}_{\text{minlength}} \left\{ \text{Line}\,\langle A, \text{Line}\,\langle M, Y, N\rangle, B\rangle \right. \\
&\left. \in R^2 \mid \text{Line}\,\langle A, \text{Iter_Line}\,(A, Y, B), B\rangle \cap \left(\bigcup_{j=1}^{n} O_j \right) \right. \\
&\left. = \emptyset,\ M \in \left(A_{\text{aset}}\text{-}\alpha\text{-}\gamma\right),\ N \in \left(B_{\text{aset}}\text{-}\beta\text{-}\gamma\right) \right\}.
\end{aligned} \tag{11}$$

The Antonio example is used here to illuminate the new method. Firstly the Delaunay Triangulation is created based on points A, B, C, and D. Add vertexes of barriers O_1 and O_2 to adjust the DT. The DT after adjustment is shown in Figure 3. According to DT, triangles t_1, t_2, t_3, and t_4 are found out. So $A_{\text{aset}} = \{c, d\}$ and $B_{\text{aset}} = \{e, f\}$, where c, d, e, and f represent vertexes of barriers shown in the figure. Line_$\alpha\langle A, X, B\rangle$, Line_$\beta\langle A, X, B\rangle$, and Line_$\delta\langle A, X, B\rangle$ do not exist, while Line_$\gamma\langle A, X, B\rangle$ = Line$\langle A, d, f, B\rangle$. So the shortest path from point A to point B is $A \to d \to f \to B$.

Phase I mainly accomplishes operations for determining the spatial relations of data. It is relatively independent of Phase II. The graph obtained from Phase I can be stored and used at any time in Phase II with different parameters. Therefore, the design of two phases can save significant amount of time during graph construction.

3.2. Phase II: The Heuristic Multikernel Growth

3.2.1. Problem Formulation. Before describing the heuristic multikernel growth, it is necessary to formally specify the problem formulation of interest. The problem formulation can be specified from the follow three criteria: balance of workloads, compactness of zones, and contiguity of nodes.

The following parameters are defined:

N = node set.

E = edge set.

i = index of nodes.

k = index of nodes.

j = index of potential zones.

m = number of zones to plan.

n_i = node i.

w_i = workload of the node i.

N_j = node set of the zone j.

d_{ik} = Euclidean distance between nodes i and k.

e_{ik} = edge connecting nodes i and k.

MST_j = minimum spanning tree of the potential zone j.

α = weight of compactness.

The following decision and auxiliary variables are defined:

$$a_{i\to j} = \begin{cases} 1, & \text{if node } i \text{ blongs to the district } j \\ 0, & \text{otherwise,} \end{cases}$$

$$b_{ik\to j} = \begin{cases} 1, & \text{if } e_{ik} \text{ blongs to the edge set of } \mathrm{MST}_j \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

$$e_{i\leftrightarrow k} = \begin{cases} 1, & \text{if node } i \text{ and } k \text{ are connected by an existing edge } e_{ik} \\ 0, & \text{otherwise.} \end{cases}$$

The mathematical formulation is as follows:

Balance

$$\text{Minimize}\left(\max\left\{W_1,\ldots,W_j,\ldots,W_m\right\} - \min\left\{W_1,\ldots,W_j,\ldots,W_m\right\}\right), \tag{13}$$

where

$$W_j = \sum_{n_i\in N_j}\left(w_j * a_{i\to j}\right) + \alpha * D_j \quad 1 \le j \le m. \tag{14}$$

Compactness

$$\text{Minimize}\left(\max\left\{D_1,\ldots,D_j,\ldots,D_m\right\}\right), \tag{15}$$

where

$$D_j = \sum_{n_i\in N_j}\sum_{n_k\in N_j}\left(d_{ik} * a_{i\to j} * a_{k\to j} * b_{ik\to j}\right) \quad 1 \le j \le m. \tag{16}$$

Contiguity. If a zone is an undirected graph whose nodes stand for buildings of the zone, an edge e_{ik} exists between nodes i and k if and only if $e_{i\leftrightarrow k} = 1$. A zone is contiguous if and only if the graph is connected (a path exists between every pair of adjacent nodes). A zone is feasible only if it is contiguous. Constraint (17) means a node belongs to only one zone while constraint (18) means a zone contains more than one node:

$$\sum_j a_{i\to j} = 1, \quad n_i \in N \tag{17}$$

$$\sum_{n_i\in N_j}\sum_{n_k\in N_j} e_{i\leftrightarrow k} \ge 1. \tag{18}$$

The formulations above are drove by a specific zone problem in the real-world application. The problem is different from the traditional ones because we cannot give out the exact bounds for varieties or constraints. So we cannot completely use the methods of previous work or the existing optimization software platform (such as CPLEX or MOSEK) to solve the problem. In this work, multikernel growth is employed as a simple heuristic partition method. This method begins by selecting a certain number of basic nodes as "kernels" for the zones. The algorithm then successively adds, to each kernel, neighboring basic nodes by primarily considering the workload balance and compactness objectives.

3.2.2. A Heuristic Strategy. The specific multiobjective problem is solved in this work by the heuristic multikernel growth method (AMKG for short) based on an auxiliary graph. The contiguity objective is assumed to be achieved as long as nodes are connected by edges in an auxiliary graph. The balance and compactness objectives are achieved by a heuristic strategy: the zone with the smallest workload is selected and allocated with a specific basic node which contains a relative large workload and is relatively close to the zones. Furthermore, the specific basic node is found out firstly from unallocated nodes around the zone otherwise, if it failed to be got, from allocated nodes around the zone. The specific basic node is selected by calculating the "cost" which is defined by (19). To achieve the workload balance and compactness objective, "kernels" are greedily allocated basic nodes with the largest "cost" via an iterative procedure. The "cost" of node i can be defined as follows:

$$\text{cost}(i,j) = w_i + \beta * \frac{w_i}{D_{ij}} * \frac{\overline{D}}{\overline{W}}, \tag{19}$$

where β stands for a positive number of type double, $\overline{D}$ means the average distance value of the edges in graph network, $\overline{W}$ means the average value of workload, and $D_{ij} = \min_{n_i\in N_j}(d_{ik} * a_{i\mapsto j} * a_{k\to j})$,

$$a_{i\mapsto j} = \begin{cases} 0, & \text{if node } i \text{ blongs to district } j \\ 1, & \text{otherwise.} \end{cases} \tag{20}$$

We can see from (19) that the compactness objective can be generally achieved because D_{ij} would be as small as possible in the basic nodes allocation when the largest "cost" is selected.

