



Article A Novel Multi-Objective and Multi-Constraint Route Recommendation Method Based on Crowd Sensing

Xiaoyao Zheng ¹, Yonglong Luo ^{1,*}, Liping Sun ¹, Qingying Yu ¹, Ji Zhang ² and Siguang Chen ³

- ¹ School of Computer and Information, Anhui Normal University, Wuhu 241002, China; zxiaoyao@ahnu.edu.cn (X.Z.); slp620@163.com (L.S.); ahnuyuq@ahnu.edu.cn (Q.Y.)
- ² Faculty of Health, Engineering and Sciences, University of Southern Queensland, Toowoomba 4350, Australia; ji.zhang@usq.edu.au
- ³ Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210000, China; sgchen@njupt.edu.cn
- * Correspondence: ylluo@ustc.edu.cn; Tel.: +86-0553-5910645

Abstract: Nowadays, people choose to travel in their leisure time more frequently, but fixed predetermined tour routes can barely meet people's personalized preferences. The needs of tourists are diverse, largely personal, and possibly have multiple constraints. The traditional single-objective route planning algorithm struggles to effectively deal with such problems. In this paper, a novel multi-objective and multi-constraint tour route recommendation method is proposed. Firstly, ArcMap was used to model the actual road network. Then, we created a new interest label matching method and a utility function scoring method based on crowd sensing, and constructed a personalized multi-constraint interest model. We present a variable neighborhood search algorithm and a hybrid particle swarm genetic optimization algorithm for recommending Top-K routes. Finally, we conducted extensive experiments on public datasets. Compared with the ATP route recommendation method based on an improved ant colony algorithm, our proposed method is superior in route score, interest abundance, number of POIs, and running time.

Keywords: crowd sensing; multi-constraint; route recommendation; multi-objective

1. Introduction

With the rapid development of Internet technology in recent years, the explosive growth of information has increased the burden of retrieving personally interested information and content. The birth of recommended technologies can help people acquire the resources in which they are interested more efficiently. Given the potentially huge returns, major companies such as Google, Amazon, and Taobao, to just name a few, have produced sophisticated and advanced recommender systems and technologies for effectively marketing their products to users. In comparison, due to many factors affecting tourism routes, such as the complex information of real-time traffic flow, weather, and user preferences [1–4], the recommendation of tourist routes is far less mature [5,6].

Due to the usefulness and the inherent complexity of the route recommendation problem, various types of recommendation methods have been proposed. From the perspective of this research work, route recommendation methods can be classified in terms of the number of objectives, the number of points of interest (POIs), and the presence of user constraints, even though there may be other dimensions with which to categorize the existing methods.

From the perspective of the optimization objectives, the existing methods can be divided into single-objective recommendation and multi-objective recommendation ones. Single-objective recommendation only focuses on the aspect of user requirements, and uses path planning algorithms to find the optimal Top-K routes [7,8]. Kurashima et al. [7] incorporated user preferences and location information into probabilistic behavior model by



Citation: Zheng, X.; Luo, Y.; Sun, L.; Yu, Q.; Zhang, J.; Chen, S. A Novel Multi-Objective and Multi-Constraint Route Recommendation Method Based on Crowd Sensing. *Appl. Sci.* **2021**, *11*, 10497. https://doi.org/ 10.3390/app112110497

