2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS)

Research on Travel Route Planing Problems Based on Greedy Algorithm

Guanjie Wang
School of Computer and Artificial Intelligence
Huaihua University
Huaihua, China
wgjsci@gmail.com

Yiquan Wang
College of Mathematics and System Science
Xinjiang University
Urumqi, China
ethan@stu.xju.edu.cn

Wei Li*
School of Computer and Artificial Intelligence
Huaihua University
Huaihua, China
*Corresponding author: liwei@hhtc.edu.cn

2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS) | 979-8-3315-3140-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/EIECS63941.2024.10800255

***Abstract*—The greedy algorithm based route planning problem is a method of finding the optimal or near optimal route between a given starting and ending point. This article first uses PCA method to reduce the dimensionality of urban evaluation indicators, extracts key principal components, and KMO and TOPSIS algorithms to reduce the dimensionality of the data. Secondly, for datasets that have not passed the KMO test, a comprehensive evaluation will be conducted using the entropy weight method and TOPSIS method. Finally, based on the greedy algorithm, a route planning algorithm was proposed and optimized to provide personalized route customization according to the different needs of tourists. We also took into account the local travel efficiency, the time required to visit tourist attractions, and necessary daily rest time to reduce costs and avoid falling into the local optimal solution.**

Keywords: Greedy Algorithm; KMO; TOPSIS; route planning

I. Introduction

With the rapid development of China's tourism industry and the continuous strengthening of international exchanges, more and more foreign tourists choose to come to China to experience the rich natural scenery and profound cultural heritage. According to the data of the China Migration Administration, the number of foreigners entering China at all ports increased significantly in 2024, especially the number of tourists coming to China reached 4.361 million. In order to better serve these foreign tourists, China has launched a 144 hour transit visa free policy and implemented it in multiple cities and ports. This policy not only provides more convenience for foreign tourists to travel to China, but also promotes exchanges and cooperation between Chinese and foreign personnel.

In order to enhance the travel experience of foreign tourists, we conducted a study on the attraction routing problem based on greedy algorithm. This project aims to provide scientific and reasonable tourism route planning for foreign tourists through mathematical modeling and algorithm optimization, ensuring that they can visit as many high-quality attractions as possible within a limited time, while optimizing travel costs and time arrangements.

II. Related Work

Nowadays, people are paying more and more attention to personal experience and self needs during the tourism process.[1] Therefore, the use of mathematical modeling methods to automatically generate travel route plans that meet the needs of tourists and help them enjoy a better travel experience is increasingly receiving attention from both academia and industry.

Tourism route planning is one of the popular research issues in the field of smart tourism. The current research work combines the work in operations research with the travel route planning problem in social media, and plans to study it as a variant based on directed off-road or traveling salesman problems. [2-5]

In 2016, Feiran et al. proposed a flexible multi task deep travel route planning framework called MDTRP, which integrates rich auxiliary information to achieve more effective planning.[6] Xinyi et al. proposed an interactive visual analysis method and introduced automatic route optimization algorithms and various interactions to help users optimize and adjust their itineraries for better tourism route planning.[7]

Ana et al. evaluated the theoretical basis of multidimensional concepts of tourist spatiotemporal behavior in 2017 and proposed a conceptual model based on behavioral perspectives and intercity destination backgrounds as a comprehensive analytical framework.[8]Yuan et al. developed complex statistical models using high-resolution and fine-grained spatiotemporal data in 2019, and established multinomial logit (MNL) models to identify factors that affect tourist destination choices.[9] Nithyasri et al. proposed a tourism itinerary generator in 2024. By utilizing user provided details such as destination, budget, and interests, the system can create personalized travel plans.[10]

III. METHODS

In order to simplify the analysis process and focus on the core elements, we used principal component analysis (PCA) technology to effectively reduce the dimensionality of urban evaluation indicators and extract several key principal components. For some datasets that did not pass the KMO test, we innovatively combined entropy weight method and TOPSIS method to achieve comprehensive evaluation and weight allocation of the data. When constructing the route optimization model, we applied greedy algorithm strategy to ensure the optimization and efficiency of the overall tourism planning scheme.