It shoud be noted that in the heuristic multikernel growth the allocating basic node may be an unallocated one but may be an already allocated one. The proposed algorithm is with a certain degree of self-regulation and not sensitive to the initial

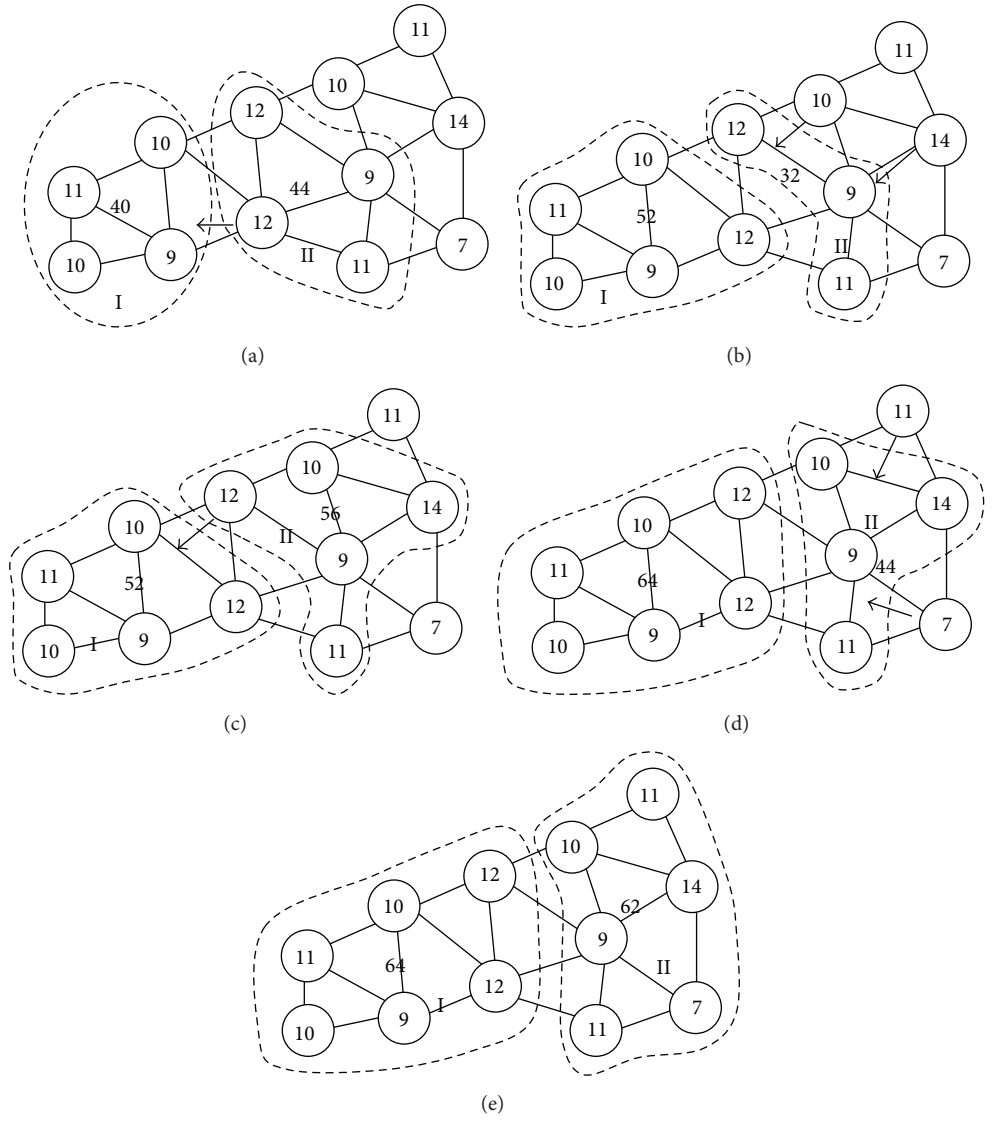


Figure 4: Kernels are growing.

choice of kernels. If allocating nodes must be unallocated nodes and the initial kernels are baddly selected, some zones would have no new unallocated node allocated in the process of zones growth. The final result may make a serious deviation from the balance object. For example, Figure 4(a) shows that zone I with the minimun workload (which is 40) needs to continue to be allocated nodes, but no unallocated nodes can be allocated. The proposed heuristic multikernel growth makes the zones growth "greedy" and "competitive." The zones with minimum workload may "eat" the adjacent node which belonged to other zones. In Figure 4(a), zone I gets the node with workload of 12 from zone II. Zone II needs to be allocated nodes because its workload becomes minimum. In accordance with the heuristic described previously the nodes with workload of 14 and 10 were allocated to zone II. According to the heuristic rules, the procedure would be continued as in Figures 4(c) and 4(d) and the final results are shown in Figure 4(e).