Academic Editor: Michele Girolami

Received: 23 July 2021 Accepted: 26 October 2021 Published: 8 November 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/). combining the topic model and the Markov model, and proposed a personalized route recommendation method which takes the optimization of user preferences as a single objective. Wang et al. [8] combined the merits of A^{*} search algorithms and deep learning to solve the personalized route recommendation problem. It can automatically learn the cost functions without handcrafting heuristics, but it can only complete the route recommendation with a single objective. Multi-objective route recommendation has been a hot topic in recent years. Many travel route recommendations are by nature multi-objective problems, composed of multiple conflicting or interacting objectives at the same time. For example, a tourist might want to visit as many scenic spots as possible but spend as little as possible. These two objectives are apparently conflicting. Currently, multi-objective evolutionary algorithms are mainly used to solve the problem of multi-objective optimization problems, but the research on route recommendation is far less mature [9,10]. Zuo et al. [9] proposed a multi-objective recommendation model which optimizes two conflicting performance metrics termed accuracy and diversity. Wang et al. [10] proposed a multi-objective evolutionary algorithm that aims to find a set of trade-off solutions by optimizing two objective functions simultaneously, which can balance accuracy with unpopular items. Zhou et al. [11] proposed a biobjective bilevel programming model to optimize road charging efficiency and the total cost of vehicle emissions, which applied a nondominated sorting genetic algorithm to solve the above two optimization problems. The traditional solutions still use single-objective models in tour route recommendations, so they cannot meet the needs of users. From the perspective of the spatial scale of the tourist routes recommended, the existing methods can be divided into micro-scale route recommendation within a specific scenic spot (i.e., single POI) and recommendation of multiple scenic spots on the macro scale (i.e., multiple POIs), which is more complicated than the single-POI scenarios. Wang et al. [8] algorithmically processed multiple sub-attractions of a scenic spot on a microscopic scale to recommend the optimal route for tourists. Due to the constraints of the area of the attraction, tourists' queuing at the scenic spots becomes a very important concern, so it is recommended to score the most efficient walking distance to the scenic route. Pop et al. [12] proposed a route choosing behavior method of crossroads based on the genetic algorithm, which uses a new fitness function computation method. Yu et al. [13] recommended serialization routes for multiple attractions in a city on a macro-scale. From the perspective of whether constraints are present, the existing methods can be divided into recommendations based on user preference constraints [1,4,7,13-15] and recommendations without user preference constraints. The constraints in route recommendation depend on the characteristics of the point of interest and the individual needs of the user, such as travel expenses, time, and transportation. In addition, effective and accurate evaluations of POIs can make the constraints better meet the needs of users. By collecting the historical data of users in social networks, using the theory of crowd sensing can complete the evaluation of each POI more comprehensively and accurately [16–18].

In general, the optimization problem solving involved in the tour route recommendation is complicated, and it is often necessary to find an optimal or a sub-optimal solution in a huge search space. The route optimization in our research work is similar to the TSP problem, but there lies an important difference: as the locations of tourists change, the attraction weight of users to each POI will change; that is, the weight of the edge will change dynamically with the locations of tourists [19–21]. Although dynamic programming and backtracking methods can be used to mitigate the high time complexity caused by exhaustive solutions, those methods cannot be directly applied to the multi-constrained route recommendation problem, because they cannot find the optimal solution in multiple objective functions with conflicts. In order to solve the problem of personalized recommendation of tourism routes under multiple objects and constraints, our work focuses on how to efficiently recommend multi-objective and multi-constrained tourism routes which are still effectively in line with user preferences.

The contributions of this paper are summarized as follows. Firstly, given the shortcomings of multi-POI route recommendation research and the difficulty of quantifying user preference constraints, a multi-objective and multi-constraint model is proposed that optimizes the obstacles and time fluctuations of urban road networks in the real world so as to establish an accurate model. Secondly, based on the theory of crowd sensing theory, we quantify the scores and distributions of POIs on tourism routes, and calculate the popularity of scenic spots by weighting. Finally, we propose a novel multi-objective and multi-constraint route recommendation method based on crowd sensing. The performance of our proposed method has been evaluated on publicly available datasets, and it was promising.

The remainder of the paper is organized as follows. We describe several typical tour route methods, the POI scoring mechanism, and the route constraint model in Section 2. We give a detailed introduction of problem definition, and propose a novel score mechanism based on crowd sensing method and a multi-constraint model in Section 3. In Section 4, we present a multi-objective variable neighborhood search optimization algorithm, which can realize Top-K route recommendation efficiently. We present a performance comparison of our algorithm with the well-known ATP algorithm and provide the evaluation results in Section 5. In Section 6, the paper is summarized, and the future research directions are described.

