

Article

# A Smart Tourism Recommendation Algorithm Based on Cellular Geospatial Clustering and Multivariate Weighted Collaborative Filtering

Xiao Zhou <sup>1</sup>, Jiangpeng Tian <sup>2,\*</sup> , Jian Peng <sup>1</sup> and Mingzhan Su <sup>2</sup> 

<sup>1</sup> College of Computer Science, Sichuan University, Chengdu 610065, China; zhouxiao@infu.ac.cn (X.Z.); jianpeng@scu.edu.cn (J.P.)

<sup>2</sup> Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China; smzh625@infu.ac.cn

\* Correspondence: tjp\_study@infu.ac.cn; Tel.: +86-135-9241-9753

**Abstract:** Tourist attraction and tour route recommendation are the key research highlights in the field of smart tourism. Currently, the existing recommendation algorithms encounter certain problems when making decisions regarding tourist attractions and tour routes. This paper presents a smart tourism recommendation algorithm based on a cellular geospatial clustering and weighted collaborative filtering. The problems are analyzed and concluded, and then the research ideas and methods to solve the problems are introduced. Aimed at solving the problems, the tourist attraction recommendation model is set up based on a cellular geographic space generating model and a weighted collaborative filtering model. According to the matching degree between the tourists' interest needs and tourist attraction feature attributes, a precise tourist attraction recommendation is obtained. In combination with the geospatial attributes of the tourist destination, the spatial adjacency clustering model based on the cellular space generating algorithm is set up, and then the weighted model is introduced for the collaborative filtering recommendation algorithm, which ensures that the recommendation result precisely matches the tourists' needs. Providing precise results, the optimal tour route recommendation model based on the precise tourist attraction approach vector algorithm is set up. The approach vector algorithm is used to search the optimal route between two POIs under the condition of multivariate traffic modes to provide the tourists with the best motive benefits. To verify the feasibility and advantages of the algorithm, this paper designs a sample experiment and analyzes the resulting data to obtain the relevant conclusion.

**Keywords:** tourist attraction cellular unit; cellular generating space; weighted collaborative filtering; smart recommendation; tour route planning



**Citation:** Zhou, X.; Tian, J.; Peng, J.; Su, M. A Smart Tourism Recommendation Algorithm Based on Cellular Geospatial Clustering and Multivariate Weighted Collaborative Filtering. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 628. <https://doi.org/10.3390/ijgi10090628>

Academic Editors: Wolfgang Kainz, Sara Shirowzhan and Aaron Costin

Received: 18 July 2021

Accepted: 14 September 2021

Published: 19 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A smart recommendation system that provides suitable tourist attraction and route recommendations for tourists is a research highlight in the field of smart tourism [1]. It can automatically provide potentially interesting tourist attractions for tourists and offer them a convenient service. The current recommendation methods include the user-based method, the content-based method, the knowledge-based method, the user characteristics-based method, and the association rules-based method, etc. Each method has its advantages and disadvantages, and the user-based method and content-based method are frequently used [2,3].

Mou [4] used the collaborative filtering method to forecast tourists' evaluation scores on tourist attractions in accordance with similar tourist attraction scores. Filipe Santos [5] proposed a recommendation system that considers users' functionality levels. Yanqing Cui [6] designed a kind of tourist attraction label system. According to the association model among the tourist, tourist attraction and tourism label, the user interest

model was set up to recommend tourist attractions. Choi Il Young [7] proposed a travel recommendation system based on collaborative filtering and constraint satisfaction filtering. Li Guangli [8] used a questionnaire survey and the crawler algorithm to obtain the tourist and tourist attraction data. The collaborative filtering method was used to calculate the textual similarity between the target user and each cluster center and finally obtain the recommendation list. Ya Zhou [9] presented a kind of collaborative filtering recommendation method based on tourists' location labels and preferences. Mehrbakhsh Nilashi [10] presented a hybrid collaborative filtering recommendation method based on dimension reduction and prediction technology. Shi [11] designed a tourism collaborative filtering recommendation model that incorporates tourist attraction attributes. Chen [12] used the collaborative filtering algorithm to recommend the tourist attractions. Zhu [13] used a stratified sampling statistical model to obtain users' preferences. Furthermore, the improved Bayes individualized sequencing method was used to optimize the recommendation system. Li [14] presented a collaborative filtering algorithm. It used the clustering method DBSCAN to set up a user consumption model and obtain a the recommendation system for the WeChat applet. Liu [15] studied the tourism recommendation systems based on a clustering algorithm, stratified sampling statistical model and SVD++ algorithm, respectively. Chen [16] presented a tourism group recommendation method combined with collaborative filtering and user preferences. Leila Esmaeili [17] set up a tourism recommendation system under the condition of social commerce. Shen [18] obtained massive historical tour data and sets up the personalized attraction similarity (PAS) model to recommend tourist attractions.

Analyzing the related work, it can be seen that the tourism recommendation method has the following problems. Problem (1): Previous studies mainly focus on the principle and theory of the collaborative filtering algorithm in terms of the aspects of algorithm efficiency, accuracy, data sparsity, and the cold start problem. Problem (2): The collaborative filtering algorithm is still a fuzzy recommendation mode without high accuracy. It only relies on the historical tour data and comment data to determine the current tourist's interests and needs. Problem (3): The research on the precise matching degree between the tourist attraction feature attributes and tourist interest feature attributes is not sufficient. Problem (4): The collaborative filtering algorithm directly uses the historical tour route, neglecting the current tourist's interest needs and the real-world geographic conditions for the tourist. The traditional route planning algorithm uses the main roads and avenues, while it neglects the secondary roads. It may not identify the optimal route.

Figure 1 shows the sequence of the paper. Aimed at addressing these problems, this paper presents a tourism recommendation model based on cellular automata geographic space clustering and a weighted collaborative filtering algorithm. This algorithm could be used as an embedded algorithm for a smart tourism recommendation system developed, managed, and operated by the tourism sectors. Moreover, the system is ultimately used by tourists on the website. This model is able to solve the analyzed problems. The first is the problem in the precise tourist attraction recommendation mechanism of the collaborative filtering algorithm. The second is the matching mechanism between the tourist attraction and tourist in the collaborative filtering algorithm. The third is the problem of optimal tour route planning based on the precise recommendation result.

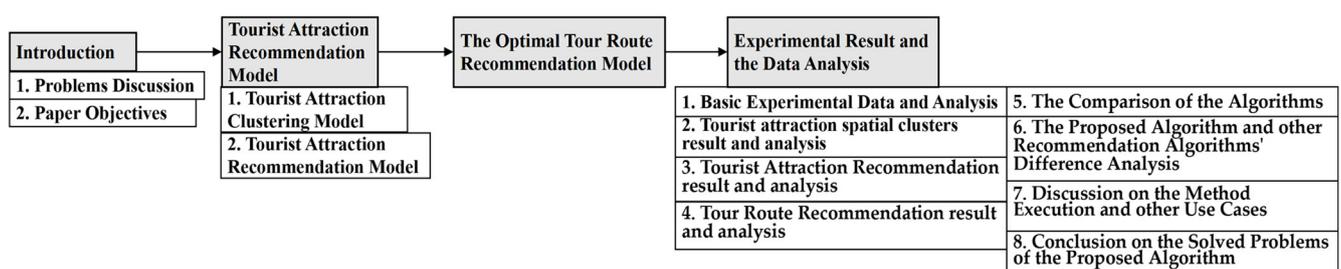


Figure 1. The sequence of the paper content.

## 2. Tourist Attraction Recommendation Model Based on Cellular Geospatial Generating and Weighted Collaborative Filtering

First, it is necessary to cluster the tourist attractions. A tourist usually takes a certain tourism city as their destination. The typical tourist attractions are the places that tourists will visit [19–22]. This paper takes the urban area as the research range to set up the algorithm. The tourist attractions in the urban area have the discrete characteristics within the geospatial range. Each tourist attraction is connected by an urban road network, and tourists can freely travel between tourist attractions by different transportation methods. The recommendation should combine feature attributes and its matching with tourists' interest needs. Here, the first group of definitions and tourist attraction clustering are presented.

Def 1.1 Tourist attraction research domain  $X$ . The  $n$  number of urban tourist attractions that possess geospatial attributes and feature attributes are grouped into one set, and this set is defined as tourist attraction research domain  $X$ .

Def 1.2 The element  $x_{(k)}$  of  $X$ . Tourist attraction in the domain  $X$  is the element  $x_{(k)}$  of  $X$ ,  $k \in (0, n] \subset \mathbb{Z}^+$ .

Def 1.3 The feature attribute vector  $\mathbf{x}_{(k)}$  for the element  $x_{(k)}$  of  $X$ . As to one element  $x_{(k)}$  of  $X$ , its attributes can be quantified into a one-dimensional feature vector, defined as the feature attribute vector  $\mathbf{x}_{(k)}$  for the element  $x_{(k)}$  of  $X$ .

Def 1.4 Tourist attraction cluster  $X_{(j)}$ . Divide the  $n$  number of tourist attractions into  $c$  number of subsets  $X_{(j)}$  and ensure that the tourist attributions of different subsets have a low intimacy degree while tourist attributions in the same subset have a high intimacy degree. The subset is defined as the tourist attraction cluster  $X_{(j)}$ ,  $j \in (0, c] \subset \mathbb{Z}^+$ .

### 2.1. The Spatial Adjacency Tourist Attraction Clustering Model Based on the Cellular Space Generating Algorithm

Tourist attractions, urban roads and their intersections are the basic environment for tourism activities. The clustering of the tourist attractions is necessary to ensure the matching degree between the tourist attraction and tourist [23–26]. The tourist attraction clustering model based on the cellular space generating algorithm is set up.

#### 2.1.1. Tourist Attraction Cellular Space Generating Algorithm

The cellular generating algorithm is set up to generate the optimized space structure of the tourist attraction element  $x_{(k)}$ . It realizes the optimal clustering of domain  $X$ . The second group of definitions and the algorithm process are presented below.

Def 2.1 Tourist attraction cellular core  $O_{(k)}$ . The element  $x_{(k)}$  of  $X$  is defined as the tourist attraction cellular core  $O_{(k)}$ . It is the generating root of the tourist attraction cellular generating unit  $C_{(k)}$ .

Def 2.2 Tourist attraction cellular space anchor point  $P_{(v)}$ . In a random and uniform manner,  $w$  number of intersections  $P_{(v)}$  of the main roads are defined as the tourist attraction cellular space anchor point.

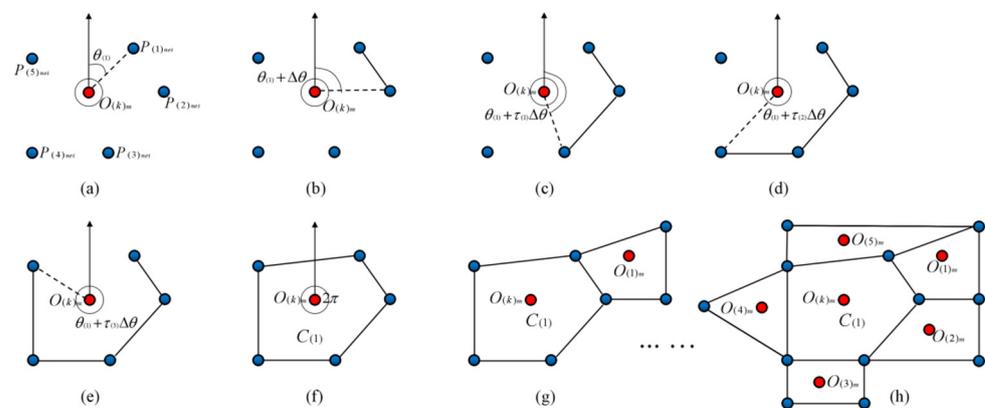
Def 2.3 Cellular generating unit  $C_{(k)}$ . The irregular closed polygon  $C_{(k)}$ , which is formed by points  $P_{(v)}$  and contains the cellular core  $O_{(k)}$ , is defined as the tourist cellular generating unit  $C_{(k)}$ . It is the basic unit to generate the cellular space, and it is the minimum unit that contains the cellular core  $O_{(k)}$ .

Def 2.4 Tourist attraction cellular space  $\mathbf{C}$ . Expand the unit to form the neighborhood cellular core  $C_{(k)nei}$ , which contains the neighborhood core  $O_{(k)nei}$ . Starting from  $C_{(k)}$ , perform the directional expansion. When all the points  $O_{(k)nei}$  converge and all  $C_{(k)}$  are formed, the space  $\mathbf{C}$  is generated. The cellular generating unit  $C_{(k)}$  and cellular space  $\mathbf{C}$  generating algorithm based on the core  $O_{(k)}$  and points  $P_{(v)}$  is set up, and it is defined as Algorithm 1. Below is the Algorithm 1 flow pseudo-code.

Figure 2 is an example to form the space  $\mathbf{C}$ . Figure 2a–f illustrate the process of forming the cellular unit  $C_{(1)}$ , and Figure 2g–h illustrate the topological process of forming the cellular space  $\mathbf{C}$  based on the cellular unit  $C_{(1)}$ .

**Algorithm 1** The cellular generating unit  $C_{(k)}$  and cellular space  $C$  generating algorithm

- 1: **Step 1:** Generate and confirm the  $O_{(k)}$  set for the tourist attraction cellular cores  $O_{(k)}$  and the set  $P_{(v)}$  for the cellular geospatial anchor points  $P_{(v)}$ .
- 2: **Sub-step 1:** Generate the set  $O_{(k)}$ .
- 3: **Sub-step 2:** Encode the points  $P_{(v)}$  in set  $P_{(v)}$ .
- 4: **Sub-step 3:** Encode the points  $O_{(k)}$  in set  $O_{(k)}$ .
- 5: **Sub-step 4:** Confirm the coordinate  $O_{(k)}$  as  $y_{(1,O_{(k)})}$  and  $y_{(2,O_{(k)})}$ ,  $P_{(v)}$  as  $y_{(1,P_{(v)})}$  and  $y_{(2,P_{(v)})}$ .
- 6: **Step 2:** Generate the initial cellular generating unit  $C_{(1)}$ .
- 7: **Sub-step 1:** Generate one ray  $lO_{(k)}$  to the north direction starting from the cellular core  $O_{(k)}$ .
- 8: **Sub-step 2:** Set up the open list  $O_{(k)}$  and closed list  $C_{l(i)}$  for  $P_{(v)}$ .
- 9: **Sub-step 3:** Turn  $lO_{(k)}$  and search the neighborhood points  $P_{(v)nei}$ .
- 10: **Step 3:** Generate the unit  $C_{(2)}$  of the initial cellular generating unit  $C_{(1)}$ .
- 11: **Step 4:** Generate all the unit  $C_{(1)nei}$  of the cellular generating unit  $C_{(1)}$ .
- 12: **Step 5:** Continue generating neighborhood units until all the points converge.



**Figure 2.** The process of forming the space  $C$  based on  $O_{(k)}$  and  $P_{(v)}$ . (a–f) illustrate the process of forming the cellular unit  $C_{(1)}$ , and (g,h) illustrate the topological process of forming the cellular space  $C$  based on the cellular unit  $C_{(1)}$ .

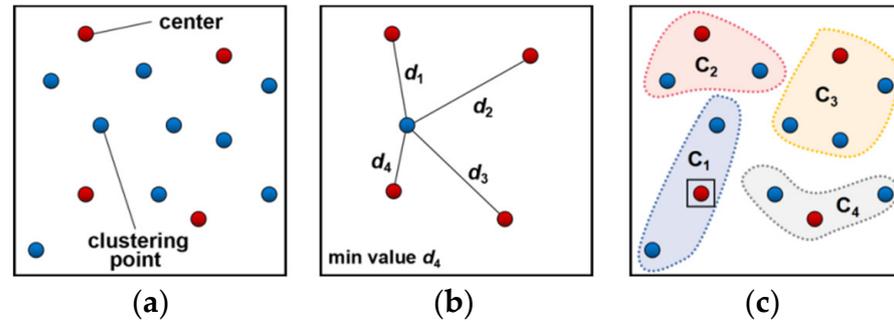
### 2.1.2. Tourist Attraction Clustering Algorithm Based on Geospatial Feature Attribute

In the process of clustering, to generate the cluster  $C_{l(u)}$ , the cellular core  $O_{(k)}$  is defined as the dynamic clustering center  $C_e$ . The  $C_e$  will change with the spatial geometric topology of the cluster  $C_{l(u)}$  in the process of multiple iterations and form the steady-state clustering center. The tourist attraction clustering algorithm based on the geospatial feature attribute is set up, and it is defined as Algorithm 2, and the Algorithm 2 pseudo-code is as follows. After the search, the tourist attraction space  $C$  forms the cluster  $C_{l(u)}$  distribution state with the core of the steady-state clustering centers  $C_{e(v)}$ .