3.2.3. A Simple Optimization. The heuristic multikernel growth also has a simple and quick procedure of optimization for the result of nodes allocation after all nodes are allocated. The most classic optimization procedure for zoning problem may be the local search. It usually needs a huge time to run and it is hard to know when to stop the optimization procedure. The optimization procedure of the heuristic multikernel growth follows the same heurisic strategy as the nodes allocation. Its objective is to make the maximun workload of zones as small as possible (see (21)). So it is simple to be implemented and a little time comsuming:

$$\text{Minimize}\left(\max\left\{W_1,\ldots,W_j,\ldots,W_m\right\}\right). \tag{21}$$

3.2.4. The Implementation. The improved multikernel growth method includes three steps: (1) seeds selection; (2) nodes allocation; (3) simple optimization. There exist several approaches for determining a new configuration of

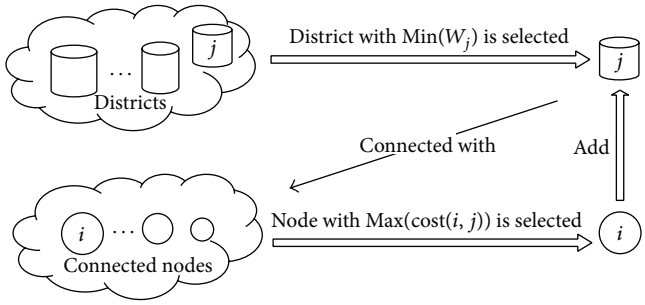


Figure 5: Allocating the basic nodes.

zone centers. Kalcsics et al. [26] thought a commonly used method was to solve in each territory resulting from the last allocation phase a 1-median problem. But it would be very complex, because much iteration should be taken to avoid bad cases. In our method, the initial seeds should be nodes which contain the largest number of companies. Of course it may not be optimization to take the initial seeds as kernels. If it is not optimization, this method will use a heuristic algorithm to adjust kernels. Nodes allocation takes into account three criteria: contiguity, balance, and compactness. The contiguity objective is assumed to be achieved as long as nodes are connected by edges in the graph. The balance and compactness objectives are achieved step by step in the allocation procedure. The general allocation procedure of improved multikernel growth method is illustrated in Figure 5. Zone j with the minimum workload would be selected to "grow." The connected basic node i with the maximum cost(i, j) will be selected and allocated to zone j. The simple optimization is similar to nodes allocation. Both of them follow the same heuristic strategy, but the main iteration of simple optimization is that if the first node with maximum "cost" in the nodes sorted list enlarges the minimum workload of zones or reduces the maximum workload of zones, it should be switched; otherwise the next node with the second maximum "cost" would be selected and repeat that logic. The iterative procedure would be continued until the maximum workload of zones could not be smaller any more by the heuristic strategy.

Assuming that the number of graphs which kernels generate is noted as g, the workload of the i-sub graph ($1 \le j \le g$) is noted as W_j. The main procedure of improved multikernel growth is shown as follows:

(a) Take one graph, the workload of which is noted as W_{sg}, and identify the number of kernels which is c where

$$c = m * \frac{W_{sg}}{\sum_{i=1}^{g} W_i}. \tag{22}$$

(b) Obtain c nodes with the largest number of companies as centers for zones, and take the number of companies as the initial workload of zones.

(c) Find out the zone (noted as K) with the minimum workload and find out its adjacent nodes via auxiliary subgraph firstly from unallocated nodes. If there are no unallocated nodes, the algorithm will find out its adjacent nodes from allocated ones; simultaneously calculate their cost by (19).

(d) Allocate the node with the maximum cost to the zone K and recalculate its workload W_j.

(e) If all the nodes in the subgraph are allocated, turn to step (c); otherwise continue.

(f) Take the zone with the minimum workload, and try to "eat" its adjacent node with the biggest "cost." If the maximum workload is not smaller or the zone whose basic unit is "eaten" becomes not continuous, another adjacent node with the next biggest "cost" would be tried.

(g) Continue step (f) until the maximum workload could not be smaller any more.

(h) Exit the procedure if the entire graph is handled; otherwise turn to step (a).

4. Computational Experiments and Results

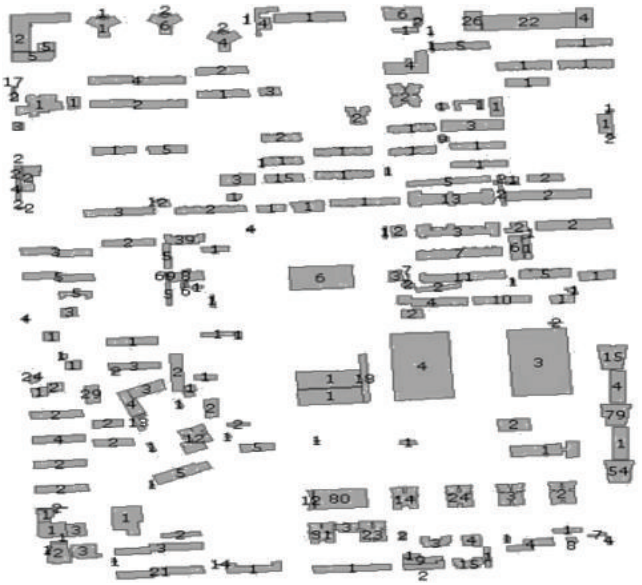
To test the performance of the proposed solution procedure, a series of experiments were generated. All problem experiments were solved on a 2.53 GHz Pentium processor with 1.93 AGPB of RAM running with Windows XP as the operating system. Our procedure model is solved based on ArcEngine 9.3. A real-world application from an economic census whose operations consist of surveying the financial state of companies within a service region was performed in this work. Test data were all obtained from the real data of an economic census in Beijing, China. We solved two different sizes of instances, which are classified by number of nodes and zones:

Test one: 213 nodes and 3 zones.

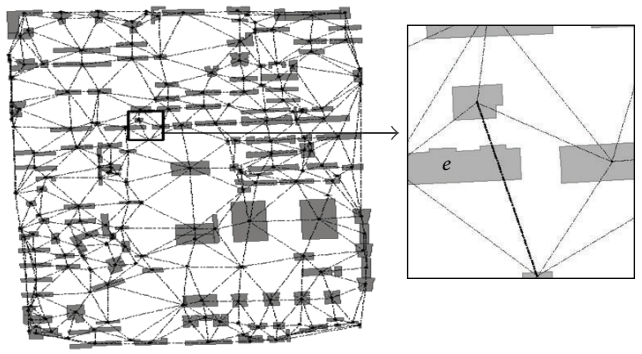
Test two: 741 nodes and 3\4\5 zones.