2. Related Work

2.1. Tour Route Recommendation

Due to the complexity of traditional shortest path solutions for tourism routes, some researchers [2,5] use the improved A* algorithm to find the shortest paths. Boonsri et al. [22,23] employed the Dijkstra algorithm and Yen's algorithm to generate the shortest path of the specified graph. The path generated by the above work was only for the shortest time or distance, and did not take into account the individual needs of users. Therefore, many researchers consider users' preferences as an important factor in tourism routes. Kotiloglu et al. [7] proposed an FFTS framework to solve the problem of user-specific multi-time-window tour route recommendation. In addition, an iterated tabu search algorithm has also been proposed which uses a greedy constructive heuristic algorithm based on serial insertion to generate a feasible initial solution, and generates a set of routes that meet the requirements from the must-see attractions and optional spots. However, the iterative mechanism led to a longer running time and higher algorithmic complexity. Zheng et al. [8] proposed a four-step heuristic algorithm for personalized one-day travel. First, the algorithm encodes the route; then it uses the improved greedy algorithm to construct the initial route, and employs a genetic algorithm and a differential evolution algorithm to optimize the route. Finally, the generated route is evaluated and iterated repeatedly within the set threshold. In addition, by adding tourist fatigue to the user preferences model and taking the time window as the influencing factor of attractions, it provides a selection strategy for different groups of people for route recommendation. Cao et al. [24] proposed a keyword-aware optimal route search system (KORS) covering multiple types of POIs, which provides users with the service of querying POIs to generate optimal routes, but KORS is unable to generate a route that matches preferences for users who know little about a city's information. Luan et al. [25] proposed a maximal-marginalrelevance-based personalized trip recommendation method that can recommend travel itineraries to users according to their preferences in an acceptable time. In addition, an ant-colony-optimization based trip planning algorithm named ATP was developed to balance trip quality and computational complexity. However, the above recommendation algorithm can only recommend single-target routes, and cannot help solve the problem of multi-objective and multi-constrained tour route recommendation.

2.2. POI Scoring Mechanism

The development of location-based social networks (LBSNs) has produced a wealth of historical information related to users, such as geographical locations and user preferences. Many studies [9,13,26–28] have utilized the check-in data generated by social tools, such

as Foursquare, Flicker, and Facebook, or photo data with geo-tagged information, as a scoring basis to generate personalized tour routes based on user preferences. However, when utilizing a simple popularity score as the basis for recommendations, the produced routes cannot adequately satisfy the personal interests of users. Some researchers [29,30] have computed the matching degree between user preferences and POIs and utilized collaborative filtering to predict POIs that users may be interested in and generate tour routes based on user preferences. Su et al. [31] leveraged crowd sourcing and utilized the knowledge of domain experts to solve complex route recommendation problems. Luo et al. [32] proposed a crowdsourcing-based route recommendation algorithm that utilizes existing route mining algorithms and candidate routes generated by map service providers, which employs domain knowledge to obtain the route and fits user's preferences. All of the above work utilizes the popularity of POIs, or user preferences, or domain experts to recommend routes, and thus cannot comprehensively measure the objective and subjective scores of attractions. An objective and reasonable measure of the popularity of POI is the basis for resolving tour route recommendations.

2.3. Route Constraint Model

When considering user preferences, some researchers proposed various constraints, such as time constraints, cost constraints, distance constraints, and open time constraints of attractions [33]. Kotiloglu et al. [7] adopted the Euclidean distance, which ignores the obstacles of the real-world urban road network, so that the errors generated in the route recommendation are relatively large. Chen et al. [4] proposed a two-stage personalized travel planning algorithm named TripPlanner, which combines Foursquare data and taxi trajectory data based on social network data. It can provide three types of constraints: time, must-see, and meal time. In addition, the system takes the opening hours of POIs and the travel times between POIs into account. However, in TripPlanner, the POIs to be added only need to meet the open time to join the candidate route; it fails to consider the fact that users must arrive before the POIs is opened. Bao et al. [30] took into account the travel time of scenic spots, the travel times between scenic spots, and so on, but they did not consider the costs and the number of scenic spots desired. Therefore, it is also not possible to recommend a personalized tour route for the user that way. Luan et al. [25] considered not only time and cost, but also POI diversity in the design of their tour route recommendation model. Therefore, it features better performance in diversity of recommended routes. As the constraints model has a great influence on the accuracy of recommended routes, it is very important to establish a complete multi-constraints model for tourism route recommendation.