**Algorithm 2** The tourist attraction clustering algorithm based on the geospatial feature attribute

- 1: **Step 1:** Randomly and uniformly choose  $k$  number of dynamic clustering centers  $C_{e(v)}$ .  $v \in (0, k] \subset Z^+$ , noted  $C_{e+}$ .
- 2: **Step 2:** Store  $C_{e(v)}$  into vector  $K_{(1)}$ , store  ${}^{-}C_{e(v2)}$  into  $K_{(2)}$ , note the element as  $K_{(1,v1)}$  and  $K_{(2,v2)}$ .  $v1 \in (0, k] \subset Z^+$ , and  $v2 \in (0, n - k] \subset Z^+$ . Calculate  $d_{(C_{e(v1)}, {}^{-}C_{e(v2)})}$ .
- 3: **Sub-step 1:** Calculate  $d_{(K_{(2,1)}, K_{(1,1)})}$ .
- 4: **Sub-step 2:** Converse  $v1 \in (0, k] \subset Z^+$ , calculate  $d_{(K_{(2,1)}, K_{(1,v1)})}$ .
- 5: **Sub-step 3:** Set up matrix  $\sim K_{(1)}$  to store  $d_{(K_{(2,1)}, K_{(1,v1)})}$  in ascending order.
- 6: **Sub-step 4:** Take the minimum element  $\sim K_{(1)}$  of  $K_{(2,1)}$  as the nearest  $O_{(k)}$  for  $O_{(v1)}$  and absorb in  $C_{l(k(1,v1))}$ .
- 7: **Step 3:** Calculate  $d_{(K_{(2,2)}, K_{(1,v)})}$ ,  $v1 \in (0, k] \subset Z^+$ . Take the minimum element  $\sim K_{(1)}$  of  $K_{(2,2)}$  as the nearest  $O_{(k)}$  for  $O_{(v1)}$  and absorb in  $C_{l(k(1,v1))}$ .
- 8: **Step 4:** Converse to calculate  $d_{(K_{(2,v2)}, K_{(1,v)})}$ , take the minimum element  $\sim K_{(1)}$  of  $K_{(2,v2)}$  as the nearest  $O_{(k)}$  for  $O_{(v1)}$  and absorb in  $C_{l(k(1,v1))}$ .

Figure 3 is an example for Algorithm 2. Four centers (red points) are chosen; see Figure 3a. For one point out of the center (blue point), calculate the value  $d$  between the blue point and all the red points and determine the minimum value; see Figure 3b. The blue point with the minimum value belongs to the red point cluster (with square); see Figure 3c. Perform the process on all the blue points and determine which blue point belongs to which red point cluster.



**Figure 3.** An example for Algorithm 2. (a) is the sample distribution, (b) shows the  $d$  samples, and the (c) shows the clustering sample.

## 2.2. Tourist Attraction Recommendation Model Based on Weighted Collaborative Filtering

Note the tourist attractions'  $m$  dimension feature attributes and the topological attributes as the keywords for the text mining. By mining the tourist attraction feature attribute labels, tourists' preferences regarding the tourist attraction classification can be obtained from the statistical data [27–29]. The  $K$ -neighborhood algorithm is used to obtain the historical tourists who best match the current tourists' feature attributes so as to recommend the best-matching tourist attraction classification for the current tourists. Then, they choose a certain quantity of tourist attractions in each classification according to the time schedule. Below is the third group of definitions and the tourist attraction recommendation model based on the weighted collaborative filtering algorithm.

**Def 3.1** Tourist attraction feature classification  $\mathbf{T}_{(i)}$ . Urban tourist attractions could be classified into  $c$  classifications  $\mathbf{T}_{(i)}$ . Its tourist attraction is noted as  $T_{(i,j)}$ . The  $\mathbf{T}_{(i)}$  divides the domain  $X$  into  $n$  classifications. The tourist attractions  $T_{(i,j)}$  of the same feature classification  $\mathbf{T}_{(i)}$  have similar feature attributes.

Tourist attractions  $T_{(i,j)}$  in the two different  $\mathbf{T}_{(i)}$  and  $\mathbf{T}_{(-i)}$  have different feature attributes. The domain  $X$  is divided into  $n$  classifications in order to divide the set  $\mathbf{T} = \{T_{(i)} | 0 < i \leq n\}$  into certain finite classifications  $\mathbf{T}_{(i)}$ . The classification meets the condition:  $\mathbf{T}_{(1)} \cup \mathbf{T}_{(2)} \cup \dots \cup \mathbf{T}_{(c)} = \mathbf{T}$ ,  $\mathbf{T}_{(i1)} \cup \mathbf{T}_{(i2)} = \emptyset$ ,  $0 < i1 \neq i2 \leq c$ ,  $\mathbf{T}_{(i)} \neq \emptyset$ ,  $\mathbf{T}_{(i)} \neq \mathbf{T}$ ,  $0 < i \leq c$ .

**Def 3.2** Feature attribute label meta vector  $\mathbf{L}_{(i)}$ , feature attribute label topological vector  $\mathbf{L}_{(i)to}$  and feature attribute label matrix  $\mathbf{L}_M$ . The  $m$  dimension feature attributes  $x_{(k1)}, x_{(k2)}, \dots, x_{(km)}$  are extracted as the keywords and stored in the  $m \times 1$  dimension column vector, noted as  $\mathbf{L}_{(i)} = \{L_{(1)}, L_{(2)}, \dots, L_{(m)}\}^T$ . It is called feature attribute label meta vector  $\mathbf{L}_{(i)}$ . Based on the single label element  $L_{(i)}$  in the vector  $\mathbf{L}_{(i)}$ , expand  $n$  number of keyword labels  $L_{(i,j)}$  with the same or similar semantics to the label  $L_{(i)}$  and absorb them into the  $1 \times n$  dimension vector  $\mathbf{L}_{(i)to}$  with the initial element  $L_{(i)}$ . This vector is defined as feature attribute label topological vector  $\mathbf{L}_{(i)to}$ , noted as  $\mathbf{L}_{(i)to} = \{L_{(i,1)}, L_{(i,2)}, \dots, L_{(i,n)}\}$ . The  $m$  number of vectors  $\mathbf{L}_{(i)to}$  are arranged in the sequence of the feature attribute label  $L_{(i)}$  sign to form the  $m \times n$  dimension matrix, defined as the feature attribute label matrix  $\mathbf{L}_M$ , noted as  $\mathbf{L}_M = \{\mathbf{L}_{(1)to}, \mathbf{L}_{(2)to}, \dots, \mathbf{L}_{(m)to}\}^T$ . The topological vector  $\mathbf{L}_{(i)to}$  and label  $L_{(i,j)}$  in the matrix  $\mathbf{L}_M$  meet the conditions:  $\mathbf{L}_{(i)} \neq \emptyset \wedge \mathbf{L}_{(i)to} \neq \emptyset \wedge \mathbf{L}_M \neq \emptyset$ ,  $\forall \mathbf{L}_{(i1)} \cap \forall \mathbf{L}_{(i2)} = \emptyset$ ,  $\forall \mathbf{L}_{(i1)to} \cap \forall \mathbf{L}_{(i2)to} = \emptyset \wedge \forall L_{(i,j1)} \cap \forall L_{(i,j2)} = \emptyset$ ,  $\forall L_{(i,j1)} \cap \forall L_{(i,j2)} = \emptyset$ ,  $\mathbf{L}_{(i)} \subseteq \mathbf{L}_{(i)to} \subseteq \mathbf{L}_M$ .

Def 3.3 Feature attribute label word frequency matrix  $\mathbf{F}_{q(L_M)}$ . In the process of text mining, the appearance time for the label  $\forall L_{(i)}$  or  $\forall L_{(i,j)}$  of the matrix  $\mathbf{L}_M$  is defined as the feature attribute label word frequency  $\mathbf{F}_{q(L_{(i)})}$  or  $\mathbf{F}_{q(L_{(i,j)})}$ . As to the element mark  $i$  and  $j$  of the attribute label in the matrix  $\mathbf{L}_M$ , the  $m \times n$  dimension matrix  $\mathbf{F}_{q(L_M)}$  with the feature attribute label word frequency  $\mathbf{F}_{q(L_{(i)})}$  or  $\mathbf{F}_{q(L_{(i,j)})}$  is defined as the feature attribute label word frequency matrix  $\mathbf{F}_{q(L_M)}$ .

Def 3.4 Tourist feature attribute factor  $f_{(k)to}$ , feature attribute vector  $\mathbf{F}_{(i,j)to}$ , feature attribute matrix  $\mathbf{F}_{Mto}$  and feature attribute factor weighted parameter  $\delta_{(k)to}$ . Tourists usually predetermine certain conditions, such as the budget and cost, the traveling time, tourist attraction hot index, tour purpose, and transportation mode. Define the preset conditions as the tourist feature attribute factor  $f_{(k)to}$ . Set up a  $\max k \times 1$  dimension matrix  $\mathbf{F}_{(i,j)to}$ . According to the code sequence  $k$  of the feature attribute factor  $f_{(k)to}$ , store the factor  $f_{(k)to}$  into the element  $f_{(i)to}$  of the vector  $\mathbf{F}_{(i,j)to}$ , and the vector  $\mathbf{F}_{(i,j)to}$  is defined as the tourist feature attribute vector. Of all the factors  $f_{(k)to}$ , the budget and cost factor is defined as the weighted expense for the tourists (Unit: yuan). The traveling time factor is defined as the weighted traveling time for the tourists (Unit: hour). The tourist attraction hot index is defined as the weighted capability of the tourist attraction to attract tourists. The tour purpose factor is defined as the purpose for which the tourists visit the tourism city [30–33]. The transportation mode factor is defined as the transportation method that the tourists tend to use. For convenience of calculation, quantify the textual tourist feature attribute factor  $f_{(k)to}$  into numerical data. In order to ensure that the impact of each tourist feature attribute factor on the modeling is in the same order of magnitude, the normalization is performed on the factors by the weighted parameter, defined as the feature attribute factor weighted parameter  $\delta_{(k)to}$ . Perform topology on each element  $F_{(i)to}$  of the vector  $\mathbf{F}_{(i,j)to}$  and form the topology vector  $\mathbf{F}_{(i)to}$ ; all the vectors  $\mathbf{F}_{(i)to}$  form the matrix  $\mathbf{F}_{Mto}$ , defined as the feature attribute matrix. The matrix  $\mathbf{F}_{Mto}$  is the topological matrix that contains the tourists' alternative needs and requirements. Define the vector  $\mathbf{F}_{(i,j)to}$  and its factor  $f_{(i,j)to}$  as follows.

- Tourist feature attribute vector  $\mathbf{F}_{(i,j)to}$ :  $\{F_{(1)to}$ : tour budget and cost;  $F_{(2)to}$ : traveling time;  $F_{(3)to}$ : tourist attraction hot index;  $F_{(4)to}$ : tour purpose;  $F_{(5)to}$ : transportation mode};
- Tour budget and cost  $\mathbf{F}_{(1)to}$ :  $\{F_{(1,1)to}: f_{(1,1)to} \in [0, 50]$ ;  $F_{(1,2)to}: f_{(1,2)to} \in (50, 100]$ ;  $F_{(1,3)to}: f_{(1,3)to} \in (100, 150]$ ;  $F_{(1,4)to}: f_{(1,4)to} \in (150, +\infty)\}$ ,  $f_{(i,j)to} \subset \mathbf{R}^+$ ;
- Traveling time  $\mathbf{F}_{(2)to}$ :  $\{F_{(2,1)to}: f_{(2,1)to} \in (0, 1.00]$ ;  $F_{(2,2)to}: f_{(2,2)to} \in (1.00, 2.00]$ ;  $F_{(2,3)to}: f_{(2,3)to} \in (2.00, 3.00]$ ;  $F_{(2,4)to}: f_{(2,4)to} \in (3.00, +\infty)\}$ ,  $f_{(i,j)to} \subset \mathbf{R}^+$ ;
- Tourist attraction hot index  $\mathbf{F}_{(3)to}$ :  $\{F_{(3,1)to}: f_{(3,1)to} \in (0, 0.25]$ ;  $F_{(3,2)to}: f_{(3,2)to} \in (0.25, 0.50]$ ;  $F_{(3,3)to}: f_{(3,3)to} \in (0.50, 0.75]$ ;  $F_{(3,4)to}: f_{(3,4)to} \in (0.75, 1.00)\}$ ,  $f_{(i,j)to} \subset \mathbf{R}^+$ ;
- Tour purpose  $\mathbf{F}_{(4)to}$ :  $\{F_{(4,1)to}$ : leisure, 0.25;  $F_{(4,2)to}$ : health care, 0.50;  $F_{(4,3)to}$ : on vacation, 0.75;  $F_{(4,4)to}$ : business affairs, 1.00};
- Transportation mode  $\mathbf{F}_{(5)to}$ :  $\{F_{(5,1)to}$ : taking taxi, 0.25;  $F_{(5,2)to}$ : cycling, 0.50;  $F_{(5,3)to}$ : walking, 0.75;  $F_{(5,4)to}$ : taking the public bus, 1.00}.

Set up the matrix  $\mathbf{F}_{Mto}$ . Tourists confirm their interests by the matrix  $\mathbf{F}_{Mto}$ . The intelligent system confirms the nearest neighborhood historical tourists and then recommends the preferred tourist attraction classifications. Equation (1) is the founded  $5 \times 4$  dimension matrix  $\mathbf{F}_{Mto}$ ; and each row of the matrix represents one vector  $\mathbf{F}_{(i)to}$ , and its elements are the specific quantified values of  $f_{(k)to}$ .

$$\mathbf{F}_{Mto} = \begin{bmatrix} F_{(1,1)to} & F_{(1,2)to} & F_{(1,3)to} & F_{(1,4)to} \\ F_{(2,1)to} & F_{(2,2)to} & F_{(2,3)to} & F_{(2,4)to} \\ F_{(3,1)to} & F_{(3,2)to} & F_{(3,3)to} & F_{(3,4)to} \\ F_{(4,1)to} & F_{(4,2)to} & F_{(4,3)to} & F_{(4,4)to} \\ F_{(5,1)to} & F_{(5,2)to} & F_{(5,3)to} & F_{(5,4)to} \end{bmatrix}, F_{(i,j)to} \sim f_{(i,j)to} \quad (1)$$

Def 3.5 Neighborhood tourist searching objective function  $G_{(T(hi),T(cu))}$ . According to the definition of the Euclidean distance, the neighborhood relationship model between

the historical tourists and current tourists is set up, shown as Equation (2). The weighted parameter  $\delta_{(i)to}$  is brought into the objective function  $G_{(T(hi),T(cu))}$  as the regulatory factor,  $i \in (0, \max k] \subset \mathbb{Z}^+$ ,  $\delta_{(i)to} \in (0, 1] \subset \mathbb{R}^+$ . Function  $G_{(T(hi),T(cu))}$  represents the close relationship between the current tourists and historical tourists.

$$G_{(T(hi),T(cu))} = \left[ \sum_{i=1}^{\max k} \left| \delta_{(i)to} (f_{(i,j_{hi})to}^{T(hi)} - f_{(i,j_{cu})to}^{T(cu)}) \right|^2 \right]^{1/2} \quad s.t. \forall j_{hi}, j_{cu} \quad (2)$$

The tourist attraction recommendation model based on the weighted collaborative filtering is set up, and it is defined as Algorithm 3. Below is the Algorithm 3 pseudo-code.