Each instance has two phases. Phase I checked the topology of polygons and got touching polygons recording zero distance. Then AGP would be generated. After AGP was created, we continued to turn the graph into AGPB wherever geometrical barriers existed. In Phase II, the heuristic multikernel growth approach was performed. The nodes with a relatively largest workload were selected as initial kernels. Nodes allocation and simple optimization would be applied following proposed heuristic strategy.

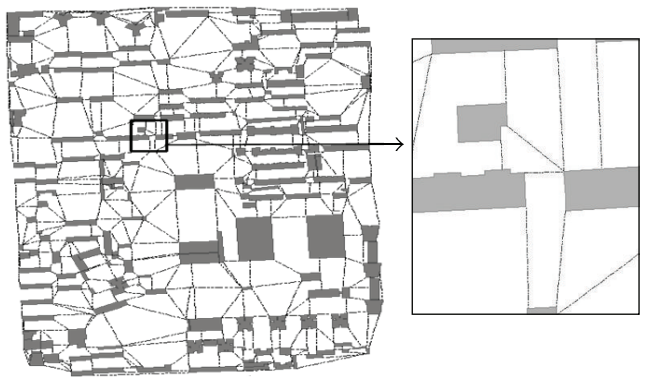
In test one, polygons were input and AGP was built in Phase I. In Phase II, several experiments are repeated on different values of parameters α and β. In Test two, geographical barriers are considered. Therefore, AGPB was built in Phase I. Several distances are solved based on different numbers of zones with the same values of α and β. Further, a comparison test between considering barriers and nonconsidering barriers is given in Section 4.2. This paper uses the Standard Deviation of Workload Values of zones to evaluate the balance of workload (STDEV_w) and uses the AVERAGE sum length of all MST edges in zones to evaluate the compactness of zones (AVERAGE_c). A smaller value of STDEV_w means a better balance result while a smaller value of AVERAGE_c means a better compactness result. Finally, time cost of instances was discussed.



(a) Polygon data and workload



(b) Construct the AGP based on points



(c) Construct the AGP based on polygons

Figure 6: Generation of the auxiliary graph.

4.1. Test One. The test data of test one is selected by a random rectangle from the economic census data. 213 buildings are selected into the test data and 1189 companies are involved. This indicates that there are 213 nodes whose total workload is 1189, and the number of companies located in every building is shown in Figure 6(a). Two spatial auxiliary graphs based on polygon center points and polygons were, respectively, built as shown in Figures 6(b) and 6(c). In Figure 6(b), edge e is a spatial relational edge crossing a building polygon. Edge e should not exist as a path in real world. What is more, length of edges in Figure 6(b) is generally longer than that in Figure 6(c) which could more accurately represent the true path length.

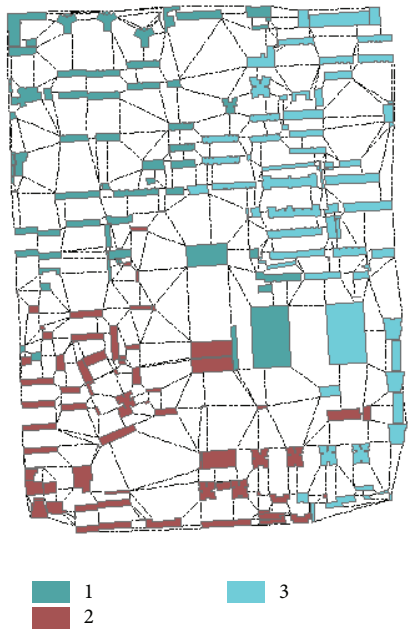


Figure 7: Result when $\alpha = 0$ and $\beta = 10$ according to polygons.

In generation of the graph, triangles between polygons were initially created and then AGP was created consecutively. Figure 6(b) shows details of AGPB. Finally, we decided to generate three zones from the test data.

In the first step of Phase II, we selected three nodes with the largest number of companies as kernels. In the second step, we set different values to α and different results are obtained. In Figure 7, we thought travel time on road can be negligible relative to the investigation time in companies, so we set $\alpha = 0$ and $\beta = 10$. The average value of workload is 396.33 (the number of companies actually) and that of three zones is 397, 396, and 396, respectively. This means that the result has got a good balance of workload for zoning. We also can find out that in Figure 7 continuity of zones has been held. Compactness of zones is used to control the total travel time in our application so it would be acceptable due to negligible travel time.

When $\alpha > 0$, the workload content included the travel cost and the compactness objective would take into account the distance of edges between polygons by setting $\beta = 10$. We set different values for α and the numerical results are presented in Figure 8. The parameter n stands for the number of companies, d for the average length of MST, and w for the workload in zones.

In Figure 8, we note that, in general, values of STDEV_w and AVERAGE_c almost have no large change with the change of the value of α. Two charts (see Figure 8) are generated from test results and support the assumption stated above. This indicates that the balance of workload and

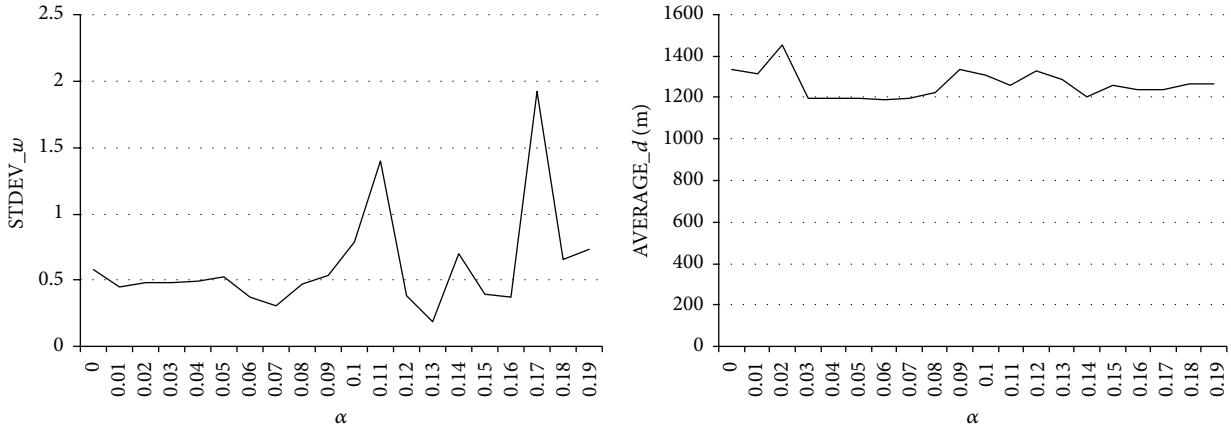


Figure 8: Standard deviation of workload and average length of MST.