Our data comes from the Fifth "Huashu Cup" National University Student Mathematical Modeling Competition in 2024. [1] It contains 352 cities with 100 attractions in each city, and the information of each attraction contains the name of the attraction, address, description of the attraction, opening hours and so on.

We adopt a comprehensive evaluation system to simplify the process, and classify the city according to multiple dimensions, such as its size, environmental status, cultural heritage, transportation conditions, as well as local climate and culinary characteristics. Specifically, for the indicators that meet the KMO test criteria, we apply principal component analysis to realize their dimensionality reduction; for the indicators that fail to meet this criterion, we introduce the entropy-based TOPSIS method to complete the simplification to one dimension.

In response to the data collected, where direct modeling resulted in a large number of indicators, this study chose to simplify the dataset by means of dimensionality reduction, which in turn led to the construction of an assessment model for analysis. The KMO test was used as a measure of the effectiveness of the different dimensionality reduction methods, which is mainly used to determine whether the correlation between the variables is sufficient to perform factor analysis. The KMO value ranges from 0 to 1. Generally speaking, when the KMO value exceeds 0.6, the dataset is considered to be suitable for factor analysis. The KMO algorithm can be found in Pseudocode 1.

Pseudocode 1 Conceptual KMO Test Process

Input: Dataset X, containing n observations and p variables.
Output: KMO statistic, for assessing the suitability of factor analysis.
Compute Correlation Matrix:
    Calculate the correlation matrix R between all variables in X.
Compute Partial Correlation Matrix:
    Use elements of the inverse of R to compute the partial correlation matrix. ▷ This step involves complex matrix operations, typically automated by statistical software.
Calculate KMO Statistic:
    For each variable i, compute the ratio of the sum of squared partial correlations S_i to the total variance V_i.
    Calculate the KMO statistic: $KMO = \frac{\sum_{i=1}^{p}(V_i - S_i)}{\sum_{i=1}^{p} V_i}$.
Interpret KMO Statistic:
    If KMO is close to 1, it indicates that partial correlations among variables are small, suitable for factor analysis.
    If KMO is low (e.g., less than 0.5), factor analysis may not be appropriate.

The magnitude of the KMO value shows a high degree of shared components among the variables, indicating that the dataset is suitable for factor analysis. Principal Component Analysis (PCA) takes a mathematical dimensionality reduction approach to find a few composite variables to replace the original multitude of variables, so that these composite variables can represent as much information as possible about the original variables and are uncorrelated with each other. [12] The PCA algorithm can be found in Pseudocode 2.

Pseudocode 2 Principal Component Analysis (PCA) Algorithm

Data Standardization:
$X_{\text{norm}} \leftarrow \text{Standardize}(X)$ ▷ Normalize each feature to have mean 0 and variance 1
Compute Covariance Matrix:
$C \leftarrow \frac{1}{n-1} X_{\text{norm}}^T X_{\text{norm}}$ ▷ Covariance matrix
Eigenvalue Decomposition:
$[V, D] \leftarrow \text{EigenvalueDecomposition}(C)$ ▷ V are the eigenvectors, D is the diagonal matrix of eigenvalues
Compute Principal Components:
$PC \leftarrow V$ ▷ Matrix of principal components, i.e., the eigenvector matrix
Select Number of Principal Components:
$k \leftarrow \text{Select}(D)$ ▷ Select based on some criterion (e.g., cumulative variance explained)
Compute Cumulative Variance Explained:
total_variance $\leftarrow \sum_{i=1}^{d} D[i,i]$
cumulative_variance $\leftarrow 0$
for $i = 1$ to k **do**
    cumulative_variance $\leftarrow$ cumulative_variance $+ D[i,i]$
end for
percentage_variance $\leftarrow \frac{\text{cumulative\_variance}}{\text{total\_variance}} \times 100$
Return: $PC[:, 1:k]$ and k

PCA can effectively reduce the dimensionality of variables while retaining the main variation information of the data, and is a commonly used technique for dimensionality reduction. The results of the TOPSIS and KMO downscaling are shown in Figure 1.