---

**Algorithm 3** Tourist attraction recommendation algorithm based on the weighted collaborative filtering

---

- 1: **Step 1:** Set up tourist attraction feature attribute label word frequency matrix  $F_{q(LM)}$  and word frequency storage vector  $TN_{(F_{q(LM)})}$ .
  - 2:     **Sub-step 1:** Initialize the word frequency matrix  $F_{q(LM)}$ .
  - 3:     **Sub-step 2:** Set up the evaluation data set  $F_{text}$  for the historical tourists  $T_{(hi)}$ .
  - 4:     **Sub-step 3:** Search word frequency  $F_{q(LM)roi} \sim F_{q(L(i,j))}$  of each row's vector  $L_{(i)to}$  in the matrix  $F_{q(LM)}$ . Define the total number of each label  $L(i, j)$  word frequency  $F_{q(L(i,j))}$  in the no.  $i$  vector  $L_{(i)to}$  as  $TN_{(F_{q(LM)roi})}$ .
  - 5:     **Sub-step 4:** Form matrix  $F_{q(LM)}$  via  $L_{(i)to}$ . Output the vector  $TN_{(F_{q(LM)})}$  containing each row's vector  $L_{(i)to}$  total word frequency.
  - 6: **Step 2:** Set up the tourist attraction classification interest degree set  $T_{(hi)(u)} \sim TN_{(F_{q(LM)})(u)}$  of the historical tourists. The number of historical tourists:  $\sigma$ ; the no.  $u$  historical tourists:  $T_{(hi)(u)}$ .
  - 7: **Step 3:** Set up the tourist sight classification recommendation model based on the interest degree vector  $TN_{(F_{q(LM)})(u)}$  of the nearest neighborhood historical tourists.
  - 8:     **Sub-step 1:** Confirm each interest degree vector  $TN_{(F_{q(LM)})(vi)}$  and its elements of the historical tourists related to the no.  $v_1 \sim v_{n(Lo)}$  location points.
    - Sub-step 2:** Set up the tourist attraction classification recommendation vector  $R$ .
    - Sub-step 3:** Define the recommendation degree function  $R_{(j)}$ . The no.  $j$  element of vector  $TN_{(F_{q(LM)})(vi)}$  relates to one kind of tourist attraction classification word frequency. The average value of the no.  $j$  element word frequencies calculated by the  $v_{n(Lo)}$  number of neighborhood historical tourists  $T_{(hi)(vi)}$  is defined as the recommendation function.
 
$$R_{(j)} = \frac{\sum_{i=1}^{n(Lo)} n(TN_{(F_{q(LM)})(vi,j)})}{n(Lo)}$$
    - Sub-step 4:** Calculate  $R_{(j)}$  value for the  $n(Lo)$  number of location points  $Lo(T_{(hi)(vi)})$ . Store the values in the descending order in vector  $R$ .
  - Step 5:** Set up the precise tourist attraction recommendation algorithm based on the tourist attraction search-optimized generating space.
    - Sub-step 1:** Confirm the starting point  $S_t$  of the tour route and the tourist attraction number  $n_{(i)}$  for the tourists.
    - Sub-step 2:** Confirm the cellular generating unit  $C_{(k)}$  containing the starting point  $S_t$ .
    - Sub-step 3:** Set up the Open list  $O_{(k)}$  and the Closed list  $C_{(k)}$  for the tourist attraction.
    - Sub-step 4:** Confirm the number of common edges of  $C_{(k)nei(s)}$  containing  $S_{t(cu)}$  and its related unit  $C_{(k)2nei(s)}$  and the cellular core  $O_{(k)2nei(s)}$ . Search  $O_{(k)2nei(s)}$  and store the feasible ones in  $R_{to}$ . Delete it from the list  $O_{(k)}$ , and store it in the list  $C_{(k)}$ . The current tourist attraction  $C_{(k)2nei(s)}$  is searched and stored in the vector  $R_{to}$ .
- 

Figure 4 is an example for Algorithm 3. Calculate the  $R_{(j)}$  value for the historical tourists. Then, determine the closest ones for the current tourist by  $G_{(T(hi),T(cu))}$ . From the closest historical tourists' choice (average value), search the nearby cluster from the starting point (green point). If the nearest point  $a$  in cluster  $C_1$  meets the needs, choose it (with square); see Figure 4a. Continue searching—if the next nearest point  $b$  in cluster  $C_4$  meets the needs, choose it (with square); see Figure 4b. Continue searching—if the next nearest point  $c$  in cluster  $C_3$  meets the needs, choose it (with square); see Figure 4c. If the quantity is sufficient, stop searching and obtain the precise points  $a$ ,  $b$ , and  $c$ .

The above steps are modeled to form the vector  $R_{to}$  and finally output the precise tourist attractions for the tourists. Based on the precise tourist attractions, proper tour routes should be recommended to the tourists.

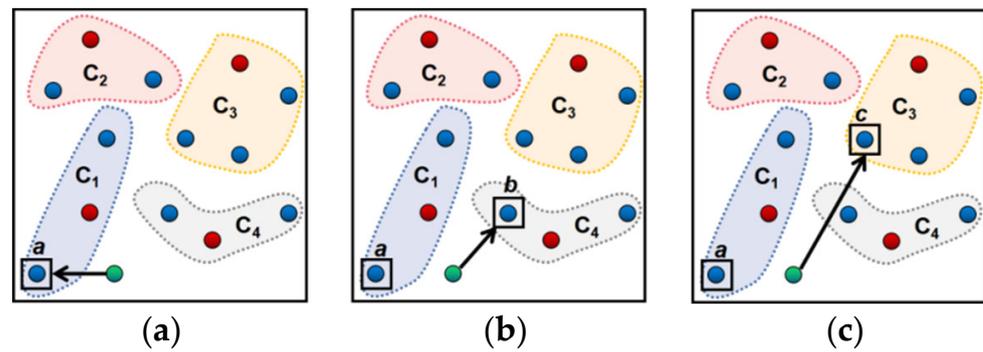


Figure 4. An example for Algorithm 3. (a–c) shows the searching process of points  $a$ ,  $b$ , and  $c$ .

### 3. The Optimal Tour Route Recommendation Model Based on Precise Tourist Attraction Approach Vector Algorithm

In an optimal tour route, tourist attractions all satisfy tourists' interests. The traveling cost should be the lowest [34–37]. The total cost of visiting tourist attractions can be seen as a fixed value. Thus, the travel cost should be considered; it is determined by the transportation mode, traveling distance, road congestion, and the traveling fee, etc. When tourists choose one transportation mode, the connecting route, traveling distance, road congestion condition, and the traveling fee are all different. Thus, the lowest road congestion index and the smallest traveling fee would be ideal [38–40]. Below is the fourth group of definitions and the modeling process of the optimal tour route recommendation based on the precise tourist attraction approach vector algorithm.

Def 4.1 Tourist attraction neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ . Two arbitrary tourist attractions  $\mathbf{R}_{to(i)}$  and  $\mathbf{R}_{to(j)}$  form a road section, defined as  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ .

Def 4.2 Neighborhood section edge weight model  $g(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  and the global tour route weight iteration model  $G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ . The model formed by the three constraint factors of the shortest travel distance  $\zeta_1(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  (Unit: km), the road congestion index  $\zeta_2(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  and the traveling fee  $\zeta_3(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  (Unit: ¥yuan) to determine the neighborhood edge weight value is defined as the neighborhood section edge weight model. Usually, the factors are normalized into (0, 1). Set the amended parameter of the constraint factor  $\zeta^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  as  $\varepsilon^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ . Then, Equation (3) shows the neighborhood section weighted edge weight model  $g(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ .

$$g(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) = \sum_{i=1}^3 \varepsilon^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}) \cdot \zeta^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}) \quad (3)$$

In Equation (3), the factor  $\zeta_1(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  is determined by the approaching vector algorithm. The factor  $\zeta_2(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  is obtained from the city statistical transportation information data. The traveling fee factor  $\zeta_3(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  is determined by the chosen transportation mode  $\tau_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  and the actual cost generated by the travel process in the neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ . If one tour route is formed by the  $z$  number of neighborhood sections  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  connected by the starting point  $\mathbf{R}_{to(0)}$  and each terminal point  $\mathbf{R}_{to(i)}, i \in (0, n] \subset \mathbb{Z}^+$ , the integral route model formed by the connecting  $z$  number of neighborhood sections  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  is defined as the global tour route weight iteration model  $G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ , shown in Equation (4). The optimal solution of the model  $G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  is the global minimum value  $\min G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ .

$$\begin{cases} G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) = \sum_{z=1}^{\max z} g^{(z)}(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) \\ G(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) = \sum_{z=1}^{\max z} \sum_{i=1}^3 \varepsilon^{(z,i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}) \cdot \zeta^{(z,i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}) \end{cases} \quad (4)$$

Def 4.3 Tourist attraction geospatial directed weighted graph  $D$ . Take all the element tourist attractions  $\mathbf{R}_{to(i)}$  in the vector  $\mathbf{R}_{to}$  as the vertexes of the graph. Take the neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  between two tourist attractions as the directed edge, and each edge's weight value is set as  $g(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ . The closed graph  $D = \langle V, A \rangle$  that is formed by the vertex set  $V = \mathbf{R}_{to}$  and weight edge set  $A = Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  is defined as the tourist attraction geospatial directed weighted graph  $D$ .

Def 4.4 The approaching model  $f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  of the tourist attraction neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ . When the tourists travel between  $\mathbf{R}_{to(i)}$  and  $\mathbf{R}_{to(j)}$ , the shortest path is the road section combination that most closely approaches one certain fitting function curve in the trend between two points [41–43]. The  $\mathbf{R}_{to(i)}$  and  $\mathbf{R}_{to(j)}$  are set as the terminal points to fit the approaching function  $f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ , defined as the approaching model of the section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ . The order of the function is determined by the algorithm complexity and the actual geospatial data. It intersects with certain city roads between the points  $\mathbf{R}_{to(i)}$  and  $\mathbf{R}_{to(j)}$ . Equation (5) is the approaching model of the tourist attraction neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ , in which  $p$  is the function order,  $a_{(p)}$  and  $b$  are the function coefficients, and the footnote  $\mathbf{R}_{to}$  is the selected point set.

$$y_{\mathbf{R}_{to(2)}} = a_{(1)}y_{\mathbf{R}_{to(1)}}^p + a_{(2)}y_{\mathbf{R}_{to(1)}}^{p-1} + \dots + a_{(p)}y_{\mathbf{R}_{to(1)}} + b \quad (5)$$

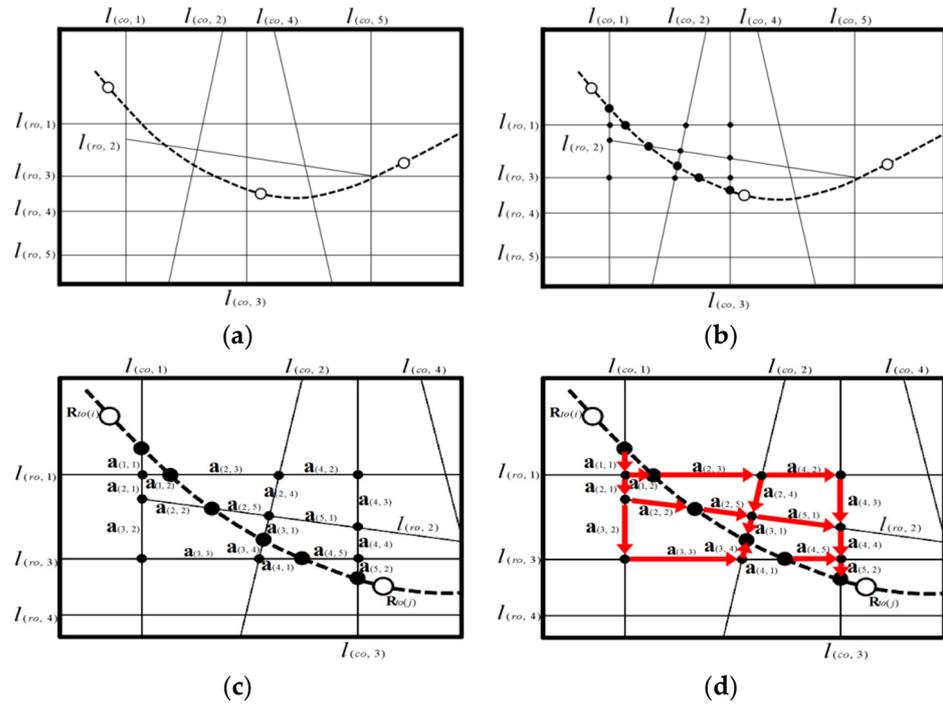
Def 4.5 The access road set  $\mathbf{L}$  and the road intersect point set  $\mathbf{K}$ . Divide the city roads into the horizontal road and the vertical road. The horizontal road set is confirmed as  $\mathbf{L}_{(ro)}$ , and its element is the road  $l_{(ro,i)}$ . The vertical road set is confirmed as  $\mathbf{L}_{(co)}$ , and its element is the road  $l_{(co,j)}$ . The intersection point is noted as  $K_{(i,j)}$ , in which the  $i$  is the mark for the horizontal road  $l_{(ro,i)}$ , and the  $j$  is the mark for the vertical road  $l_{(co,j)}$ . If the two roads are both horizontal or vertical, the intersection point is noted as  $K_{s(i,j)}$ .

Def 4.6 The approaching point set  $\mathbf{P}$ . The function curve  $f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  on the tourist attraction neighborhood section  $Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  intersects with certain horizontal roads  $l_{(ro,i)}$  or  $l_{(co,j)}$  to form intersection points  $P_{(f,l)(i)}$ , defined as the approaching point set  $\mathbf{P}$ . The shortest route is searched using the method.

Set up  $f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ . Quantify set  $\mathbf{L}$ , including the elements  $l_{(ro,i)}$  and  $l_{(co,j)}$ . The quantify function of the horizontal and vertical road element is  $f_{(l_{(ro,i)})}^{(1)}$  and  $f_{(l_{(co,j)})}^{(2)}$ . The intersection points between  $\nabla f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  and  $f_{(l_{(ro,i)})}^{(1)}$  or  $\nabla f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  and  $f_{(l_{(co,j)})}^{(2)}$  form the set  $\mathbf{P}(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  of the shortest route. The intersection points of the function  $f_{(l_{(ro,i)})}^{(1)}$  and  $f_{(l_{(co,j)})}^{(2)}$  form the set  $\mathbf{K}(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ , satisfying Equation (6).

$$\begin{cases} (\nabla f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) \cap f_{(l_{(ro,i)})}^{(1)}) \cup (\nabla f(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) \cap f_{(l_{(co,j)})}^{(2)}) = \mathbf{P}(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) \\ (f_{(l_{(ro,i)})}^{(1)} \cap f_{(l_{(ro,-i)})}^{(1)}) \cup (f_{(l_{(co,j)})}^{(2)} \cap f_{(l_{(co,-j)})}^{(2)}) \cup (f_{(l_{(ro,i)})}^{(1)} \cap f_{(l_{(co,j)})}^{(2)}) = \mathbf{K}(Zo_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}) \end{cases} \quad (6)$$

Visualize all the elements in Figure 5. Figure 5a shows the relationship between the connecting lines formed by certain tourist attractions and the city roads. Figure 5b shows the intersection point set. Figure 5c is the abstracted intersection point set's geospatial distribution of the approaching algorithm between two tourist attractions. Figure 5d shows the searching direction of the approaching vector.



**Figure 5.** The point set spatial distribution of the tourist attraction approaching vector searching algorithm and the dynamic graph of the searching vectors. (a) shows the relationship between the connecting lines and the city roads. (b) shows the intersection point set. (c) is the abstracted intersection point set's geospatial distribution. (d) shows the searching direction of the approaching vector.

Set up the closest approach vector algorithm.

Step 1 Take the closest neighborhood approach points  $P_{(f,l)(1)}$  and  $P_{(f,l)(maxi)}$  for the terminal points  $R_{to(i)}$  and  $R_{to(j)}$  in the set  $P(Z_{o(R_{to(i)}, R_{to(j)})})$ .

Step 2 Search the approach point set  $P(Z_{o(R_{to(i)}, R_{to(j)})})$  and the set  $K(Z_{o(R_{to(i)}, R_{to(j)})})$ .

- ① Search the route  $S_{(1,u)}$  from the point  $P_{(f,l)(1)}$  to  $P_{(f,l)(2)}$ . Start from the point  $P_{(f,l)(1)}$  to search  $l_{(co,1)}$  to the point  $K_{(1,1)}$  and form the vector  $a_{(1,1)}$ ; search  $l_{(ro,1)}$  to the point  $P_{(f,l)(2)}$  and form the vector  $a_{(1,2)}$ . There is no other access route, the searching process is completed, the route is  $S_{(1,1)}$ , and the approach vector set is  $A_{(1,1)} = \{a_{(1,1)}, a_{(1,2)}\}$ .
- ② Search the route  $S_{(2,u)}$  from the point  $P_{(f,l)(1)}$  to  $P_{(f,l)(3)}$ . Start from the point  $K_{(1,1)}$  to search  $l_{(co,1)}$  to the point  $K_{(2,1)}$  and form the vector  $a_{(2,1)}$ ; search  $l_{(ro,2)}$  to the point  $P_{(f,l)(3)}$  and form the vector  $a_{(2,2)}$ ; the route is  $S_{(2,1)}$ , and form the vector set  $A_{(2,1)} = \{a_{(1,1)}, a_{(2,1)}, a_{(2,2)}\}$ . If it is judged that there is another access route, continue searching. Start from the point  $P_{(f,l)(2)}$  to search  $l_{(ro,1)}$  to the point  $K_{(1,2)}$  and form the vector  $a_{(2,3)}$ ; start from the point  $K_{(1,2)}$  to search  $l_{(co,2)}$  to the point  $K_{(2,2)}$  and form the vector  $a_{(2,4)}$ ; start from the point  $K_{(2,2)}$  to search  $l_{(ro,2)}$  to the point  $P_{(f,l)(3)}$  and form the vector  $a_{(2,5)}$ ; the route is  $S_{(2,2)}$ , and form the vector set  $A_{(2,2)} = \{a_{(1,1)}, a_{(1,2)}, a_{(2,3)}, a_{(2,4)}, a_{(2,5)}\}$ . If it is judged that there is no other access route, the searching process is completed. Judge  $S_{(2,1)}$  and  $S_{(2,2)}$ .
  - (i) If  $S_{(2,1)} \geq S_{(2,2)}$ , retain the access road of the  $S_{(2,2)}$  and its approach vector  $A_{(2,2)}$ .
  - (ii) If  $S_{(2,1)} < S_{(2,2)}$ , retain the access road of the  $S_{(2,1)}$  and its approach vector  $A_{(2,1)}$ .
- ③ Search the route  $S_{(3,u)}$  from the point  $P_{(f,l)(1)}$  to  $P_{(f,l)(4)}$ . Start from the point  $K_{(2,2)}$  to search  $l_{(co,2)}$  to the point  $P_{(f,l)(4)}$  and form the vector  $a_{(3,1)}$ ; the route

is  $S_{(3,1)}$ , and form the vector set  $\mathbf{A}_{(3,1)} = \{\mathbf{a}_{(1,1)}, \mathbf{a}_{(1,2)}, \mathbf{a}_{(2,3)}, \mathbf{a}_{(2,4)}, \mathbf{a}_{(3,1)}\}$ . If it is judged that there is another access road, continue searching. Start from the point  $K_{(2,1)}$  to search  $l_{(ro,2)}$  and  $l_{(co,2)}$ , pass through the points  $P_{(f,l)(3)}$  and  $K_{(2,2)}$  and reach  $P_{(f,l)(4)}$ , form the vector set  $\mathbf{A}_{(3,2)} = \{\mathbf{A}_{(2,1)}, \mathbf{a}_{(2,5)}, \mathbf{a}_{(3,1)}\}$ , and the route is  $S_{(3,2)}$ . Start from the point  $K_{(2,1)}$  to search  $l_{(co,1)}$ ,  $l_{(ro,3)}$ ,  $l_{(co,2)}$ , pass through the points  $K_{(3,1)}$  and  $K_{(3,2)}$  and reach  $P_{(f,l)(4)}$ , form the vector set  $\mathbf{A}_{(3,3)} = \{\mathbf{a}_{(1,1)}, \mathbf{a}_{(2,1)}, \mathbf{a}_{(3,2)}, \mathbf{a}_{(3,3)}, \mathbf{a}_{(3,4)}\}$ , and the route is  $S_{(3,3)}$ . If it is judged that there is no other access route, the searching process is completed. Compare the three values  $S_{(3,1)}$ ,  $S_{(3,2)}$ , and  $S_{(3,3)}$ .