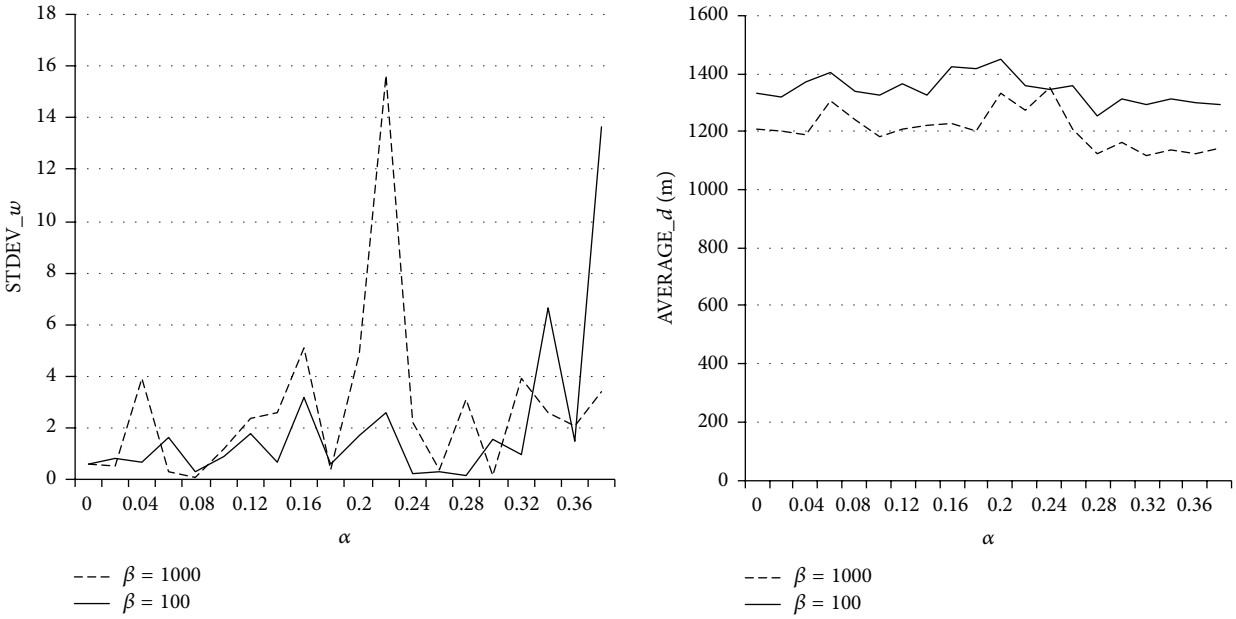


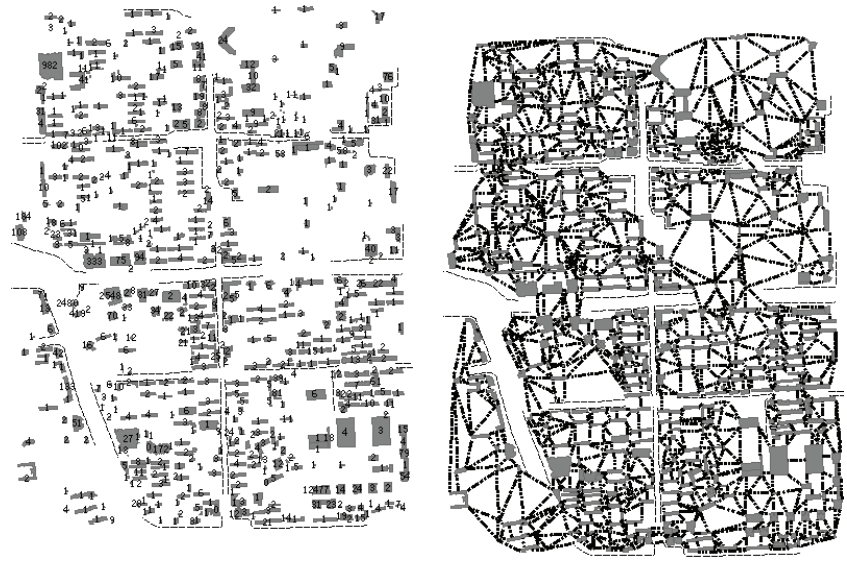
Figure 9: STDEV_w and AVERAGE_d/m when $\beta = 100$ and $\beta = 1000$.

the compactness objectives are affected not too much by the composition of workload content. In practical application, the appropriate value of α needs to be considered when considering workload content and we can set its value as a ratio of investigation time to travel time. With changes in the value of α, standard deviation of workload is stable and is always approximately 0.6 in most cases. The average value of distances changes around 1250. This indicates that the proposed heuristic works and holds the balance of workload and compactness to a certain extent when β is specified and whatever value α is.

To test the effect of the parameter β in the heuristic multi-kernel growth, another twenty experiments were performed with twenty different values of α and two specific values of β. The test result of experiments is shown as in Figure 9. It is noted that when the value of β is relatively small, the balance of workload is in general well held but the compactness is badly achieved. This means that β is a control to coordinate compactness and balance objective. From Figure 9 we can find out that a specific β corresponds to a relatively specific value of AVERAGE_c. So in the real application we usually specify β by considering the travel time limits or means of transportation.