3. Preliminaries

3.1. Problem Definition

Definition 1 (User preference tag set). Different users have different preferences. For example, user u_1 likes gardens, historical buildings, and monuments; user u_2 loves distinctive areas, memorials, and attractions of city parks. Define the user u_d 's preference tag set as $LU^d = \left\{ LU_1^d, LU_2^d, LU_3^d, \dots, LU_p^d \right\}$.

Definition 2 (Attractions label set). Similarly to the user interest tag set, the attraction a_i tag set is $LA^i = \{LA_1^i, LA_2^i, LA_3^i, \dots, LA_q^i\}$. The attraction tag has a certain relevance to the user interest tag.

Definition 3 (Target Set). The target set provided by the user u_d is $Obj^d = \{obj_1^d, obj_2^d, obj_3^d, \cdots, obj_x^d\}$, Obj^d is determined by the user and the route attribute. Targets can be route costs, route times, route scores, and more.

Definition 4 (Route Set). Route = $\{rt_1, rt_2, rt_3, \dots, rt_y\}$ is the route set generated by the route recommendation algorithm. In this paper, the recommended route is displayed in front of the user in the form of TOP-K. In addition, the types of attributes contained in each route are represented by RL, such as route score, route time, route cost, and number of route attractions, where $\forall rt_y \in Route, Obj^d \subseteq RL$.

Definition 5 (Route Network). The starting point of user u_d and attractions constitute an undirected graph $G\langle v, w \rangle$, where v is the attraction set and w is the edge set where the attractions are connected to each other. Each node contains multiple attributes, such as ID, latitude, longitude, and score. Although w can be expressed by Euclidean straight-line distance, it is quite different from the actual urban road. Therefore, we build a network dataset and use ArcMap to generate an OD matrix, which obtains all edges in the undirected graph with real road network distance. Therefore, the problem of this paper can be abstracted into finding K optimal travel routes from $G\langle v, w \rangle$ that satisfy users' multiple constraints and recommending them to users.

3.2. Score Mechanism

In order to quantify the attractiveness of scenic spots to different tourists, we introduce the interest tag matching score, the comprehensive crowd sensing scores, and the distance dynamic score, which can be used as the input for generating route of recommendation algorithm.

3.2.1. Interest Tag Matching Score

Taking the scenic spot as an example, suppose the user u_d sets his own preferences to P tags, such as monuments, historical buildings, museums, and temples, and the total number of target attractions is m, thereby establishing a Boolean matrix \mathcal{R} with dimensions m * n. Since each attraction has its own unique label, we match the tag set owned by attraction a_i according to the order of interest tags listed by the user. If a certain tag LA_q^i of attraction a_i is in the tag set LU_d of the user u_d , the matrix element value is set to 1, and otherwise 0. After the matching is completed, the ratio of the total number of matrix elements with matrix element value 1 and the number of elements of user u_d 's interest tag set is calculated, thereby obtaining the interest matching score G_i^I of the user u_d for the attraction a_i .

3.2.2. Crowd Sensing Scores

1. Crowd Sensing Social Score. Generally speaking, the evaluation of netizens reflects the reputation and characteristics of the attractions to a certain extent. The rating of the scenic spot a_i is represented as N_i^A , and the rating of the user u_j 's interest the attraction a_i is r_{ij} , which is synthesized by ratings and text reviews. Then, the crowd sensing social score of the attraction is expressed as follows:

$$G_{i}^{A} = \frac{\sum_{y=1}^{N_{i}} r_{ij}}{N_{i}^{A}} \times \lg(N_{i}^{A} + 1)$$
(1)

In this paper, the same quantitative method is used to obtain the group intelligence perception social scores G^H and G^R of hotels and restaurants. Min-max standardization processing is performed to standardize the score.