- (i) If the  $S_{(3,1)}$  is the minimum one, retain the access road of the  $S_{(3,1)}$  and its approach vector  $\mathbf{A}_{(3,1)}$ .
  - (ii) If the  $S_{(3,2)}$  is the minimum one, retain the access road of the  $S_{(3,2)}$  and its approach vector  $\mathbf{A}_{(3,2)}$ .
  - (iii) If the  $S_{(3,3)}$  is the minimum one, retain the access road of the  $S_{(3,3)}$  and its approach vector  $\mathbf{A}_{(3,3)}$ .
- ④ Continue searching:
- (i) If searching is complete, return to step ①~③ and continue searching the route  $S_{(i-1,u)}$  between the point  $P_{(f,l)(1)}$  and  $P_{(f,l)(i)}$ . Choose the access road of the  $\min S_{(i-1,u)}$  and  $\mathbf{A}_{(i-1,u)}$ .
  - (ii) If searching is not complete, continue searching the route  $S_{(\max i-1,u)}$  between the point  $P_{(f,l)(1)}$  and  $P_{(f,l)(\max i)}$ . Choose the access road of the  $\min S_{(\max i-1,u)}$  and  $\mathbf{A}_{(\max i-1,u)}$ . Output  $\min S_{(\max i-1,u)}$  as the shortest access road.
- ⑤ Confirm the set  $\nabla \mathbf{P}_{(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))}$  according to the route  $\min S_{(\max i-1,u)}$ .

Due to the constraints on the traveling time, physical condition and traveling cost, the number of tourist attractions is no more than 10. When the node number is  $n = 10$ , the maximum operation time complexity  $O(n)$  is 3.62 s. Thus, the algorithm time is feasible, and the exhaustive method on the heap sort is used to search the global optimal solution  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$ . The  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  is stored in the last element of the last level in the maximum heap.

#### 4. Experimental Results and Data Analysis

For the experiment, 20 typical tourist attractions in Zhengzhou city's downtown area are selected. They are divided into four categories:  $\mathbf{T}_{(1)}$ : Venue and memorial;  $\mathbf{T}_{(2)}$ : Park and greenland;  $\mathbf{T}_{(3)}$ : Amusement park;  $\mathbf{T}_{(4)}$ : Leisure shopping. The experiment is performed on a real-world environment; the original basic data such as longitude and latitude data, the data of road distance, travel time, traveling fee, and road congestion index, etc., are obtained from the Zhengzhou city's geospatial database, the GIS website and electronic maps [44–46]. Thus, the final results are obtained based on real-world data.

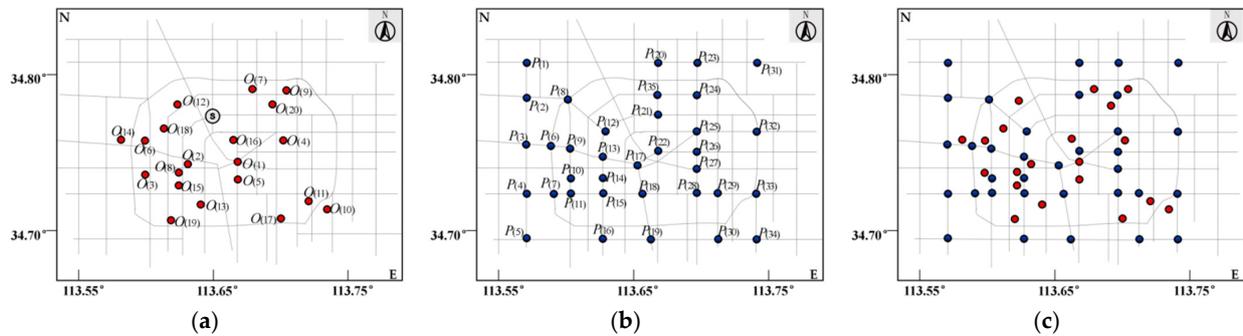
##### 4.1. Basic Experimental Data Collection, Calculation and Analysis

The required basic data include the tourist attraction name and classification, latitude and longitude coordinates, tourists' evaluation textual data and feature attribute data, geographic information data and traffic information data, etc.

###### 4.1.1. The Basic Data of the Tourist Attraction Domain

Set up  $X$  with element  $x_{(k)}$ .  $X = \{x_{(1)}: \text{ErQi Memorial}; x_{(2)}: \text{Bishagang Park}; x_{(3)}: \text{Zhongyuan Wanda}; x_{(4)}: \text{Zijingshan Park}; x_{(5)}: \text{Dehua Pedestrian street}; x_{(6)}: \text{CC mall}; x_{(7)}: \text{Henan provincial museum}; x_{(8)}: \text{Zhengzhou science and technology museum}; x_{(9)}: \text{Zhengzhou zoo}; x_{(10)}: \text{Zhongyuan tower}; x_{(11)}: \text{Century amusement park}; x_{(12)}: \text{Children fun park}; x_{(13)}: \text{ErQi Wanda}; x_{(14)}: \text{Xiliuhu park}; x_{(15)}: \text{Zhengzhou museum}; x_{(16)}: \text{Renmin}$

park;  $x_{(17)}$ : Zhengxin park;  $x_{(18)}$ : Zhengzhou Wangfujing;  $x_{(19)}$ : Dihu amusement park;  $x_{(20)}$ : Guomao}. Set up the  $O_{(k)}$  and  $P_{(v)}$ . Figure 6a shows the distribution of each  $O_{(k)}$ . Figure 6b is the distribution of  $P_{(v)}$ . Figure 6c is the distribution that overlays the set  $O_{(k)}$  and  $P_{(v)}$ .



**Figure 6.** The distribution of tourist attraction cellular core set and road intersection point set and their overlying map. (a,b) show the distribution of tourist attractions and road intersections. (c) shows all the points.

#### 4.1.2. The Basic Information Data of the Tourist Attractions and Location Points

The latitude and longitude website GPSspg is used to obtain the coordinates of the tourist attractions and location points, shown in Table 1.

**Table 1.** The coordinate data of the tourist attraction cellular unit  $x_{(k)} \sim O_{(k)}$  and location points  $P_{(v)}$ .

$l, B$		$l, B$		$l, B$		$l, B$	
$O_{(1)}$	113.673, 34.757	$O_{(15)}$	113.627, 34.745	$P_{(9)}$	113.603, 34.763	$P_{(23)}$	113.682, 34.826
$O_{(2)}$	113.637, 34.758	$O_{(16)}$	113.663, 34.761	$P_{(10)}$	113.603, 34.742	$P_{(24)}$	113.682, 34.786
$O_{(3)}$	113.607, 34.752	$O_{(17)}$	113.690, 34.698	$P_{(11)}$	113.603, 34.736	$P_{(25)}$	113.682, 34.763
$O_{(4)}$	113.695, 34.767	$O_{(18)}$	113.613, 34.762	$P_{(12)}$	113.631, 34.763	$P_{(26)}$	113.682, 34.756
$O_{(5)}$	113.671, 34.757	$O_{(19)}$	113.615, 34.715	$P_{(13)}$	113.629, 34.757	$P_{(27)}$	113.682, 34.752
$O_{(6)}$	113.609, 34.763	$O_{(20)}$	113.681, 34.784	$P_{(14)}$	113.629, 34.742	$P_{(28)}$	113.682, 34.737
$O_{(7)}$	113.678, 34.790	$P_{(1)}$	113.568, 34.826	$P_{(15)}$	113.629, 34.736	$P_{(29)}$	113.703, 34.737
$O_{(8)}$	113.627, 34.746	$P_{(2)}$	113.568, 34.808	$P_{(16)}$	113.629, 34.695	$P_{(30)}$	113.703, 34.695
$O_{(9)}$	113.685, 34.789	$P_{(3)}$	113.568, 34.794	$P_{(17)}$	113.649, 34.754	$P_{(31)}$	113.726, 34.826
$O_{(10)}$	113.729, 34.723	$P_{(4)}$	113.568, 34.736	$P_{(18)}$	113.656, 34.737	$P_{(32)}$	113.726, 34.756
$O_{(11)}$	113.720, 34.730	$P_{(5)}$	113.568, 34.695	$P_{(19)}$	113.665, 34.695	$P_{(33)}$	113.726, 34.737
$O_{(12)}$	113.618, 34.781	$P_{(6)}$	113.592, 34.772	$P_{(20)}$	113.667, 34.826	$P_{(34)}$	113.726, 34.695
$O_{(13)}$	113.643, 34.717	$P_{(7)}$	113.585, 34.736	$P_{(21)}$	113.667, 34.774	$P_{(35)}$	113.667, 34.787
$O_{(14)}$	113.589, 34.779	$P_{(8)}$	113.594, 34.779	$P_{(22)}$	113.667, 34.756		

#### 4.1.3. Data and Result Analysis

The tourist attractions are all the typical ones in Zhengzhou city. They are distributed discretely and connected by the city roads. The selection process is fair and balanced in that the quantity of each classification is commensurate. The tourist attractions are all surrounded by different levels of roads; thus, there are sufficient location points. The latitude and longitude of all the points are collected to quantify their locations. The coordinates cause the points to have spatial attributes. They are used in the model to generate the cellular space. In regard to data collection and point confirmation, the original data set selected in the experiment is accurate and feasible and contains real-world data. Therefore, this experiment is a real-world-based experiment and not a simulation experiment, and its output result could be directly used for tourist attraction and tour route recommendation.

#### 4.2. The Output Result and Analysis of the Tourist Attraction Spatial Clusters

The tourist attraction spatial cluster  $C_{l(u)}$  has the basic structural set-up, with a cellular core  $O_{(k)}$ , cellular unit  $C_{(k)}$  and the formed cellular space  $C$ . It is obtained by multiple searching processes. The spatial cluster  $C_{l(u)}$  is generated to search the optimal tourist attractions in terms of spatial distance.

##### 4.2.1. The Output Result of the Tourist Attraction Cluster

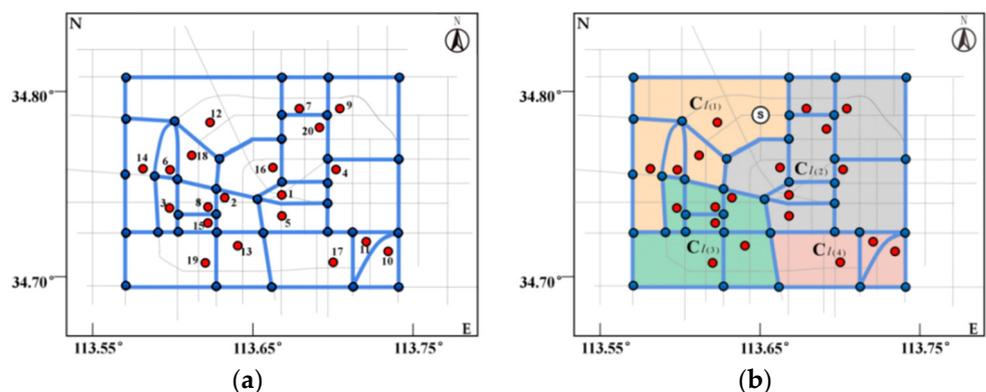
According to the tourist attraction cellular unit set  $O_{(k)}$  and the location point set  $P_{(v)}$ , the tourist attraction cellular space generating algorithm is performed and the Closed list  $C_{l(i)}$  that is used to store the structural location point  $P_{(v)C(k)}$  of the unit  $C_{(k)}$  with the cellular core  $O_{(k)m}$  in the form of level storage is confirmed, shown as the Table 2. The tourist attraction cellular space  $C$  is generated, as shown in Figure 7a, in which the number represents the tourist attraction's code. The tourist attraction clustering algorithm based on geospatial feature attributes is set up and forms the clusters on the space  $C$ . After  $n = 8$  iterations, the tourist attraction space cluster  $C_{l(u)}$  are output, as shown in Figure 7b. The generated clusters are:

$$C_{l(1)} = \{O_{(6)}, O_{(12)}, O_{(14)}, O_{(18)}\}; C_{l(2)} = \{O_{(1)}, O_{(4)}, O_{(5)}, O_{(7)}, O_{(9)}, O_{(16)}, O_{(20)}\};$$

$$C_{l(3)} = \{O_{(2)}, O_{(3)}, O_{(8)}, O_{(13)}, O_{(15)}, O_{(19)}\}; C_{l(4)} = \{O_{(10)}, O_{(11)}, O_{(17)}\}.$$

**Table 2.** The Closed list  $C_{l(i)}$ , which stores the structural location point  $P_{(v)C(k)}$  of the unit  $C_{(k)}$  with the cellular core  $O_{(k)m}$ .

	$P_{(v)C(k)}$		$P_{(v)C(k)}$
$O_{(1)}$	$P_{(17)}P_{(22)}P_{(26)}P_{(27)}$	$O_{(11)}$	$P_{(29)}P_{(30)}P_{(33)}$
$O_{(2)}$	$P_{(13)}P_{(14)}P_{(15)}P_{(17)}P_{(18)}$	$O_{(12)}$	$P_{(1)}P_{(2)}P_{(8)}P_{(12)}P_{(20)}P_{(21)}$
$O_{(3)}$	$P_{(6)}P_{(7)}P_{(9)}P_{(10)}P_{(11)}$	$O_{(13)}$	$P_{(15)}P_{(16)}P_{(18)}P_{(19)}$
$O_{(4)}$	$P_{(25)}P_{(26)}P_{(27)}P_{(28)}P_{(29)}P_{(32)}P_{(33)}$	$O_{(14)}$	$P_{(2)}P_{(3)}P_{(6)}P_{(8)}$
$O_{(5)}$	$P_{(17)}P_{(18)}P_{(27)}P_{(28)}$	$O_{(15)}$	$P_{(10)}P_{(11)}P_{(14)}P_{(15)}$
$O_{(6)}$	$P_{(6)}P_{(8)}P_{(9)}$	$O_{(16)}$	$P_{(12)}P_{(13)}P_{(17)}P_{(21)}P_{(22)}$
$O_{(7)}$	$P_{(20)}P_{(21)}P_{(9)}P_{(23)}P_{(24)}$	$O_{(17)}$	$P_{(18)}P_{(19)}P_{(28)}P_{(29)}P_{(30)}$
$O_{(8)}$	$P_{(9)}P_{(10)}P_{(13)}P_{(14)}$	$O_{(18)}$	$P_{(8)}P_{(9)}P_{(12)}P_{(13)}$
$O_{(9)}$	$P_{(23)}P_{(24)}P_{(25)}P_{(31)}P_{(32)}$	$O_{(19)}$	$P_{(4)}P_{(5)}P_{(7)}P_{(11)}P_{(15)}P_{(16)}$
$O_{(10)}$	$P_{(30)}P_{(33)}P_{(34)}$	$O_{(20)}$	$P_{(21)}P_{(22)}P_{(24)}P_{(25)}P_{(26)}P_{(35)}$



**Figure 7.** The tourist attraction cellular space  $C$  and cluster distribution generated by the algorithm. (a) shows the basic structure of space  $C$  and clusters. (b) shows the cluster results.

#### 4.2.2. The Data Result Analysis

When the starting point is confirmed, the nearest tourist attractions that conform to the tourists' interests are searched. The tourist attraction cellular units that have a common edge have two common location points. The space  $C$  and clusters are generated, and there are four spatially adjacent clusters, in which the cluster  $C_{l(1)}$  includes 4 tourist attractions, the cluster  $C_{l(2)}$  includes 7 tourist attractions, the cluster  $C_{l(3)}$  includes 6 tourist attractions, the cluster  $C_{l(4)}$  includes 3 tourist attractions. As shown in the figures, the tourist attractions in the same cluster have a close spatial relationship and high intimacy; conversely, the attractions that are far apart have a poor spatial relationship and low intimacy. The adjacency relationship between the tourist attractions within one cluster or in different clusters has no correlation to the clusters, and tourist attractions in the same cluster might be adjacent or might not be adjacent; in addition, the tourist attractions in different clusters might be adjacent or might not be adjacent.