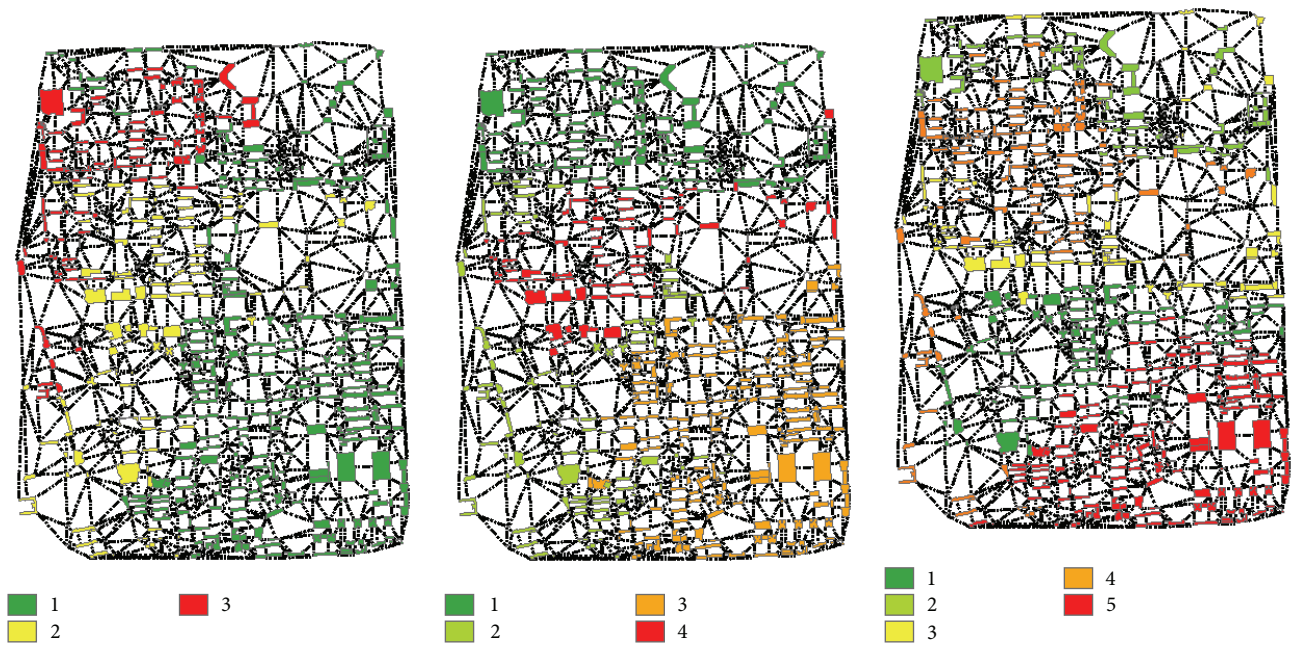
4.2. Test Two. Test data of large instances are selected by a random rectangle from the economic census data. There are 741 polygon nodes whose total workload is 7107 and the number of companies located in every building is shown in Figure 10(a). In addition, we considered some busy roads or fences through the test data as geographical barriers, which can be seen in Figure 10(b). Finally, we aimed at generating 3\4\5 zones from the test data.

To illustrate the impact of geometrical barriers to the zoning results, we have made comparison experiments without



(a) Test data for large instance (b) Auxiliary graph for large instance

Figure 10: Test data and its auxiliary graph with geometrical barriers.

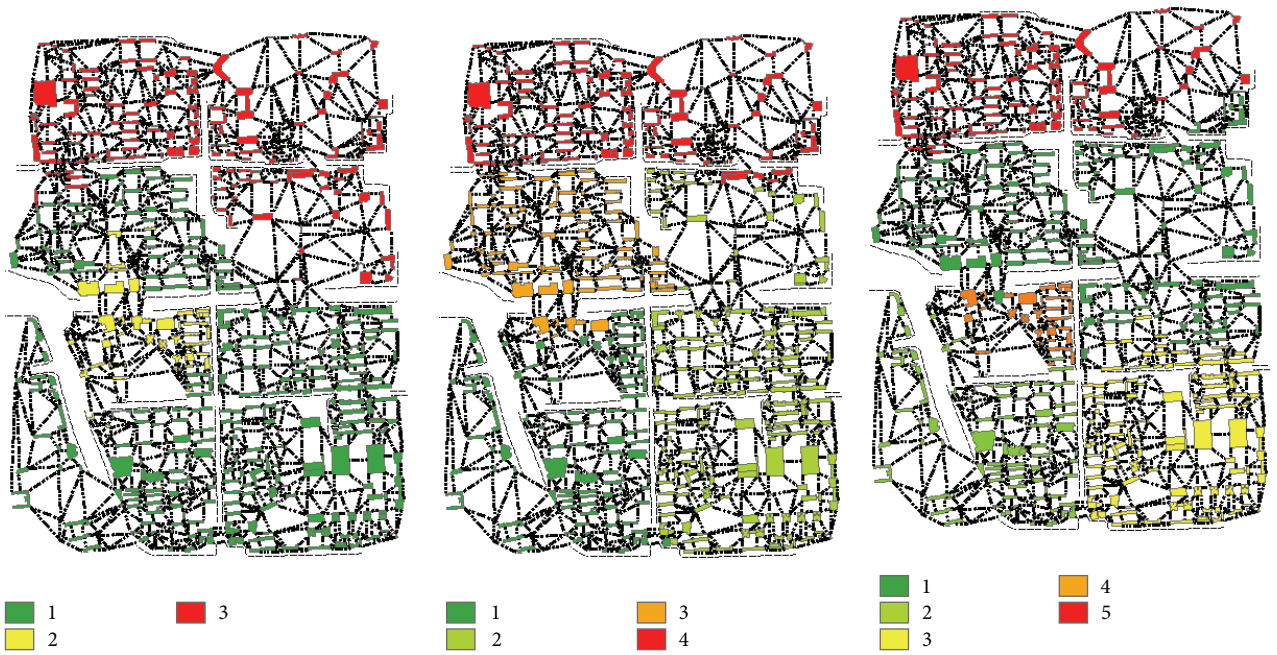


(a) 3 zones with no barriers (b) 4 zones with no barriers (c) 5 zones with no barriers

Figure 11: Results of 3\4\5 zones with no barriers.

barriers but with the same parameters α and β. In Figures 11 and 12, we could find out that geometrical barriers had affected the zoning results.