2. Crowd Sensing Location Score. According to statistical analysis, the distribution of attractions is closely related to the location distribution of restaurants and hotels. The total number of restaurants and hotels around the scenic spot represents its popularity to some extent. Based on this reasoning, the number of restaurants and hotels distributed within a given radius centered on the attraction is used as the attraction location score. Assume that the number of restaurants in the radius R of

the attraction a_i is N_i^R , and the number of hotels is N_i^H . The crowd sensing location score in radius *r* of scenic spot a_i is as follows:

$$G_{i}^{L} = \frac{\sum_{t=1}^{N_{i}^{R}} G_{t}^{R}}{N_{i}^{R}} \times \lg(N_{i}^{R}+1) + \frac{\sum_{t=1}^{N_{i}^{H}} G_{t}^{H}}{N_{i}^{H}} \times \lg(N_{i}^{H}+1)$$
(2)

3. Comprehensive Crowd Sensing Score. As the interest tag matching score contains the user's personal preference information, the comprehensive crowd sensing score is obtained from the user rating and POI geographic location. By combining the social score and location score of crowd sensing, the comprehensive crowd sensing score of scenic spot a_i is obtained, which is represented as:

$$G_{i}^{P} = \alpha \frac{2G_{i}^{A}G_{i}^{L}}{G_{i}^{A} + G_{i}^{L}} + (1 - \alpha)G_{i}^{I}$$
(3)

where α is weight parameter to balance the crowd sensing score and interest tag matching score.

4. Distance Dynamic Score. In order to introduce the influence of dynamic space-time on the accessibility of tourists, the space-time accessibility value between users and attractions is defined as:

$$G_i^D = -e^{\lambda \frac{D_i}{V_k}} \mu \tag{4}$$

where λ is the distance attenuation coefficient; D_i is the average road network distance from other attractions to the scenic spot a_i ; V_k represents the driving speed of the car, which determined by the road level; and μ is the weight. According to the comprehensive crowd sensing score G_i^P of the scenic spot a_i and the time-space accessibility value of the attraction, the metric G_i^S of the user-selected scenic spot a_i in the recommendation algorithm is expressed as follows.

$$G_i^S = \theta G_i^P + (1 - \theta) G_i^D \tag{5}$$

3.3. Multi-Constraint Model

In tour route recommendation, users often give some specific requirements, including a travel time limit, a cost limit, and so on. These requirements reflect the personalized needs of users. Based on this, we propose a multi-constraint model, which consists of three parts: time constraint, cost constraint, and route attribute constraint.

1. Time Constraint. Let P_d^s be the starting position of user u_d and T_d^s be the departure time. The distance between attractions a_i and a_j is $T(a_i, a_j)$. The user's arrival time at the first attraction is marked $T(P_d^s, a_1)$. The arrival time at scenic spot a_j is denoted as T_j^{arr} , and the total travel time spend by the user is denoted by T_d^{sum} . Then, scenic spot a_j is added to the tour route. The time constraint model should be satisfied as follows:

$$\begin{cases} T_j^{ve} + T_j^v + T(a_j, P_d^e) \le T_d^{sum} \\ T_j^{ve} - T_j^{arr} \ge T_j^v / 2 \\ T_j^b \le T_j^{arr} \end{cases}$$
(6)

where T_j^b denotes the opening time of a_j , T_j^{ve} denotes the tour end time of a_j , T_j^v denotes the recommended visiting time for a_j , and the minimum play time should meet half of the recommended visiting time in our model. $T(a_j, P_d^e)$ is the time from the last attraction a_j to the destination P_d^e via the recommended route.