#### 4.3. The Output Result and Analysis of the Tourist Attraction Recommendation Based on Weighted Collaborative Filtering Algorithm

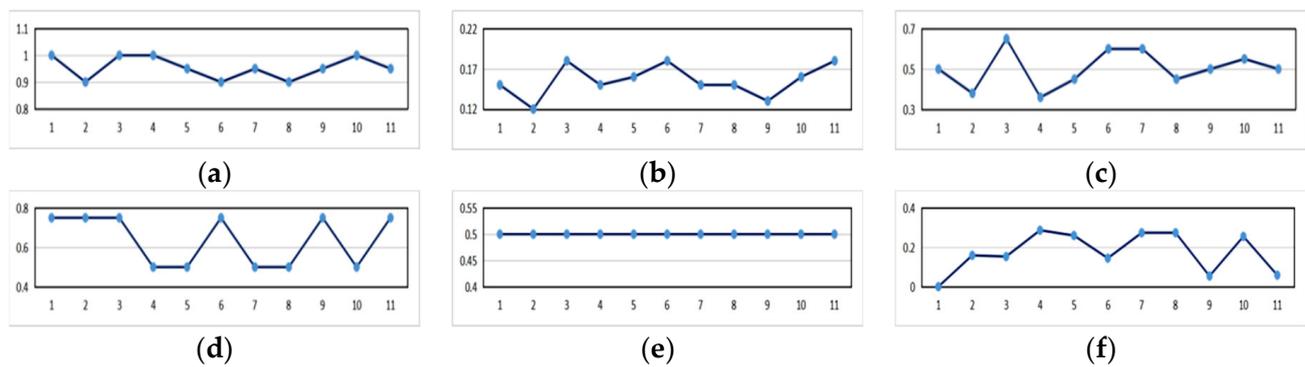
The nearest historical tourists for the current tourist are searched. By calculating the recommendation degree of the nearest historical tourists, the algorithm outputs the priority vector of the tourist attraction classification and then confirms the precise tourist attraction according to the tourists' needs and interests.

##### 4.3.1. The Tourist Attraction Recommendation Result Based on the Weighted Collaborative Filtering Algorithm

Baidu is used to obtain the textual data of the 20 tourist attractions and the crawler technique is used to obtain the evaluation data, including the historical tourists' feature attributes  $f_{(k)to}$  and the tourist attraction classification feature labels  $L_{(i,j)}$ . The list of  $f_{(k)to}$  is used to set up the feature attribute matrix  $F_{Mto}$ . The objective function  $G_{(T_{(hi)}, T_{(cu)})}$  values are calculated between the current tourist and historical tourists. The weighted coefficients are  $\delta_{(1)to} = 0.01$ ,  $\delta_{(2)to} = 0.1$ ,  $\delta_{(3)to} = 1.0$ ,  $\delta_{(4)to} = 1.0$ ,  $\delta_{(5)to} = 1.0$ . The data of the 10 nearest neighborhood historical tourists are selected, as shown in Table 3, including the feature attribute factors of the historical tourists and the current tourist, as well as the searching objective function  $G_{(T_{(hi)}, T_{(cu)})}$  values. Figure 8a–e show the fluctuating curves of the five factors  $f_{(k)to}$  confirmed by the current tourist  $T_{(cu)to}$  and the 10 nearest neighborhood historical tourists  $T_{(hi)to}$ . Figure 8f shows the fluctuating curve of the searching objective function  $G_{(T_{(hi)}, T_{(cu)})}$  in the code sequence by the current tourist  $T_{(cu)to}$  and the 10 nearest neighborhood historical tourists  $T_{(hi)to}$ .

**Table 3.** The feature attribute factors  $f_{(k)to}$  of the historical tourists  $T_{(hi)}$  and the current tourist  $T_{(cu)}$ , as well as the searching objective function  $G_{(T_{(hi)}, T_{(cu)})}$  values.

	$f_{(1)to}$	$f_{(2)to}$	$f_{(3)to}$	$f_{(4)to}$	$f_{(5)to}$	$G_{(T_{(hi)}, T_{(cu)})}$
$T_{(cu)}$	100.00	1.50	0.50	0.75	0.50	0
$T_{(hi,1)}$	90.00	1.20	0.38	0.75	0.50	0.159
$T_{(hi,2)}$	100.00	1.80	0.65	0.75	0.50	0.153
$T_{(hi,3)}$	100.00	1.50	0.36	0.50	0.50	0.287
$T_{(hi,4)}$	95.00	1.60	0.45	0.50	0.50	0.260
$T_{(hi,5)}$	90.00	1.80	0.60	0.75	0.50	0.144
$T_{(hi,6)}$	95.00	1.50	0.60	0.50	0.50	0.274
$T_{(hi,7)}$	110.00	1.50	0.45	0.50	0.50	0.274
$T_{(hi,8)}$	95.00	1.30	0.50	0.75	0.50	0.054
$T_{(hi,9)}$	100.00	1.60	0.55	0.50	0.50	0.255
$T_{(hi,10)}$	95.00	1.80	0.50	0.75	0.50	0.058

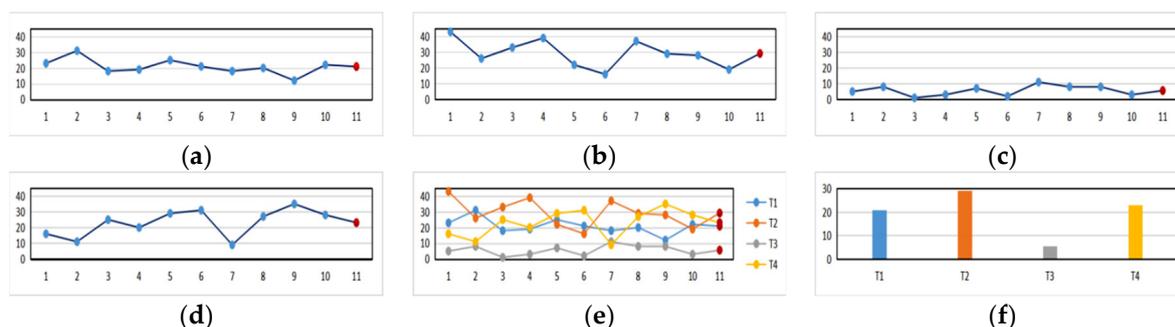


**Figure 8.** The fluctuating curves of the weighted feature attribute factors  $f_{(k)to}$  and the searching objective function  $G_{(T_{(hi)}, T_{(cu)})}$ . (a–f) represent the six column results in Table 4.

**Table 4.** The statistical data and  $R_{(j)}$  of tourist attraction classification topological label vector for neighborhood historical tourists.

	$n(TN(Fq(L_M)_{(vi,1)}))$	$n(TN(Fq(L_M)_{(vi,2)}))$	$n(TN(Fq(L_M)_{(vi,3)}))$	$n(TN(Fq(L_M)_{(vi,4)}))$
$T_{(hi,1)}$	23	43	5	16
$T_{(hi,2)}$	31	26	8	11
$T_{(hi,3)}$	18	33	1	25
$T_{(hi,4)}$	19	39	3	20
$T_{(hi,5)}$	25	22	7	29
$T_{(hi,6)}$	21	16	2	31
$T_{(hi,7)}$	18	37	11	9
$T_{(hi,8)}$	20	29	8	27
$T_{(hi,9)}$	12	28	8	35
$T_{(hi,10)}$	22	19	3	28
$R_{(j)}$	20.9	29.2	5.6	23.1

The recommended tourist attraction classification is determined by the output recommendation degree function  $R_{(j)}$  based on the selected tourist attraction classification feature labels  $L_{(i,j)}$  of multiple historical tourists. The experiment chooses the topological vector of each tourist attraction classification word frequency  $L_{(1)to}$ : history, science, venue, memorial, museum;  $L_{(2)to}$ : sightseeing, flower, park, reenland, scenery;  $L_{(3)to}$ : swim, sports, playground, theme park, children animation;  $L_{(4)to}$ : shop, commercial complex, leisure, restaurant. The tourist attraction feature attribute label matrix  $L_M$  is set up. Via the matrix  $L_M$ , the tourist attraction feature attribute label word frequency matrix  $F_q(L_M)$  of the nearest neighborhood historical tourists is formed. Then, the recommendation degree  $R_{(j)}$  is output, as shown in Table 4. Figure 9a–d show an example of tourist attraction classification label word frequency and its related recommendation degree  $R_{(j)}$  in the nearest neighborhood historical tourists' evaluation data. In each figure, the last red point of the curve represents the degree  $R_{(j)}$ . Figure 9e presents a comparison of the different tourist attraction classification word frequencies and the degrees  $R_{(j)}$  of the historical tourists. Figure 9f shows a comparison of each tourist attraction classification recommendation degree  $R_{(j)}$ . The priority vector  $\mathbf{R} = \{T_{(2)}, T_{(4)}, T_{(1)}, T_{(3)}\}$  is obtained. Regarding the sample tourist, when he obtains the vector  $\mathbf{R}$ , he makes a decision that, in one day, he will visit two tourist attractions of the cluster  $T_{(2)}$ , one tourist attraction of the cluster  $T_{(4)}$ , and one tourist attraction of the cluster  $T_{(1)}$ . The coordinates of the starting point  $St$  are (113.658, 34.774), shown as the white circle with the letter S in Figure 7b. The vector  $\mathbf{R}_{to} = \{O_{(2)}, O_{(7)}, O_{(16)}, O_{(20)}\}$  is output.



**Figure 9.** The fluctuating curves of the tourist attraction classification label word frequency and a comparison of the tourist attraction classification recommendation degree. (a–d) represent the six column results in Table 5. (e) shows the four curves together. (f) shows the comparison of  $R_{(j)}$ .

**Table 5.** The constraint factor  $\zeta_{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , correction parameter  $\varepsilon_{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , and weight value  $g(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  of each tourist attraction adjacency section  $Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  in different transportation modes  $\tau(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ .

$\tau_{(i)}$	$\tau = 1$			$\tau = 2$			$\tau = 3$			$\tau = 1$			$\tau = 2$			$\tau = 3$		
$\zeta_{(i)}$	$\zeta_{(1)}$	$\zeta_{(2)}$	$\zeta_{(3)}$	$\zeta_{(1)}$	$\zeta_{(2)}$	$\zeta_{(3)}$	$\zeta_{(1)}$	$\zeta_{(2)}$	$\zeta_{(3)}$	$g$								
$\varepsilon_{(i)}$	0.10	1.00	0.10	0.10	1.00	0.10	0.10	1.00	0.10									
$S_t, O_{(2)}$	5.50	0	1.50	8.10	0.25	15.10	9.20	0.25	1.00	0.700	2.570	1.270						
$S_t, O_{(7)}$	1.10	0	1.50	2.30	0.34	6.45	1.20	0.34	1.00	0.260	1.215	0.560						
$S_t, O_{(16)}$	3.10	0	1.50	5.40	0.39	11.10	5.90	0.39	1.00	0.460	2.040	1.080						
$S_t, O_{(20)}$	2.10	0	2.00	2.30	0.35	6.45	2.20	0.35	1.00	0.410	1.225	0.670						
$O_{(2)}, O_{(7)}$	7.00	0	2.50	8.60	0.30	15.90	9.50	0.30	1.00	0.950	2.750	1.350						
$O_{(2)}, O_{(16)}$	3.80	0	1.50	4.10	0.23	9.10	4.00	0.23	1.00	0.530	1.550	0.730						
$O_{(2)}, O_{(20)}$	7.30	0	3.00	8.40	0.40	15.60	8.70	0.40	1.00	1.030	2.800	1.370						
$O_{(7)}, O_{(16)}$	3.30	0	2.00	4.20	0.36	9.30	3.80	0.36	1.00	0.530	1.710	0.840						
$O_{(7)}, O_{(20)}$	0.92	0	1.50	1.30	0.13	6.00	1.00	0.13	1.00	0.242	0.860	0.330						
$O_{(16)}, O_{(20)}$	3.60	0	2.00	4.60	0.37	9.90	4.10	0.37	1.00	0.560	1.820	0.880						

### 4.3.2. The Analysis of the Data Results

The generated nearest neighborhood historical tourists have feature attribute factors that have a small discrepancy. As shown in Figure 8, the traveling costs of the neighborhood tourists have a very tiny discrepancy, and it is displayed as a fluctuating curve with waved weighted values between 0.9 and 1.1. This illustrates that the tendency value for the tour process in the tourist attributes ranges between CNY 90 yuan and 110 yuan. (CNY: Chinese monetary unit) Regarding the attribute of traveling time, the weighted values of the traveling time requirement for the neighborhood historical tourists vary from 0.12 to 0.18. This illustrates that the preferred traveling time of the neighborhood historical tourists is 1.2–1.8 h within 2 h. The preferred tourist attraction hot indexes of the neighborhood historical tourist vary from 0.36 to 0.65, centering at 0.50. This illustrates that the historical tourists’ preference for the popular tourist attractions is average, and the tourist attraction hot index is not the determined factor in choosing a tourist attractions. Regarding the selection of a tourist attraction destination, the values 0.50 and 0.75 appear frequently, i.e., the main purpose of traveling tends to be health care or vacation. Regarding the selection of the transportation mode, the current tourist and historical tourists all choose cycling, and the weighted value is 0.50. The searching objective function values calculated by the feature attribute factor range from 0.054 to 0.287, and the values are all relatively small. This illustrates that the attributes of the current tourist and historical tourists are similar. These historical tourists’ interests can reflect the current tourist’s interests.

The tourist attraction classification label word frequency of each neighborhood historical tourist fluctuates with different values, but they are generally close. The word

label frequency for the first, the second and the fourth classification is relatively high; the average value is 20.9, 29.2, and 23.1, respectively. This illustrates that the neighborhood tourists' preferences regarding these classifications are high. Thus, the preferred tourist attraction classifications of the neighborhood historical tourists are the park and greenland, venue and memorial, and leisure shopping, which will be recommended.

#### 4.4. The Recommendation Result and Analysis on the Optimal Tour Route Based on the Precise Tourist Attraction Approach Vector Algorithm

The tourist attraction geospatial directed weighted graph  $D$  is output, as shown in Figure 10. The edges' weight values are calculated and the global route weight iteration value  $G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  of each tour route and the optimal tour route are output.

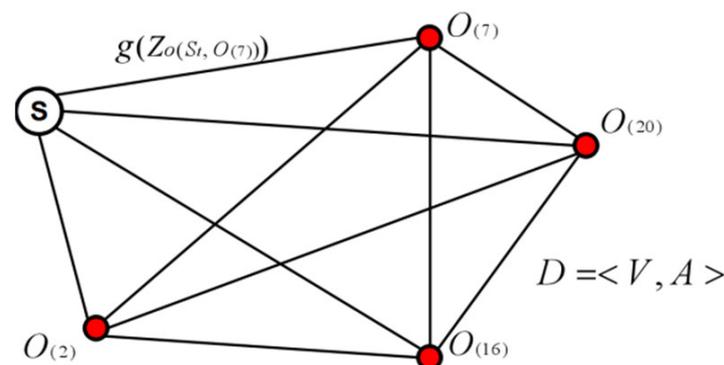


Figure 10. The tourist attraction geographic space directed weighted graph  $D$ .