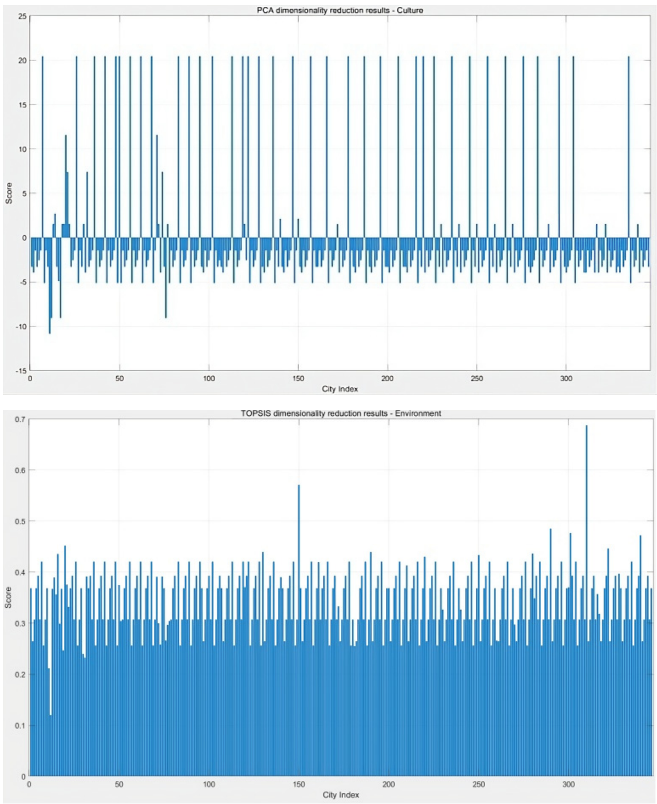

Figure 1. The results of the TOPSIS and KMO downscaling

We comprehensively applied the entropy weighting method and TOPSIS evaluation method, two advanced evaluation tools, to conduct an in-depth analysis of multiple dimensions, including city size, ecological environment, cultural and historical depth, transportation accessibility, climate characteristics, and specialties and cuisines. After a rigorous evaluation process, we have carefully selected 50 cities that are highly attractive to international travelers. This selection not only reflects the comprehensive strength of these cities, but also demonstrates their unique charms and characteristics. TOPSIS is a common and effective multi-attribute decision-making method, which is used to rank the advantages and disadvantages of options by calculating the relative distances of each option from the theoretical optimal point and the theoretical disadvantage point. [4] The TOPSIS algorithm can be found in Pseudocode 3.

Pseudocode 3 Pseudocode for TOPSIS Method

```
procedure TOPSIS(DecisionMatrix, Weights)
    Input:
    DecisionMatrix: Decision matrix where rows represent alternatives and
  columns represent attributes
    Weights: Weights vector for attributes
    Output:
    Ranking: Ranking of alternatives
    // Step 1: Normalize the decision matrix
    NormalizedMatrix ← Normalize(DecisionMatrix)
    // Step 2: Weight the normalized decision matrix
    WeightedMatrix ← MultiplyMatrices(NormalizedMatrix, Weights)
    // Step 3: Determine the ideal best and ideal worst solutions
    IdealBest ← ColumnWiseMax(WeightedMatrix)
    IdealWorst ← ColumnWiseMin(WeightedMatrix)
    // Step 4: Calculate distances from each alternative to the ideal best and
  ideal worst solutions
    for i = 1 to NumberOfAlternatives do
        DistanceToBest_i ← EuclideanDistance(WeightedMatrix[i], IdealBest)
        DistanceToWorst_i ← EuclideanDistance(WeightedMatrix[i], IdealWorst)
    end for
    // Step 5: Calculate the relative closeness of each alternative
    for i = 1 to NumberOfAlternatives do
        Closeness_i ← DistanceToWorst_i / (DistanceToBest_i + DistanceToWorst_i)
    end for
    // Step 6: Rank alternatives based on relative closeness
    Ranking ← SortAlternativesByCloseness(Closeness)
    return Ranking
end procedure
```

The final results are shown in Figure 2 below. The figure shows that the highest rated city is about 0.75, and the subsequent cities have closer ratings, indicating that these cities are relatively close in terms of comprehensive indicators.