2. Cost Constraint. The sum of POI consumption in the recommended route of the visitor should be less than or equal to the upper limit given by the user, which represented as:

$$\sum_{i=1}^{n} C_j \le C_d^{sum} \tag{7}$$

3. Multi-objective constraints. Considering the user's target preference, obj_d is a collection of goals given by the user u_d , which can be expressed as:

$$\exists V(rt_v, obj_w^d) = min \oplus max\{V(rt_v, obj_w^d) | rt_v \in Route, obj_w^d \in Obj^d\}$$
(8)

where $V(\cdot)$ is the route attribute value under the specified target.

4. Multi-Constraint Model. By considering all the above three constraints, the multiconstraints model of the overall tourism route is obtained as follows.

$$\begin{pmatrix}
T_j^{ve} + T_j^v + T(a_j, P_d^e) \leq T_d^{sum} \\
T_j^v \leq T_j^l \\
T_j^b \leq T_j^{arr} \leq T_j^e - T_j^v / 2 \\
\sum_{j=1}^n C_j \leq C_d^{sum} \\
\exists V(rt_v, obj_w^d)
\end{cases}$$
(9)

4. Multi-Objective and Multi-Constraint Route Recommendation Algorithm

Aiming at the multi-constrained multi-objective tour route recommendation problem, we propose a multi-objective variable neighborhood search optimization algorithm, named MOVNS, which realizes Top-K route recommendation. The algorithm consists of two components: variable neighborhood search algorithm (VNS) and hybrid particle swarm genetic optimization algorithm (HPSG). The main idea of the algorithm is as follows. Firstly, the VNS algorithm is used to generate the basic route satisfying the multi-objective and multi-constraints of users as the initial population. On this basis, the HPSG algorithm is used to optimize the basic route, and the Pareto optimal solution is obtained by using the fast nondominated sorting algorithm [33]. Finally, the top routes are selected from the Pareto optimal set and recommended to users.

4.1. Variable Neighborhood Search Algorithm

The VNS algorithm mainly consists of four major steps:

Step1. Divide the attractions set *A* into *numN* neighborhood sets, every of which contains *numA* attractions. The sum value of attractions' comprehensive crowd sensing score G_i^p in each neighborhood set is regarded as its weight; then, all *numN* neighborhood sets' weights are normalized by the random probability, where the probability sum of all neighborhood sets is 1.

Step2. Select the neighborhood set to be searched in the way of roulette, and select the POI as a candidate point to join the candidate route according to user's target. The target can be the highest route score or the largest number of route attractions. That is to say, according to the previously selected POIs, continue to search for POIs that meet the user's requirements in the neighborhood until the end of the loop.

Step3. Divide the attractions set *A* into *numN* neighborhood sets randomly again. Candidate POIs that were added to the candidate route are used as references. In this way, the rating values of attractions in each neighborhood set are dynamically changed, which adjusts the neighborhood set's weights. Thus, the selected probability of a neighborhood set is updated.

Step4. Redo Step1 to Step3 *firstThreshold* times; thus, the candidate route set can be generated.

The details of the algorithm NVS are shown in Algorithm 1.

Algorithm 1 Variable neighborhood search algorithm.

Input: A, res, numN, numR, obj, firstThreshold **Output:** Route // Route satisfying constraints in the first stage 1: **while** *count* < *numR* **do** $ngb \leftarrow DivideRandomly(A, numN) / / divide A into numN sets$ 2: 3: *StartProbability*(*A*, *ngb*, *res*) 4: $selectNgb \leftarrow Roulette(ngb)$ 5: $preLoc \leftarrow SPO(A, res, ngb, selectNgb, obj) / / Select POIs according to user's targets$ 6: if $preLoc \ge 0$ then 7: *route* \leftarrow *route* + a_{preLoc} and $A \leftarrow A - a_{preLoc}$ while *iterator* < *firstThreshold* **do** 8: 9: $postLoc \leftarrow PO(A, res, ngb, preLoc) / / Update neighborhood set and select POIS$ 10: if $postLoc \ge 0$ then route \leftarrow route $+ a_{postLoc}$ 11: 12: ChangeState(A) and $preLoc \leftarrow postLoc$ 13: end if 14: end while 15: end if 16: end while