##### 4.4.1. The Output Result of the Optimal Tour Route Recommendation Based on the Precise Tourist Attraction Approach Vector Algorithm

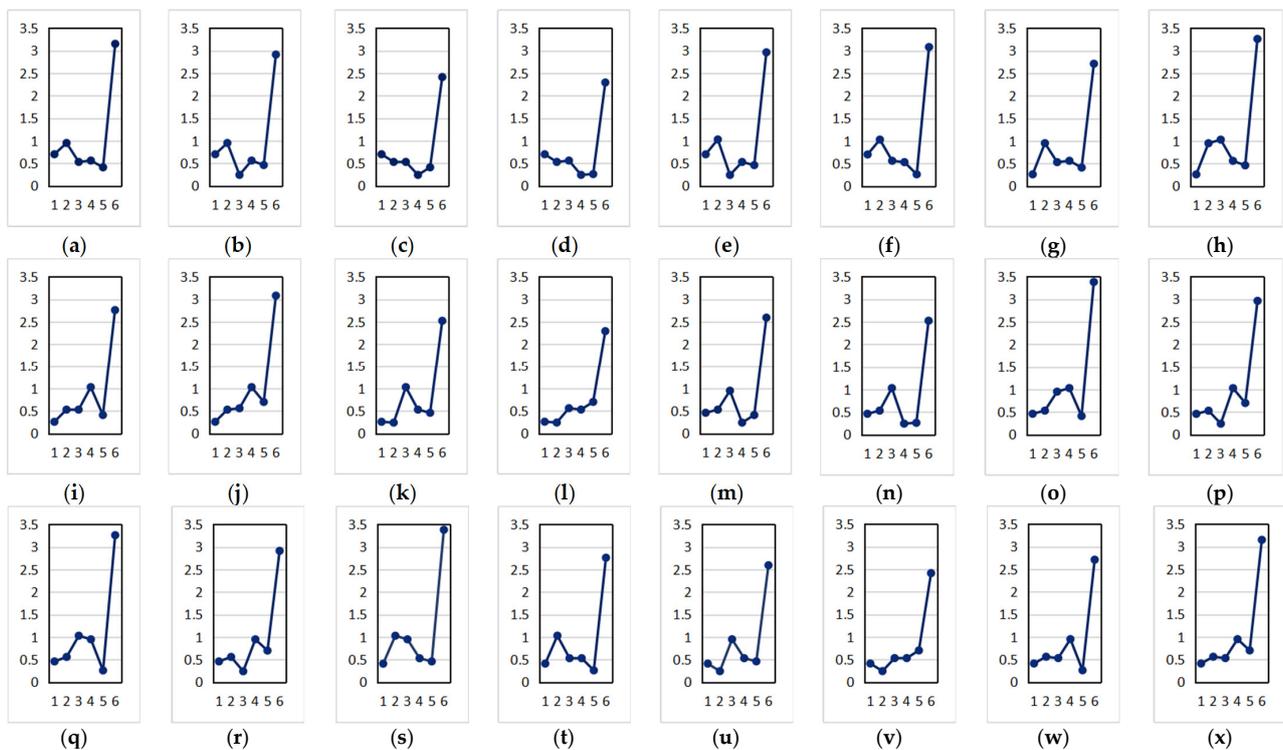
The constraint factors  $\zeta^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  and parameters  $\varepsilon^{(i)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  of each section  $Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  in different transportation modes  $\zeta^{(3)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , and the weight value  $g(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  for each section  $Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , are calculated, as shown in Table 5. The  $\tau_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})} = 1$  represents cycling, the  $\tau_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})} = 2$  represents the use of a taxi and the  $\tau_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})} = 3$  represents the use of the public bus. The weight value  $g(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  for each section  $Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  of each tour route in different transportation modes, as well as the global weight iteration value  $G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  are output. The five values represent the five sections' weight for one tour route, shown in Table 6. Each tour route's starting point and terminal point are both the  $S_t$ . The representation method for each tour route is  $O_{(1,2,3,4)}$ , representing the tour route  $S_t O_{(1)} O_{(2)} O_{(3)} O_{(4)} S_t$ . Figures 11–13 shows the fluctuating tendency of the weight value  $g(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  for each section  $Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ . According to the searching process of the global optimal solution  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  and the difference value  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$ , the formed maximum heap  $T$  is output, as shown in Figure 14. According to the result of Table 6 and Figure 14, the global optimal solution  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  and the difference value  $\min G(Z_o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  and the optimal tour route are output.

##### 4.4.2. The Data Result Analysis

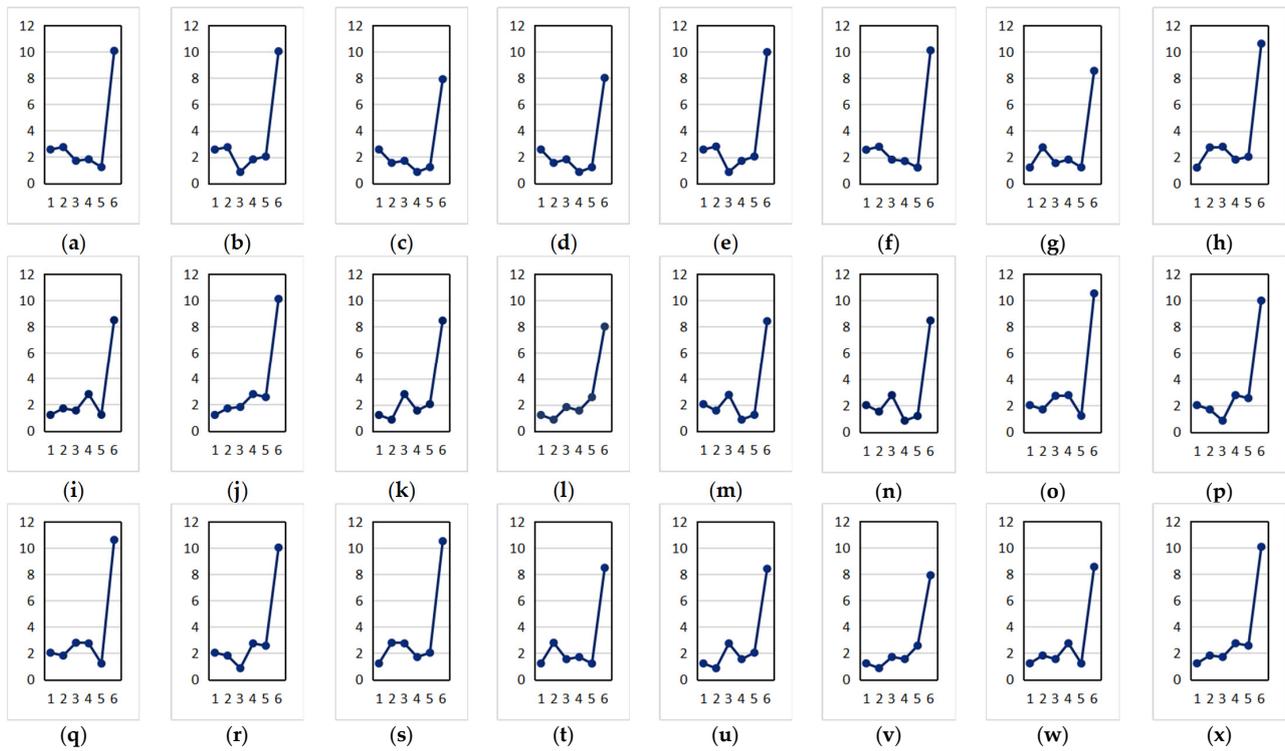
In Figure 10, the edge weight reflects the generated motive benefit value when the tourists travel between two points. The smaller the edge weight is, the lower the traveling cost and the higher the tourists' motive benefit values will be. The positive and the negative direction of the edge represents the travel direction. Since the geographic information and traffic information are not influenced by the direction, the weight values will not be influenced by the direction.

**Table 6.** Tour route section weight and global weight iteration value in different transportation modes.

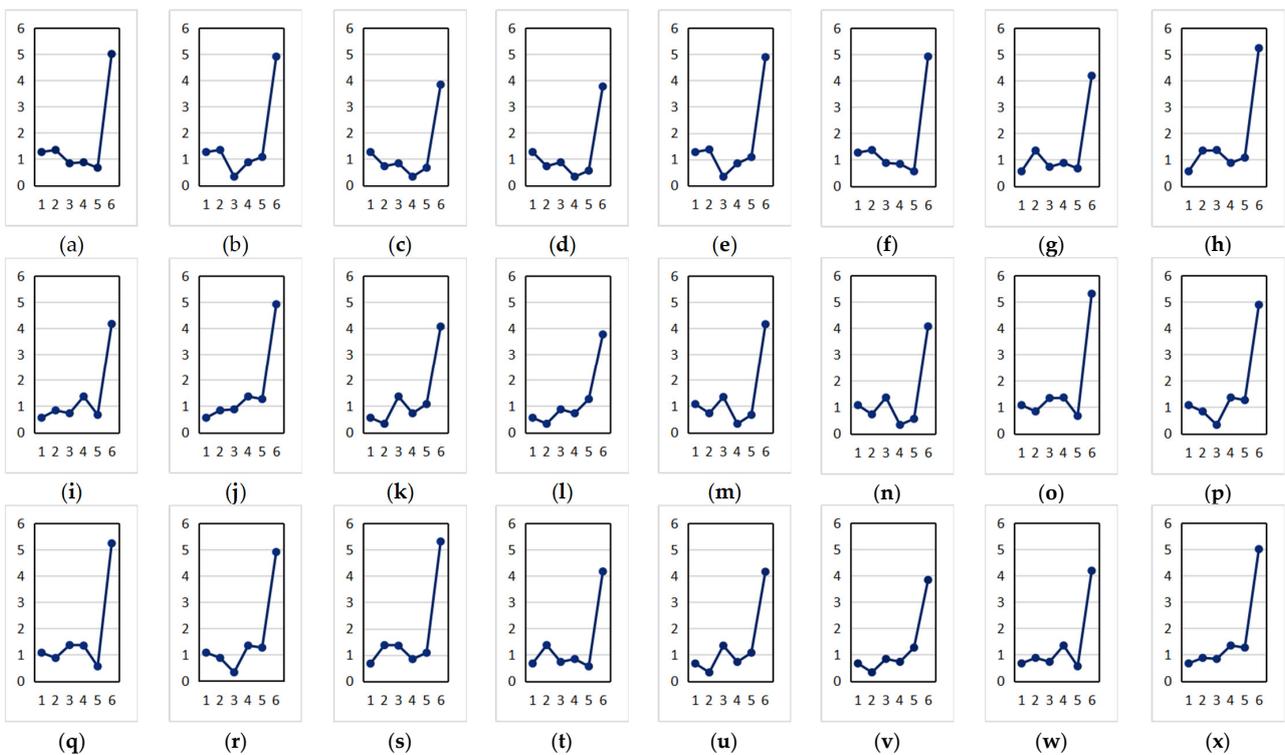
	$\tau(i)$	$\tau = 1$			$\tau = 2$			$\tau = 3$		
		$\zeta(i)$	$g$	$G$	$g$	$G$	$g$	$G$		
1	$O_{(2,7,16,20)}$		0.7, 0.95, 0.53, 0.56, 0.41	3.15	2.57, 2.75, 1.71, 1.82, 1.225	10.075	1.27, 1.35, 0.84, 0.88, 0.67	5.01		
2	$O_{(2,7,20,16)}$		0.7, 0.95, 0.242, 0.56, 0.46	2.912	2.57, 2.75, 0.86, 1.82, 2.04	10.040	1.27, 1.35, 0.33, 0.88, 1.08	4.91		
3	$O_{(2,16,7,20)}$		0.7, 0.53, 0.53, 0.242, 0.41	2.412	2.57, 1.55, 1.71, 0.86, 1.225	7.915	1.27, 0.73, 0.84, 0.33, 0.67	3.84		
4	$O_{(2,16,20,7)}$		0.7, 0.53, 0.56, 0.242, 0.26	2.292	2.57, 1.55, 1.82, 0.86, 1.215	8.015	1.27, 0.73, 0.88, 0.33, 0.56	3.77		
5	$O_{(2,20,7,16)}$		0.7, 1.03, 0.242, 0.53, 0.46	2.962	2.57, 2.8, 0.86, 1.71, 2.04	9.980	1.27, 1.37, 0.33, 0.84, 1.08	4.89		
6	$O_{(2,20,16,7)}$		0.7, 1.03, 0.56, 0.53, 0.26	3.080	2.57, 2.8, 1.82, 1.71, 1.215	10.115	1.27, 1.37, 0.88, 0.84, 0.56	4.92		
7	$O_{(7,2,16,20)}$		0.26, 0.95, 0.53, 0.56, 0.41	2.710	1.215, 2.75, 1.55, 1.82, 1.225	8.5600	0.56, 1.35, 0.73, 0.88, 0.67	4.19		
8	$O_{(7,2,20,16)}$		0.26, 0.95, 1.03, 0.56, 0.46	3.260	1.215, 2.75, 2.8, 1.82, 2.04	10.625	0.56, 1.35, 1.37, 0.88, 1.08	5.24		
9	$O_{(7,16,2,20)}$		0.26, 0.53, 0.53, 1.03, 0.41	2.760	1.215, 1.71, 1.55, 2.8, 1.225	8.500	0.56, 0.84, 0.73, 1.37, 0.67	4.17		
10	$O_{(7,16,20,2)}$		0.26, 0.53, 0.56, 1.03, 0.7	3.080	1.215, 1.71, 1.82, 2.8, 2.57	10.115	0.56, 0.84, 0.88, 1.37, 1.27	4.92		
11	$O_{(7,20,2,16)}$		0.26, 0.242, 1.03, 0.53, 0.46	2.522	1.215, 0.86, 2.8, 1.55, 2.04	8.465	0.56, 0.33, 1.37, 0.73, 1.08	4.07		
12	$O_{(7,20,16,2)}$		0.26, 0.242, 0.56, 0.53, 0.7	2.292	1.215, 0.86, 1.82, 1.55, 2.57	8.015	0.56, 0.33, 0.88, 0.73, 1.27	3.77		
13	$O_{(16,2,7,20)}$		0.46, 0.53, 0.95, 0.242, 0.41	2.592	2.04, 1.55, 2.75, 0.86, 1.225	8.425	1.08, 0.73, 1.35, 0.33, 0.67	4.16		
14	$O_{(16,2,20,7)}$		0.46, 0.53, 1.03, 0.242, 0.26	2.522	2.04, 1.55, 2.8, 0.86, 1.215	8.465	1.08, 0.73, 1.37, 0.33, 0.56	4.07		
15	$O_{(16,7,2,20)}$		0.46, 0.53, 0.95, 1.03, 0.41	3.380	2.04, 1.71, 2.75, 2.8, 1.225	10.525	1.08, 0.84, 1.35, 1.37, 0.67	5.31		
16	$O_{(16,7,20,2)}$		0.46, 0.53, 0.242, 1.03, 0.7	2.962	2.04, 1.71, 0.86, 2.8, 2.57	9.980	1.08, 0.84, 0.33, 1.37, 1.27	4.89		
17	$O_{(16,20,2,7)}$		0.46, 0.56, 1.03, 0.95, 0.26	3.260	2.04, 1.82, 2.8, 2.75, 1.215	10.625	1.08, 0.88, 1.37, 1.35, 0.56	5.24		
18	$O_{(16,20,7,2)}$		0.46, 0.56, 0.242, 0.95, 0.7	2.912	2.04, 1.82, 0.86, 2.75, 2.57	10.040	1.08, 0.88, 0.33, 1.35, 1.27	4.91		
19	$O_{(20,2,7,16)}$		0.41, 1.03, 0.95, 0.53, 0.46	3.380	1.225, 2.8, 2.75, 1.71, 2.04	10.525	0.67, 1.37, 1.35, 0.84, 1.08	5.31		
20	$O_{(20,2,16,7)}$		0.41, 1.03, 0.53, 0.53, 0.26	2.76	1.225, 2.8, 1.55, 1.71, 1.215	8.5	0.67, 1.37, 0.73, 0.84, 0.56	4.17		
21	$O_{(20,7,2,16)}$		0.41, 0.242, 0.95, 0.53, 0.46	2.592	1.225, 0.86, 2.75, 1.55, 2.04	8.425	0.67, 0.33, 1.35, 0.73, 1.08	4.16		
22	$O_{(20,7,16,2)}$		0.41, 0.242, 0.53, 0.53, 0.7	2.412	1.225, 0.86, 1.71, 1.55, 2.57	7.915	0.67, 0.33, 0.84, 0.73, 1.27	3.84		
23	$O_{(20,16,2,7)}$		0.41, 0.56, 0.53, 0.95, 0.26	2.71	1.225, 1.82, 1.55, 2.75, 1.215	8.56	0.67, 0.88, 0.73, 1.35, 0.56	4.19		
24	$O_{(20,16,7,2)}$		0.41, 0.56, 0.53, 0.95, 0.7	3.15	1.225, 1.82, 1.71, 2.75, 2.57	10.075	0.67, 0.88, 0.84, 1.35, 1.27	5.01		



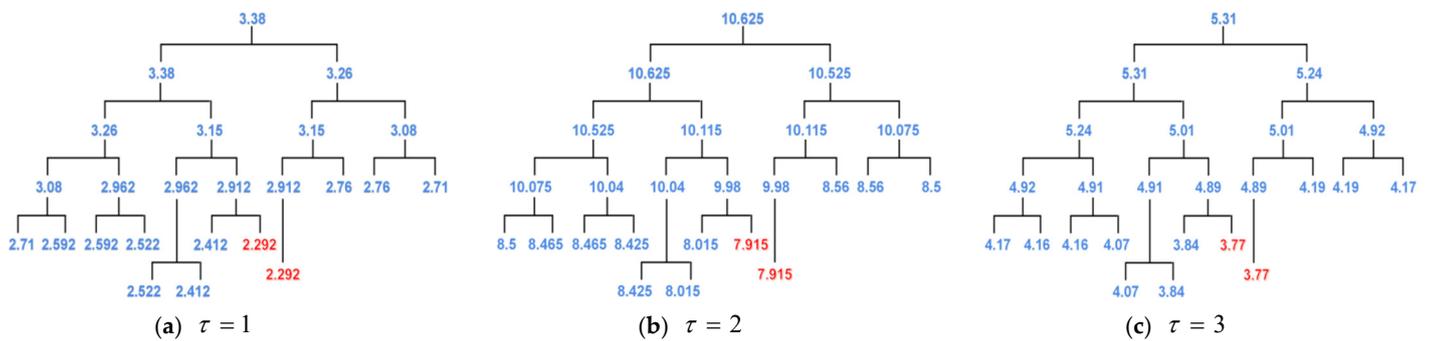
**Figure 11.** The increasing tendency of the weight  $g(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  of each tour route tourist attraction adjacency section  $Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  under the condition of transportation mode of cycling. The figure (a–x) represent the No.1~No.24 tour routes shown in the Table 6 under the condition of transportation mode of cycling  $\tau = 1$ .