The comparison between effects of zoning results also has been made. Set $\alpha = 0.08$ and $\beta = 100$, and effects of balance and compactness between the 3\4\5 zones with barriers or no barriers are shown in Table 1. We can see that geometrical barriers have affected the zoning results mainly affecting the distribution of members in zones but having relatively little effect to zoning results.



(a) 3 zones with barriers (b) 4 zones with barriers (c) 5 zones with barriers

Figure 12: Results of 3\4\5 zones with barriers.

Table 1: Results of large instances.

α	β	Barriers	Three zones		Four zones		Five zones	
			STDEV_w	AVERAGE_c	STDEV_w	AVERAGE_c	STDEV_w	AVERAGE_c
0.08	100	Do not exist	0.215094	12598.68	0.555236	9234.909	1.406661	7297.194
0.08	100	Exist	0.598867	11977.43	0.429749	9460.601	0.505428	7682.173

Table 2: Time cost in different instances.

Instances	Parameters	Barriers	Phase I	Phase II	Total time
213_3	$\alpha, \beta = 10$	Do not exist	15.406 s	6.25 ms	15.412 s
	$\alpha, \beta = 100$	Do not exist	15.406 s	6.25 ms	15.412 s
	$\alpha, \beta = 1000$	Do not exist	15.406 s	6.25 ms	15.412 s
741_3	$\alpha = 0.08,\ \beta = 100$	Exist	53.875 s	46.875 ms	53.922 s
		Do not exist	52.921 s	53.125 ms	52.974 s
741_4	$\alpha = 0.08,\ \beta = 100$	Exist	53.875 s	48.437 ms	53.923 s
		Do not exist	52.921 s	51.562 ms	52.973 s
741_5	$\alpha = 0.08,\ \beta = 100$	Exist	53.875 s	48.438 ms	53.923 s
		Do not exist	52.921 s	51.563 ms	52.973 s

Constrained Delaunay Triangulation can successfully deal with our zoning problem with barriers (see Figures 11 and 12) while holding an acceptable quality of zoning.

4.3. Time Cost. For different situations, we may construct different graphs by identifying the data types and existence of barriers in Phase I and adopt the heuristic multikernel growth in Phase II. In Sections 4.1 and 4.2, we observe that acceptable results can be obtained. What is more, time cost of the procedure is very low, which is one of notable features discussed in this paper. In Table 2, time cost for instances discussed above is presented.

Instances of 213_3 have been repeated for 20 times with different values of α and computational time of Phase II is the average value of results from 20 repeated tests. We can observe that the time cost in Phase II of the heuristic multikernel growth is so little that it can be almost negligible comparing with the time cost in the whole process of zoning. In this zoning procedure, running times in our results are in milliseconds while traditional methods usually consider a few seconds or minutes to optimize zoning results [25–27]. The construction of the graph in Phase I consumed too much computational time. Fortunately, however, the procedure of Phase I is separate and does not interact with that of Phase

II. Therefore, once the graph is built up and stored in a file, there is no necessity to rebuild the auxiliary graph and the file only needs to be read prior to the procedure of the proposed heuristic strategy with different values of parameters α and β. Therefore, the average computational time will be decreased greatly and can well meet our requirements of the special zoning problem in the economic census.

5. Conclusions and Possible Further Work

This paper addresses a special zoning problem for economic census that is motivated by a real-world application. Firstly, we introduced the Constrained Delaunay Triangulation to solve several problems such as polygon-based graph construction problem and geographical barriers problem which are generally countered in the zoning problem. The difficulty of this NP-complete problem motivated us to propose the heuristic multikernel growth following a heuristic strategy to speed up the zoning process greatly in the premise of the required quality of zoning. According to real-world instances, we present problem formulations to optimize three criteria: contiguity, compactness, and balance of workload among zones.

Two special instances for economic census were performed, highlighting the applicability of this approach. They resulted in acceptable compact zones with considerable balance of workload. Test one showed the partition results of touching and nontouching polygons. Some experiments showed how to determine the values of α and β and the meanings of them; that is, the values of α determine workload composition while β determines balance and compactness object. Test one also showed that compactness would be better achieved if β is larger while balance object would become worse. Test two showed that our method could successfully solve zoning problem with barriers and still could achieve some reasonable results. Test three in Section 4.3 showed that our method could get optimized results within acceptable time cost.

The simple spatial heuristic approach in this paper makes a good performance in most of cases, but sometimes it maybe results in bad compactness objective. The reason is that the compactness objective is achieved by minimizing the maximum length of MST constructed in the zones when the kernels are growing without considering the shape of zones. In the future, we will do our efforts on the problem and maybe take into account the shape of zones for compactness.

Conflict of Interests

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

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant nos. 41222009 and 41271405) and the foundation of BNU (Project no. 111007022).

References

[1] L. Muyldermans, D. Cattrysse, and D. Van Oudheusden, "District design for arc-routing applications," *Journal of the Operational Research Society*, vol. 54, no. 11, pp. 1209–1221, 2003.

[2] M. Altman, "The computational complexity of automated redistricting: is automation the answer?" *Rutgers Computer & Technology Law Journal*, vol. 23, no. 1, pp. 81–142, 1997.

[3] P. K. Bergey, C. T. Ragsdale, and M. Hoskote, "A simulated annealing genetic algorithm for the electrical power districting problem," *Annals of Operations Research*, vol. 121, no. 1, pp. 33–55, 2003.