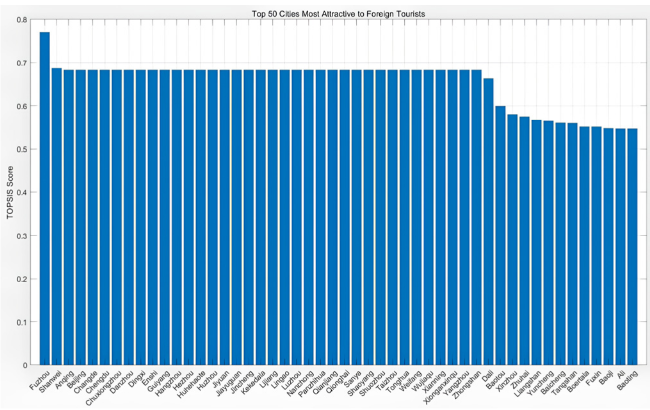

Figure 2. TOPSIS score

We build optimization models by calculating travel distance and time.

High Speed Rail Distance Calculation: The Harversine formula measures the spatial straight-line distance between any two points in a geographic coordinate system. The formula is specifically used to determine the shortest route distance between two locations on a spherical surface.

$$d = 2 \times 6371 \times \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \times \cos(\phi_2) \times \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (1)$$

We made deeper extrapolations and extensions based on the preparatory work. For the part of city and latitude/longitude data: it involves the details of geographic coordinates of 352 cities so that we can accurately calculate the distance between different cities. For the mountain view information data: we compiled information about the most popular mountain views in each city, including the name of the view, its rating, the recommended time to visit, and information about the entrance fee.

1. Data Processing and Initialization: Read and process the city and mountain view data to determine the starting parameters, which involves such aspects as total time constraints, time and cost at this stage, the city where it is located, and a list of cities and landscapes that have been previously traveled.

2. Selection of entry cities: the first town to be reached is selected based on the highest rated mountain scenery.

3. Route planning: A greedy algorithm is used to determine the next city to visit. All the cities that have not yet been visited are examined one by one, the distance from the high speed train to the target city, the travel time and the cost are calculated, and finally the best rated mountain scenery with the lowest ticket price is selected for the tour. Calculate the statistics of the complete travel time, covering the time needed to travel by high-speed rail, move around locally, and visit tourist attractions, and incorporate rest periods. Prioritize the city with the lowest total cost as the next target to visit within the specified time.

4. Update current status: Refresh the current city, date, expenses, list of visited cities and list of sightseeing landmarks. Repeat step 3 if you run out of time or are unable to locate the desired next destination.

5. Output Results: Output the complete travel route, visited attractions, total travel time, total cost and total number of attractions.

The greedy algorithm can find a solution that is close to the global optimum for the tourism route optimization problem in this study. Moreover, due to the selection of local optimal solutions at each step, the greedy algorithm has high execution efficiency and fast computation speed. The specific process of the greedy algorithm is shown in Pseudocode 4.

Pseudocode 4 Travel Planning Algorithm

```
procedure PLANTRIP(initial_city, total_time_limit = 144 hours)
    Initialize:
    current_time ← 0 hours
    current_cost ← 0 yuan
    current_city ← initial_city
    VisitedCities ← {initial_city}
    VisitedSpots ← {}
    total_spots ← 0
    while (current_time < total_time_limit) AND (not reached termination condition) do
        Select Next Best City:
        // Assuming a function to calculate and evaluate cities based on rating, time, and cost
        best_city, best_spot_name ← SelectBestCity()
        if best_city is not None then
            Update Travel Information:
            // Calculate travel time and cost using Haversine formula and train speed/cost
            travel_time, travel_cost ← CalculateTravelTimeAndCost
            current_time ← current_time + travel_time
            current_cost ← current_cost + travel_cost
            Visit the Spot:
            // Set local travel time and sightseeing time
            local_travel_time ← 0.5 hours
            sightseeing_time ← SpotSpecificTime(best_spot_name)
            total_city_time ← local_travel_time + sightseeing_time
            current_time ← current_time + total_city_time
            Rest Time Calculation:
            if (current_time − LastRestTime > 24 hours) AND (current_time + 8 ≤ total_time_limit) then
                current_time ← current_time + 8 hours // 8 hours rest
            end if
            Update State:
            VisitedCities ← VisitedCities ∪ {best_city}
            VisitedSpots ← VisitedSpots ∪ {best_spot_name}
            current_city ← best_city
        else
            Break // No more suitable cities to visit
        end if
    end while
    Output Results:
    // Output VisitedCities, VisitedSpots, final time, and final cost
end procedure
```

Finally, we can get the travel planning route based on the greedy algorithm model, see Figure 3. in making the travel planning, we take the tourists' interest, tour time and budget into consideration in order to provide the most cost-effective travel plan.