**Figure 12.** The increasing tendency of the weight  $g(Z_0(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  of each tour route tourist attraction adjacency section  $Z_0(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  under the condition of transportation mode of using a taxi. The figure (a–x) represent the No.1~No.24 tour routes shown in the Table 6 under the condition of transportation mode of using a taxi  $\tau = 2$ .



**Figure 13.** The increasing tendency of the weight  $g(Z_0(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)}))$  of each tour route tourist attraction adjacency section  $Z_0(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  under the condition of transportation mode of using the public bus. The figure (a–x) represent the No.1~No.24 tour routes shown in the Table 6 under the condition of transportation mode of using the public bus  $\tau = 3$ .



**Figure 14.** The maximum complete binary tree based on the global weight iteration value  $G(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ . (a–c) show the maximum heap result in different transportation modes.

According to the calculation result, when the tourists choose cycling, the optimal tour routes are  $S_t O_{(2)} O_{(16)} O_{(20)} O_{(7)} S_t$  and  $S_t O_{(7)} O_{(20)} O_{(16)} O_{(2)} S_t$ , and the minimum global weight iteration value is 2.292. When the tourists choose to use a taxi, the optimal tour routes are  $S_t O_{(2)} O_{(16)} O_{(7)} O_{(20)} S_t$  and  $S_t O_{(20)} O_{(7)} O_{(16)} O_{(2)} S_t$ , and the minimum global weight iteration value is 7.915. When the tourists choose to use the public bus, the optimal tour routes are  $S_t O_{(2)} O_{(16)} O_{(20)} O_{(7)} S_t$  and  $S_t O_{(7)} O_{(20)} O_{(16)} O_{(2)} S_t$ , and the minimum global weight iteration value is 3.770. Figures 11–13 show that, for different transportation modes, each tour route’s output neighborhood section weight fluctuates throughout the entire traveling process and reaches the maximum value at the terminal point. The fluctuating waves for each tour route have different shapes. When cycling, the weight value ranges from 0 to 3.5. When taking a taxi, the weight value ranges from 0 to 12.0. When using the public bus, the weight value ranges from 0 to 6.0.

In the maximum heap, the global weight iteration values appear at the last two elements in the last level. In Figure 14, the elements storing the global weight iteration values are shown in red and relate to the optimal tour routes under the related transportation modes.

#### 4.5. The Comparison of the Algorithms

Three electronic maps are chosen as the control group. The electronic maps include the Baidu map, Gaode map and Tengxun map. The embedded algorithms are defined as the Baidu Algorithm (BA), Gaode Algorithm (GA), and Sougou Algorithm (SA), and the proposed algorithm is referred to as PA.

##### 4.5.1. The Comparison Result of the Algorithm Data

The searching methods of the algorithms are different. The proposed algorithm searches for the shortest path by using the approach function points, while the traditional maps frequently search the main roads. Therefore, when generating the directed weighted graph, the weight  $g(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  of the section  $Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  differs from each other since the shortest distance factor  $\zeta_{(1)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , road congestion index  $\zeta_{(2)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$ , and the traveling fee  $\zeta_{(3)}(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})$  are different; thus, the same section  $Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  will generate different weight values. This will cause a difference in the optimal tour route.

Under the same experimental conditions, the final optimal tour route, each route’s weight  $g(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  of each section  $Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$ , the global optimal solution  $\min G(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  and the difference value  $\min \Delta G(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$ , and the difference value of the global optimal solution  $\min \Delta G(Z_{o(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})})$  in different transportation modes  $\tau_{(\mathbf{R}_{to(i)}, \mathbf{R}_{to(j)})}$  are output, as shown in Table 7. The bold value of the weight  $g$  indicates the global optimal solution  $\min G$ . Figure 15a–c show the weight tendency of the optimal tour route in the different algorithms under the modes of  $\tau = 1$ ,  $\tau = 2$ , and  $\tau = 3$ . Different colors relate to the tourist attractions

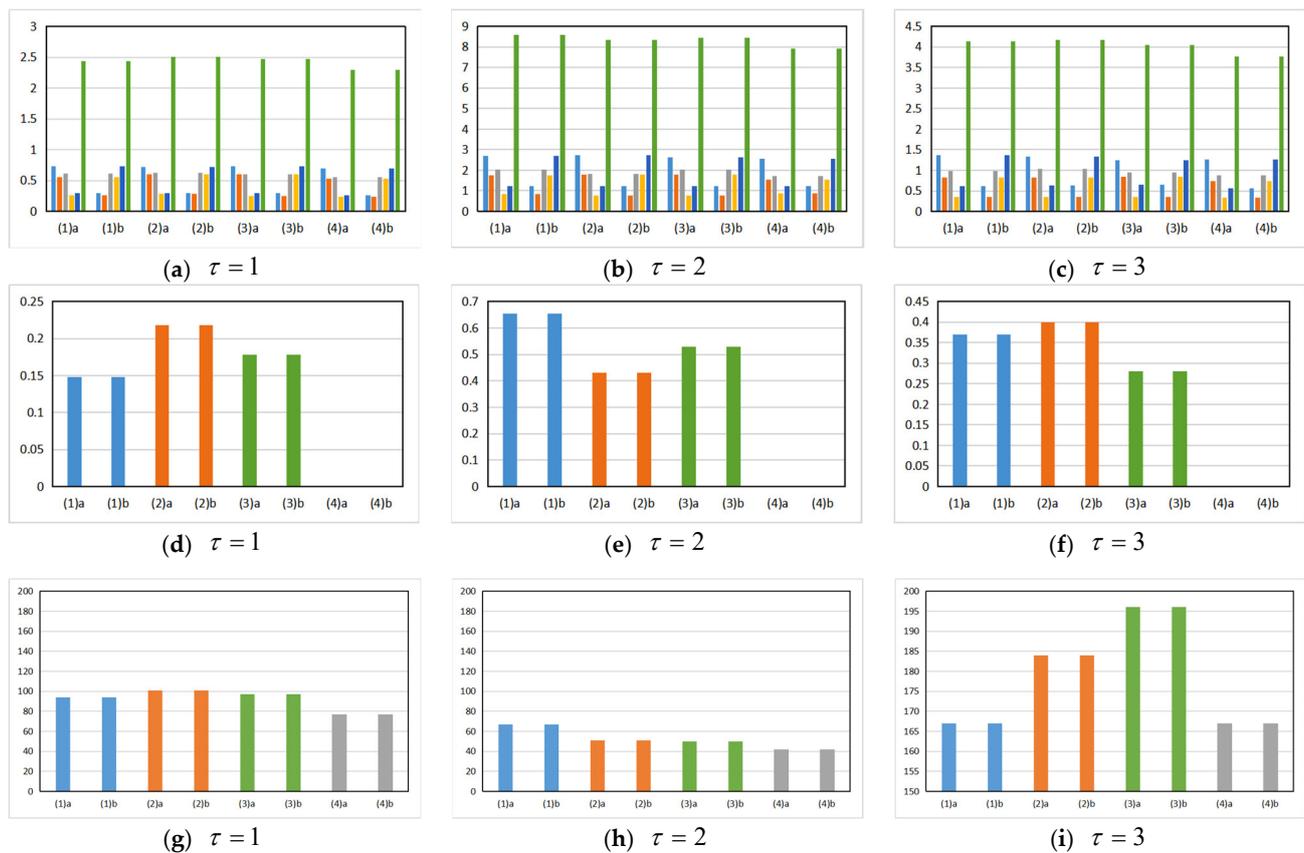
on the route. Figure 15d–f show the global optimal solution differences  $\min\Delta G(Z_{o(\mathbf{R}_{io(i)}, \mathbf{R}_{io(j)})})$  between the BA, GA, SA, and the PA under the modes of  $\tau = 1$ ,  $\tau = 2$ , and  $\tau = 3$ , represented by the blue, orange, and green color, respectively. The minimum time expense  $t(\text{min})$  for the travel process in the entire route is different for each algorithm. Table 8 shows a comparison of the traveling time of the optimal tour route for each algorithm. Figure 15g–i show the traveling time of the optimal tour route for each algorithm under different modes. The blue, orange, green, and gray colors represent BA, GA, SA, and PA, respectively.

**Table 7.** The comparison of the global optimal solution  $\min G(Z_{o(\mathbf{R}_{io(i)}, \mathbf{R}_{io(j)})})$  and the difference value  $\min\Delta G(Z_{o(\mathbf{R}_{io(i)}, \mathbf{R}_{io(j)})})$ . The bold value of the weight  $g$  indicates the global optimal solution  $\min G$ .

$\tau$	(1) BA	(2) GA	(3) SA	(4) PA	$g \sim \min G$					$\min\Delta G$			
					(1) a	(1) b	(2) a	(2) b	(3) a		(3) b	(4) a	(4) b
1	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	(1) a	0.73	0.55	0.61	0.26	0.29	<b>2.44</b>	0.148
						(1) b	0.29	0.26	0.61	0.55	0.73	<b>2.44</b>	0.148
						(2) a	0.72	0.6	0.62	0.28	0.29	<b>2.51</b>	0.218
						(2) b	0.29	0.28	0.62	0.60	0.72	<b>2.51</b>	0.218
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	(3) a	0.73	0.60	0.60	0.25	0.29	<b>2.47</b>	0.178
						(3) b	0.29	0.25	0.60	0.60	0.73	<b>2.47</b>	0.178
						(4) a	0.70	0.53	0.56	0.242	0.26	<b>2.292</b>	–
						(4) b	0.26	0.242	0.56	0.53	0.70	<b>2.292</b>	–
2	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	(1) a	2.71	1.75	2.05	0.825	1.235	<b>8.57</b>	0.655
						(1) b	1.235	0.825	2.05	1.75	2.71	<b>8.57</b>	0.655
						(2) a	2.75	1.785	1.81	0.765	1.235	<b>8.345</b>	0.430
						(2) b	1.235	0.765	1.81	1.785	2.75	<b>8.345</b>	0.430
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(20,7,16,2)}$	(3) a	2.645	1.775	2.025	0.775	1.225	<b>8.445</b>	0.530
						(3) b	1.225	0.775	2.025	1.775	2.645	<b>8.445</b>	0.530
						(4) a	2.57	1.55	1.71	0.86	1.225	<b>7.915</b>	–
						(4) b	1.225	0.86	1.71	1.55	2.57	<b>7.915</b>	–
3	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	(1) a	1.36	0.83	0.98	0.35	0.62	<b>4.14</b>	0.370
						(1) b	0.62	0.35	0.98	0.83	1.36	<b>4.14</b>	0.370
						(2) a	1.33	0.82	1.03	0.35	0.64	<b>4.17</b>	0.400
						(2) b	0.64	0.35	1.03	0.82	1.33	<b>4.17</b>	0.400
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	(3) a	1.25	0.85	0.95	0.35	0.65	<b>4.05</b>	0.280
						(3) b	0.65	0.35	0.95	0.85	1.25	<b>4.05</b>	0.280
						(4) a	1.27	0.73	0.88	0.33	0.56	<b>3.77</b>	–
						(4) b	0.56	0.33	0.88	0.73	1.27	<b>3.77</b>	–

**Table 8.** A comparison of the traveling time expense of the optimal tour route for each algorithm under the different transportation modes.

$\tau$		(1) BA	(2) GA	(3) SA	(4) PA	$t(\text{min})$			
						(1) BA	(2) GA	(3) SA	(4) PA
1	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	94	101	97	77
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	94	101	97	77
2	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,7,20)}$	67	51	50	42
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(20,7,16,2)}$	67	51	50	42
3	a	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	$O_{(2,16,20,7)}$	167	184	196	167
	b	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	$O_{(7,20,16,2)}$	167	184	196	167



**Figure 15.** The weight fluctuation for each algorithm and a comparison of the global optimal solution difference values under the different transportation modes. (a–c) show the weight tendency of the optimal tour route in different algorithms under the modes. (d–f) show the global optimal solution differences in different algorithms under the modes. (g–i) show the ferry traveling time of the optimal tour route for each algorithm under different modes.

#### 4.5.2. The Data Result Analysis

For the three transportation modes, the neighborhood section weights of the optimal route in each algorithm fluctuate to different degrees. When the tourists choose cycling, the BA experiences the largest fluctuation, followed by the GA and SA; the PA has the smallest fluctuation. The weights vary from 0 to 3.5. When the tourists choose to use a taxi, the fluctuation is roughly the same, and the weights vary from 0 to 12.0. When the tourists choose to use the public bus, the BA and GA have relatively large fluctuations, followed by the SA; the PA has the smallest fluctuation. The weights vary from 0 to 6.0. In this aspect, the PA has better stability than the control group algorithms.

When the tourists choose cycling, the global optimal solutions of BA, GA, and SA are larger than the PA, in which the BA, GA, and SA are larger than the PA for 0.148, 0.218, and 0.178, respectively. When the tourists choose to use a taxi, the global optimal solutions of BA, GA, and SA are larger than the PA, in which the BA, GA, and SA are larger than the PA for 0.655, 0.430, and 0.530, respectively. When the tourists choose to use the public bus, the global optimal solutions of BA, GA, and SA are larger than the PA, in which the BA, GA, and SA are larger than the PA for 0.370, 0.400, and 0.280, respectively. Regarding the comparison data, the control group algorithms' global optimal solution values are all larger than that of the proposed algorithm, which illustrates that the output global optimal solution value of the proposed algorithm is the smallest, and the tourists will spend the least traveling time and money to achieve the best motive benefit satisfaction.

When the tourists choose cycling, the GA requires the most traveling time, followed by the SA and BA; the lowest time is needed for the PA. When the tourists choose to use a taxi, the BA requires the most traveling time, followed by the GA and SA; the least time

is required for the PA. When the tourists choose to use the public bus, the SA requires the most traveling time, followed by the GA; the least time is needed for the PA and BA. The PA is the least time-consuming, except when the tourist chooses to use the public bus, for which the PA is equally as time-consuming as the BA. On the whole, the proposed algorithm is the optimal one.

#### 4.6. The Proposed Algorithm and Other Recommendation Algorithms' Difference Analysis

The proposed recommendation algorithm (PRA) and other recommendation algorithms (ORA) display great discrepancies.

The first discrepancy is the different algorithm essence. The PRA's essence is to search the tourist attractions that best match tourists' interests, and then to search the optimal tour route, which is different from ORA. The ORA's essence is to make a fuzzy recommendation based on historical data—for instance, recommendations based on users, based on objects, based on content, and based on association rules. Recommendations based on users recommend related users' favorite objects. Recommendations based on objects recommend the user's previous favorite objects. Recommendations based on content are based on users' evaluation data. Recommendations based on association rules use probability to recommend objects. Different in essence, the PRA directly obtains tourists' precise interest data and searches tourist attractions one by one; it then searches the tour route along the road;

The second discrepancy is the different objective. In order to increase the accuracy, the ORA needs to solve the problems associated with cold start, data sparsity, etc. However, the PRA does not aim to solve these problems. Its aim is to obtain the best match between tourists' interests and the tourist attraction attributes, as well as to search the optimal tour route;

The third discrepancy is the different accuracy degree. The PRA is based on the interest data directly generated by the tourists; thus, the recommended tourist attractions are guaranteed to satisfy their interests. This is the advantage of the PRA. The ORA is still a fuzzy recommendation in essence, which can only increase the accuracy, but it is unable to thoroughly match tourists' precise interests;

The fourth discrepancy is the different principles and algorithm process. Due to the differences in essence and objectives, the realization process of the PRA is different from that of the ORA. Its main steps include the searching of precise tourist attractions and the searching of the optimal tour route, which is more accurate than the ORA, since the ORA usually directly recommends a tour route previously visited by historical tourists. The mode has relatively low accuracy and cannot meet the individualized needs.

#### 4.7. The Discussion on the Method Execution and Other Use Cases

The aim of this study is to propose an intelligent algorithm for a smart tourism recommendation system; thus, the proposed algorithm can automatically realize tourist attraction and tour route recommendations. Regarding the data required to execute the algorithm, the tourist attraction data, attribute data, city geospatial information data, traffic information data, and the website evaluation data are fixed, and they form the algorithm's underlying database. Meanwhile, the tourists' interest data and the starting point of the tour are two variables, and they are key factors to determine the recommended results. When an arbitrary variable changes, the recommended results will alter accordingly because, under the control of the inner algorithm, the initial variable changes will cause different intermediate results in the searching process. Based on the experiment's basic data, other use cases are discussed as follows.