[4] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau, "Nonpartisan political redistricting by computer," *Operations Research*, vol. 13, no. 6, pp. 998–1006, 1965.

[5] M. E. T. Horn, "Solution techniques for large regional partitioning problems," *Geographical Analysis*, vol. 27, no. 3, pp. 230–248, 1995.

[6] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser, "An optimization based heuristic for political districting," *Management Science*, vol. 44, no. 8, pp. 1100–1114, 1998.

[7] F. Ricca and B. Simeone, "Local search algorithms for political districting," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1409–1426, 2008.

[8] C. Easingwood, "A heuristic approach to selecting sales regions and territories," *Operational Research Quarterly*, vol. 24, no. 4, pp. 527–534, 1973.

[9] L. M. Lodish, "Sales territory alignment to maximize profit," *Journal of Marketing Research*, vol. 12, no. 1, pp. 30–36, 1975.

[10] A. A. Zoltners and P. Sinha, "Integer programming models for sales resource allocation," *Management Science*, vol. 26, no. 3, pp. 242–260, 1980.

[11] J. A. Ferland and G. Guenette, "Decision support system for the school districting problem," *Operations Research*, vol. 38, no. 1, pp. 15–21, 1990.

[12] R. G. González-Ramírez, N. R. Smith, R. G. Askin, and V. Kalashinkov, "A heuristic approach for a logistics districting problem," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 8, pp. 3551–3562, 2010.

[13] C. F. Daganzo, "The distance traveled to visit N points with a maximum of C stops per vehicle: an analytic model and an application," *Transportation Science*, vol. 18, no. 4, pp. 331–350, 1984.

[14] G. F. Newell and C. F. Daganzo, "Design of multiple-vehicle delivery tours—I a ring-radial network," *Transportation Research Part B*, vol. 20, no. 5, pp. 345–363, 1986.

[15] A. Langevin and F. Soumis, "Design of multiple-vehicle delivery tours satisfying time constraints," *Transportation Research Part B*, vol. 23, no. 2, pp. 123–138, 1989.

[16] T. J. Cova and R. L. Church, "Contiguity constraints for single-region site search problems," *Geographical Analysis*, vol. 32, no. 4, pp. 306–329, 2000.

[17] J. C. J. H. Aerts and G. B. M. Heuvelink, "Using simulated annealing for resource allocation," *International Journal of Geographical Information Science*, vol. 16, no. 6, pp. 571–587, 2002.

[18] J. C. Williams, "A zero-one programming model for contiguous land acquisition," *Geographical Analysis*, vol. 34, no. 4, pp. 330–349, 2002.

[19] A. T. Murray and D. Q. Tong, "Coverage optimization in continuous space facility siting," *International Journal of Geographical Information Science*, vol. 21, no. 7, pp. 757–776, 2007.

[20] L. Yuan, Z. Yu, W. Luo, L. Yi, and Y. Hu, "Pattern forced geophysical vector field segmentation based on Clifford FFT," *Computers & Geosciences*, vol. 60, pp. 63–69, 2013.

[21] P. Moreno-Regidor, J. García López de Lacalle, and M.-Á. Manso-Callejo, "Zone design of specific sizes using adaptive additively weighted Voronoi diagrams," *International Journal of Geographical Information Science*, vol. 26, no. 10, pp. 1811–1829, 2012.

[22] A. G. N. Novaes, J. E. Souza de Cursi, A. C. L. da Silva, and J. C. Souza, "Solving continuous location-districting problems with Voronoi diagrams," *Computers and Operations Research*, vol. 36, no. 1, pp. 40–59, 2009.

[23] L. Svec, S. Burden, and A. Dilley, "Applying Voronoi diagrams to the redistricting problem," *The UMAP Journal*, vol. 28, no. 3, pp. 313–329, 2007.

[24] T. Shirabe, "A model of contiguity for spatial unit allocation," *Geographical Analysis*, vol. 37, no. 1, pp. 2–16, 2005.

[25] P. K. Bergey, C. T. Ragsdale, and M. Hoskote, "A decision support system for the electrical power districting problem," *Decision Support Systems*, vol. 36, no. 1, pp. 1–17, 2003.

[26] J. Kalcsics, S. Nickel, and M. Schröder, "Towards a unified territorial design approach—applications, algorithms and GIS integration," *Top*, vol. 13, no. 1, pp. 1–74, 2005.

[27] R. Z. Ríos-Mercado and E. Fernández, "A reactive GRASP for a commercial territory design problem with multiple balancing requirements," *Computers and Operations Research*, vol. 36, no. 3, pp. 755–776, 2009.

[28] C.-I. Chou and S. P. Li, "Taming the gerrymander—statistical physics approach to political districting problem," *Physica A: Statistical Mechanics and Its Applications*, vol. 369, no. 2, pp. 799–808, 2006.

[29] C.-I. Chou and S.-P. Li, "Spin systems and political districting problem," *Journal of Magnetism and Magnetic Materials*, vol. 310, no. 2, pp. 2889–2891, 2007.

[30] M. S. Heschel, "Effective sales territory development," *Journal of Marketing*, vol. 41, no. 2, pp. 39–43, 1977.

[31] L. M. Lodish, "'Vaguely right' approach to sales force allocations," *Harvard Business Review*, vol. 51, pp. 119–124, 1974.

[32] L. D. Bodin, "A districting experiment with a clustering algorithm," *Annals of the New York Academy of Sciences*, vol. 219, pp. 209–214, 1973.

[33] M. Blais, S. D. Lapierre, and G. Laporte, "Solving a home-care districting problem in an urban setting," *Journal of the Operational Research Society*, vol. 54, no. 11, pp. 1141–1147, 2003.

[34] W. Macmillan, "Redistricting in a GIS environment: an optimisation algorithm using switching-points," *Journal of Geographical Systems*, vol. 3, no. 2, pp. 167–180, 2001.

[35] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump, "A simulated annealing approach to police district design," *Computers and Operations Research*, vol. 29, no. 6, pp. 667–684, 2002.