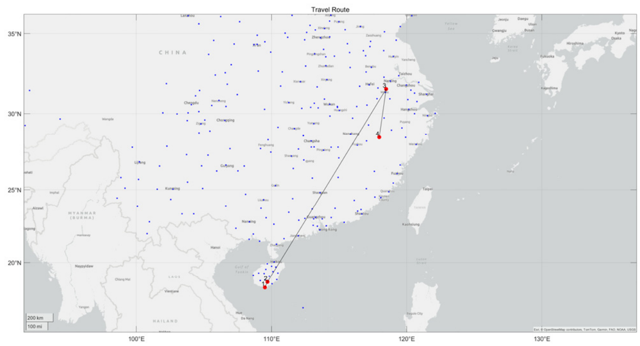

Figure 3. Travel Route

Through the greedy algorithm, it is possible to combine scenic spot rating analysis, urban comprehensive evaluation, route optimization, cost calculation, and tourist preferences to output optimized tourism routes, ensuring the optimization and efficiency of the overall tourism planning scheme.

IV. CONCLUSIONS

The results of this study not only provide scientific and reasonable tourism planning solutions for foreign tourists, but also demonstrate the enormous potential of greedy algorithms in the field of path optimization. In the future, we will continue to optimize algorithms and models to meet the personalized needs of different tourists, and keep up with the changing trends of the tourism market, contributing more wisdom and strength to the sustained high-quality development of the tourism industry. At the same time, we also look forward to promoting this research result to a wider range of tourism demand areas, bringing more tourists a wonderful travel experience.

REFERENCES

[1] Feng W, Bo H, Jinzhou Huang, et al. Research of touring route planning based on spatiotemporal awareness. Operations Research Transactions.

[2] Brilhante I, Macedo J A, Nardini F M, et al. Where shall we go today? Planning touristic tours with tripbuilder. Proceedings of the 22nd ACM international conference on Information and Knowledge Management, 2013:757-762.

[3] Brilhante I R, Macedo J A, Nardini F M, et al. On planning sightseeing tours with trip builder. Information Processing and Management, 2015,51(2): 1-15.

[4] De Choudhury M D, Feldman M, Amer-Yahia S, et al. Automatic construction of travel itineraries using social breadcrumbs. Proceedings of the 21st ACM Conference on Hypertext and Hypermedia.2010:35-44.

[5] Lim K H, Chan J, Leckie C, et al. Towards next generation touring: Personalized grouptours. Proceedings of the 40th International ACM SIGIR, Conference on Research and Development in Information Retrieval. 2017:325-334.

[6] F. Huang, J. Xu and J. Weng, Multi-Task Travel Route Planning with a Flexible Deep Learning Framework, in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 3907-3918, July 2021, doi: 10.1109/TITS.2020.2987645.

[7] Zhang, X., Pang, X., Wen, X. et al. TriPlan: an interactive visual analytics approach for better tourism route planning. J Vis 26, 231–248, 2023.

[8] Caldeira, A. M., & Kastenholz, E. (2019). Spatiotemporal tourist behaviour in urban destinations: a framework of analysis. Tourism Geographies, 22(1), 22–50.

[9] Yuan Li, Linchuan Yang, Han Shen, Zhonglong Wu, Modeling intra-destination travel behavior of tourists through spatio-temporal analysis, Journal of Destination Marketing & Management, Volume 11,2019, 260-269.

[10] N. P. S, N. V, Y. R, G. Kalaiarasi and M. Selvi, "Personalized Travel Itinerary Generator System,"2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024, pp. 1-6.

[11] The Fifth "Huashu Cup" National University Student Mathematical Modeling Competition. https://www.saikr.com/vse/chinamcm/2024

[12] Guo Xianguang. Application of Improved Entropy Method in Evaluation of Economic Result, Systems Engineering-Theory & Practice, 1998, (12):99-103.