The first use case is the condition that the tourist's interest data change, but the starting location does not change. A tourist tends to choose labels from the word frequency vectors  $L_{(3)to}$ : swim, sports, playground, theme park, children animation and  $L_{(4)to}$ : shop, commercial complex, leisure, restaurant. He chooses two tourist attractions from each classification, while the other two vectors  $L_{(1)to}$  and  $L_{(2)to}$  are labeled 0. Additionally, the

starting location (113.658, 34.774) does not change. The final recommended tourist attractions and tour routes for the different transportation modes are  $S_t O_{(12)} O_{(20)} O_{(17)} O_{(18)} S_t$  and  $S_t O_{(18)} O_{(17)} O_{(20)} O_{(12)} S_t$ ;

The second use case is the condition that the tourist's interest data do not change, but the starting location changes. Under this condition, the recommendation system outputs  $\mathbf{R} = \{\mathbf{T}_{(2)}, \mathbf{T}_{(4)}, \mathbf{T}_{(1)}, \mathbf{T}_{(3)}\}$ , while the tourist's interest data do not change; two tourist attractions of the cluster  $\mathbf{T}_{(2)}$ , one tourist attraction of the cluster  $\mathbf{T}_{(4)}$  and one tourist attraction of the cluster  $\mathbf{T}_{(1)}$ . If the starting point changes to location (113.644, 34.699), the recommended tourist attractions and tour routes for the different transportation modes are  $S_t O_{(13)} O_{(15)} O_{(2)} O_{(16)} S_t$  and  $S_t O_{(16)} O_{(2)} O_{(15)} O_{(13)} S_t$ ;

The third use case is the condition that both the tourists' interest data and starting point change. A tourist tends to choose labels from the word frequency vectors  $\mathbf{L}_{(1)t_0}$ : history, science, venue, memorial, museum;  $\mathbf{L}_{(2)t_0}$ : sightseeing, flower, park, greenland, scenery. He chooses two tourist attractions from each classification, while the other two vectors  $\mathbf{L}_{(3)t_0}$  and  $\mathbf{L}_{(4)t_0}$  are labeled 0. The starting point changes to location (113.682, 34.749). The recommended tourist attractions and tour routes are  $S_t O_{(1)} O_{(16)} O_{(4)} O_{(10)} S_t$  and  $S_t O_{(10)} O_{(4)} O_{(16)} O_{(1)} S_t$  for the different transportation modes.

From the use cases, it can be concluded that when one variable changes, the finally output tourist attraction and tour route will change accordingly. This confirms that the proposed algorithm is feasible and practical for a smart recommendation system.

#### 4.8. Conclusions on the Problems Solved by the Proposed Algorithm

As to Problem (1), previous research focused on the algorithm itself, such as the algorithm efficiency, but the aim of the proposed method is different. It focuses on searching the very tourist attractions and optimal tour route that best match tourists' interests, and this is the key aspect when setting up the algorithm process; in addition, its essence is different from that of other recommendation methods. In Section 4.6, a comparison between the proposed method and other recommendation methods is presented based on the entire research. As to Problem (2) and Problem (3), Algorithm 1 is set up in Section 2.1.1, the Algorithm 2 in Section 2.1.2, Algorithm 3 in Section 2.2, and the searching algorithm in Section 3 to solve the problems of precisely matching the tourist attractions to the tourists' interests, searching the tourist attractions and searching the tour routes. The experiment chooses the city of Zhengzhou as an example and extracts the basic data of its tourist attractions, traffic information data, and geospatial data. In Section 4.3, the recommendation degree is calculated from the chosen interest labels and the tourist attraction classification priority sequence is recommended. According to the tour starting point, the four optimal tourist attractions that best match tourists' interests are obtained. This experiment proves that the tourist attractions recommended by the proposed algorithm completely match tourists' interests with optimal feature attributes, matching degree, and locations. In Sections 4.4 and 4.5, the optimal tour routes for different transportation modes are output, which proves that they have advantages regarding the motive benefit, time consumption, etc. Moreover, Problem (2) and Problem (3) are well solved. Regarding the Problem (4), the recommended tour routes are based on the searching algorithm developed in Section 3, and through Sections 4.4 and 4.5, this method is proven to have high accuracy, as it is directly based on tourists' individualized interest data, contrasting the traditional recommendation mode with the fuzzy method in which historical tour routes are recommended to the tourists. Problem (4) is also well solved.

## 5. Conclusions

The current tourist attraction and tour route recommendation system experiences problems. Aimed at addressing these problems, this paper proposes an algorithm. The tourist attraction cellular generating space and the tourist attraction clustering model are set up. The aim is to recommend the optimal tourist attractions for the tourists. This paper also proposes a tourist attraction recommendation algorithm based on the weighted

collaborative filtering algorithm. It also involves an optimal tour route recommendation model based on the precise tourist attractions using different transportation modes. This model uses the function approach to search for the shortest path. The proposed algorithm can reduce the tourists' traveling time and costs. When the tourists choose certain modes, the output optimal routes will be different. An experiment is performed to verify the feasibility and advantages of the proposed algorithm.

This paper does not focus on the problem of cold start, data sparsity, etc. The main purpose is to study the precise matching and precise recommendations rather than to improve the algorithm's efficiency. Another innovation of the paper is the proposed function vector approach algorithm, which is used to search for the shortest path and finally reduce the tourists' traveling time and costs.

This algorithm could be used as an embedded algorithm for smart tourism recommendation system that is developed, managed, and operated by the tourism sectors and government. Then, the system is ultimately used by tourists on the website. When tourists use the system website, they can input the basic interests and needs to the system, and then it will directly recommends optimal tourist attractions and tour routes for tourists, which is automatic and intelligent. Though the proposed algorithm has advantages, it still has some limitations, which should be further studied in the future. First, the proposed algorithm is based on website big data: the accuracy of the big data mining should be improved, which will help to make the labels more accurate and better match the tourists' interests, improving the recommendation accuracy. Second, the study area of the proposed algorithm is the downtown area of the city, and the transportation modes are limited. In a future study, the study area could be expanded to the subordinate counties and districts of the city, and the transportation modes could be diversified.

**Author Contributions:** Conceptualization, Xiao Zhou, Jian Peng, Jiangpeng Tian, and Mingzhan Su; methodology, Xiao Zhou and Jiangpeng Tian; validation and formal analysis, Mingzhan Su and Jiangpeng Tian; investigation, data resources and data processing, Xiao Zhou, Mingzhan Su, and Jiangpeng Tian; writing—original draft preparation, Xiao Zhou, Jian Peng; writing—review and editing, Xiao Zhou, Jian Peng, Jiangpeng Tian, and Mingzhan Su; visualization, Mingzhan Su; supervision, Jian Peng and Xiao Zhou; project administration and funding acquisition, Jiangpeng Tian; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Military "Double Key" construction project(2021KY05), the Key research and development program of Sichuan province(2020YFG0308), the Major research and development plan(2018GZDZX0010) and the program of Sichuan tourism development research center, key research base of Sichuan federation of sciences association (SCLY-2020-LY20-07).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available from the author upon reasonable request.

**Acknowledgments:** The authors would like to thank the postdoctoral innovation practice base of Sichuan province of Leshan vocational and technical college and Computer science postdoctoral mobile station of Sichuan University. Meanwhile, we thank the editors and reviewers for their valuable comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kato, Y.; Yamamoto, K. A Sightseeing Spot Recommendation System That Takes into Account the Visiting Frequency of Users. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 411. [[CrossRef](#)]
2. Ullah, F.; Shirowzhan, S.; Sepasgozar, S. Smart digital marketing capabilities for sustainable property development: A case of Malaysia. *Sustainability* **2020**, *12*, 5402.
3. Lee, P.; Hunter, W.C.; Chung, N. Smart tourism city: Developments and transformations. *Sustainability* **2020**, *12*, 3958. [[CrossRef](#)]
4. Mou, J.; Luo, G.; Xiong, Z. Application of Collaborative Filtering Algorithm to Tourist Attraction Recommendation. *Softw. Guide* **2017**, *16*, 186–188.

5. Santos, F.; Almeida, A.; Martins, C.; Goncalves, R.; Martins, J. Using POI Functionality and Accessibility Levels for Delivering Personalized Tourism Recommendations. *Comput. Environ. Urban Syst.* **2019**, *77*, 101173. [[CrossRef](#)]
6. Cui, Y.; Huang, C.; Wang, Y. Research on personalized tourist attraction recommendation based on tag and collaborative filtering. In Proceedings of the 2019 China Control and Decision Making Conference, Nanchang, China, 6 June 2019; pp. 381–385.
7. Young, C.; Young, U.R.; Kyeong, K. A Recommender System based on Personal Constraints for Smart Tourism City. *Asia Pac. J. Tour. Res.* **2021**, *26*, 440–453.
8. Li, G.; Zhu, T.; Yuan, T.; Hua, J.; Zhang, H. Recommendation Model of Tourist Attractions by Fusing Hierarchical Sampling and Collaborative Filtering. *J. Data Acquis. Process.* **2019**, *34*, 566–576.
9. Zhou, Y.; Hu, C.; Xiong, H.; Li, L.; Wei, X. Collaborative filtering recommendation algorithm based on label of tourist spots and user preference. In Proceedings of the 2017 2nd International Conference on Machinery, Electronics and Control Simulation, Taiyuan, China, 24 June 2017; pp. 58–65.
10. Nilashi, M.; Ibrahim, O.; Bagherifard, K. A Recommender System based on Collaborative Filtering Using Ontology and Dimensionality Reduction Techniques. *Expert Syst. Appl.* **2018**, *92*, 507–520. [[CrossRef](#)]
11. Shi, R. Design and Implementation of Travel Recommendation System Based on Improved Collaborative Filtering Algorithm. Master's Thesis, Hebei University of Engineering, Handan, China, 2020; p. 12.
12. Chen, S.; Tian, J. Research on Tourist Attraction Recommendation model based on Collaborative Filtering Algorithm. *Mod. Electron. Tech.* **2020**, *43*, 132–135.
13. Zhu, T. Research on Tourist Attractions Recommendation System Based on Deep Collaborative Filtering and Multimodal Analysis. Master's Thesis, East China Jiaotong University, Nanchang, China, 2019; p. 5.
14. Li, Y. Research on the Improvement of Collaborative Filtering Algorithm Based on Tourism Recommendation. Master's Thesis, Qiangdao University of Science and Technology, Qingdao, China, 2019; p. 4.
15. Liu, B. Research on Tourist Attractions Recommendation System Based on Hierarchical Sampling Statistics and Collaborative Filtering. Master's Thesis, East China Jiaotong University, Nanchang, China, 2018; p. 6.
16. Chen, J.; Gu, T.; Chang, L.; Bin, C.; Liang, C. A Tourist Group Recommendation Method Combining Collaborative Filtering and User Preferences. *CAAI Trans. Intell. Syst.* **2018**, *13*, 999–1005.
17. Esmaeili, L.; Mardani, S.; Golpayegani, S.; Madar, Z. A novel tourism recommender system in the context of social commerce. *Expert Syst. Appl.* **2020**, *149*, 113301. [[CrossRef](#)]
18. Shen, J.; Deng, C.; Gao, X. Attraction recommendation: Towards personalized tourism via collective intelligence. *Neurocomputing* **2016**, *173*, 789–798. [[CrossRef](#)]
19. Sasaki, R.; Yamamoto, K. A Sightseeing Support System Using Augmented Reality and Pictograms within Urban Tourist Areas in Japan. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 381. [[CrossRef](#)]
20. Katayama, S.; Isogawa, N.; Obuchi, M.; Nishiyama, Y.; Okoshi, T.; Yonezawa, T.; Nakazawa, H.; Takashio, K.; Tokuda, H. SpoTrip: Evaluation of Information Provision System for Hidden Spots to Promote the Increase Repeat Travellers. *IEICE Tech. Rep.* **2017**, *116*, 185–192.
21. Kang, S.; Lee, G.; Kim, J.; Park, D. Identifying the Spatial Structure of the Tourist Attraction System in South Korea Using GIS and Network Analysis: An Application of Anchor-Point Theory. *J. Destin. Mark. Manag.* **2018**, *9*, 358–370. [[CrossRef](#)]
22. Uchida, K.; Izumi, R.; Kunieda, T.; Yamada, S.; Yonetani, K.; Gotoda, A.; Yaegashi, M. Development “KadaSola”: A Sightseeing Support System for Long Stay. In Proceedings of the 81st Annual Meeting of the Information Processing Society of Japan, Fukuoka, Japan, 14–16 March 2019; pp. 841–842.
23. Mukasa, Y.; Yamamoto, K. A Sightseeing Spot Recommendation System for Urban Smart Tourism Based on Users' Priority Conditions. *J. Civ. Eng. Archit.* **2019**, *13*, 622–640. [[CrossRef](#)]
24. Aoike, T.; Ho, B.; Hara, T.; Ota, J.; Kurata, Y. Utilising crowd information of tourist spots in an interactive tour recommender system. In *Information and Communication Technologies in Tourism*; Pesonen, J., Neidhardt, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; pp. 27–39.
25. Mizutani, Y.; Yamamoto, K. A Sightseeing Spot Recommendation System That Takes into Account the Change in Circumstances of Users. *ISPRS Int. J. Geo Inf.* **2017**, *6*, 303. [[CrossRef](#)]
26. Zhou, J.; Yamamoto, K. Development of the System to Support Tourists' Excursion Behavior Using Augmented Reality. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 197–209. [[CrossRef](#)]
27. Noguera, J.M.; Barranco, M.J.; Segura, R.J.; Martínez, L. A Mobile 3D-GIS Hybrid Recommender System for Tourism. *Inf. Sci.* **2012**, *215*, 37–52. [[CrossRef](#)]
28. Braunhofer, M.; Elahi, M.; Ricci, F. Context-Aware Places of Interest Recommendations for Mobile Users, Design, User Experience, and Usability. *Theory Methods Tools Pract. Lect. Notes Comput. Sci.* **2011**, *6769*, 531–540.
29. Ying, J.J.; Lu, E.H.; Kuo, W.; Tseng, V.S. Urban Point-of-interest Recommendation by Mining User Check-in Behaviors. In Proceedings of the ACM SIGKDD International Workshop on Urban Computing, Beijing, China, 12–16 August 2012; pp. 63–70.
30. Ikeda, T.; Yamamoto, K. Development of Social Recommendation GIS Tourist Spots. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *5*, 8–21. [[CrossRef](#)]
31. Nguyen, L.V.; Jung, J.J. Crowdsourcing Platform for Collecting Cognitive Feedbacks from Users: A Case Study on Movie Recommender System. In *Springer Series in Reliability Engineering*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 139–150.

32. Nguyen, L.V.; Hong, M.S.; Jung, J.J.; Sohn, B.S. Cognitive Similarity-Based Collaborative Filtering Recommendation System. *Appl. Sci.* **2020**, *10*, 4183. [[CrossRef](#)]
33. Meng, S.; Dou, W.; Zhang, X.; Chen, J. KASR: A Keyword-aware Service Recommendation Method on Map reduce for Big Data applications. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 3221–3231. [[CrossRef](#)]
34. Thakkar, P.; Varma, K.; Ukani, V.; Mankad, S.; Tanwar, S. Combining User-based and Item-based Collaborative Filtering using Machine Learning. *Inf. Commun. Technol. Intell. Syst.* **2019**, *15*, 173–180.
35. Liu, H.; Hu, Z.; Mian, A.; Tian, H.; Zhu, X. A New User Similarity Model to Improve the Accuracy of Collaborative Filtering. *Knowl.-Based Syst.* **2014**, *56*, 156–166. [[CrossRef](#)]
36. Bao, J.; Zheng, Y.; Wilkie, D.; Mokbel, M. Recommendations in location-based social networks: A survey. *GeoInformatica* **2015**, *19*, 525–565. [[CrossRef](#)]
37. Zhang, Y.; Shi, Z.; Zuo, W.; Yue, L.; Liang, S.; Li, X.J.N. Joint Personalized Markov Chains with Social Network Embedding for Cold-Start Recommendation. *Neurocomputing* **2019**, *386*, 208–220. [[CrossRef](#)]
38. Xu, M.; Liu, S. Semantic-enhanced and Context-aware Hybrid Collaborative Filtering for Event Recommendation in Event-based Social Networks. *IEEE Access* **2019**, *7*, 17493–17502. [[CrossRef](#)]
39. Li, A.; She, X. *The Principle, Algorithm and Application of Data Mining*, 2nd ed.; Xidian University Press: Xi'an, China, 2012; p. 220.
40. Zhou, X.; Xu, C.; Kimmons, B. Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform. *Comput. Environ. Urban Syst.* **2015**, *54*, 144–153. [[CrossRef](#)]
41. Chen, Y.-C.; Huang, H.-H.; Chiu, S.-M.; Lee, C. Joint Promotion Partner Recommendation Systems Using Data from Location-Based Social Networks. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 57. [[CrossRef](#)]
42. Renjith, S.; Sreekumar, A.; Jathavedan, M. An extensive study on the evolution of context-aware personalized travel recommender systems. *Inform. Process. Manag.* **2020**, *57*, 102078. [[CrossRef](#)]
43. Feng, S.; Li, X.; Zeng, Y.; Chee, Y.M. Personalized ranking metric embedding for next new poi recommendation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 2069–2075.
44. Xie, M.; Yin, H.; Wang, H.; Xu, F.; Chen, W.; Wang, S. Learning graph-based poi embedding for location-based recommendation. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 15–24.
45. Zhao, S.; Zhao, T.; King, I.; Lyu, M.R. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In Proceedings of the 26th International Conference on World Wide Web Companion, Geneva, Switzerland, 3–7 April 2017; pp. 153–162.
46. Memon, I.; Chen, L.; Majid, A.; Lv, M.; Hussain, I.; Chen, G. Travel recommendation using geo-tagged photos in social media for tourist. *Kluw. Commun.* **2015**, *80*, 1347–1362. [[CrossRef](#)]