The symbols and functions in Algorithm 1 are defined as follows. *A* is the attractions set, *res* is the constraint provided by the user, *numN* is the number of neighborhoods, *numR* denotes the number of recommendation routes, *obj* denotes the selected object by the user, and *firstThreshold* is the upper limit of iteration of the variable neighbor. The function of *DivideRandomly*(*A*, *numN*) is to divide the attractions set *A* into *N* neighborhoods randomly. The function of *StartProbability*(*A*, *ngb*, *res*) is to obtain the normalized probability of each neighborhood with reference to the starting point. The function of *SPO*(*A*, *res*, *ngb*, *selectNgb*, *obj*) is to obtain the index of the POI with the maximum score from set *A* according to the selected object. The function of *PO*(*A*, *res*, *ngb*, *preLoc*) is to update the score, re-calculate the probability of each neighborhood, and re-select the neighborhood that meets the user constraints through roulette algorithm dynamically. The function of *ChangeState*(*A*) is to update attractions set *A* to be selected.

4.2. Hybrid Particle Swarm Genetic Optimization Algorithm

A conventional genetic algorithm can not be directly used to solve the proposed multiobjective and multi-constraint Top-K personalized tour route recommendation. A hybrid particle swarm genetic optimization algorithm using sequential crossover operator is proposed in this paper. The algorithm crosses selected chromosomes with global extremum randomly to realize the function of mutation, and the crossover operator is selected to optimize the algorithm by using the mechanism of roulette. Secondly, according to the multi-constraints model, a new route set is generated iteratively based on the candidate route set generated by Algorithm 1. Finally, the Pareto optimal set is obtained by fast nondominated sorting algorithm, and the Top-K tour route recommendation is realized. The algorithm's steps are shown in Algorithm 2.

Note that the candidate recommendation routes are generated using the VNS algorithm. Then, the particles, i.e., tour routes, are encoded, and the individual extremum and global extremum are found based on the idea of particle swarm optimization. Roulette method is used to select two parents, *tempRoute* and *anotherRoute*; then it crosses the two parents with a certain probability and selects the offspring according to the fitness. Next, we cross the offspring and the global extremum *gBestRoute* with a certain probability, i.e., mutation. Then the system judges whether the final offspring meets the user's constraints, calculates the fitness of the particles, determines whether to accept the new chromosome and replicate it, iterates *secondThreshold* times, and generates *numR* routes under a single objective. Finally, for each object, the routes generated by hybrid particle swarm optimization algorithm are merged, and the Pareto optimal set is obtained by using a fast nondominated sorting algorithm.

The symbols and functions in Algorithm 2 are defined as follows. The symbolic definitions for *A*, *res*, and *obj* are the same as in Algorithm 1. *secondThreshold* is the upper limit of iteration of Algorithm 2 in the search stage, *route* denotes the candidate route set generated by the variable neighborhood search algorithm. The function of *InitPBestRoute(Route,obj)* is to initialize the individual extremum based on user objects. The function of *FindPBestLoc* (*pBestRoute,obj*) is to find global extremum subscripts. The function of *Roulette* (*pBestRoute*) is to obtain two parents for crossing by the roulette algorithm. The function of *Crossover* (*tempRoute, anotherRoute*) is to select two paternal chromosomes for crossover. The function of *Mutation(tempRoute, gBestRoute*) is to cross a temp chromosome with a global extremum as a variation. The function of *CalFitness(A, tempRoute, obj*) is to calculate the particle fitness and select the fitness according to *obj*. The function of *Update* (*Route, tempRoute, pBestRoute, fitness, curPar, obj*) is to update the particle. If the fitness is high, it is accepted; otherwise, it is discarded. The function of *FNS(Route'*) is to use fast non-dominated sorting algorithm to obtain the Pareto optimal set.

Algorithm 2 Hybrid particle swarm genetic optimization algorithm.

Input: A, res, Route, obj, secondThreshold **Output:** *TopKRoute* // recommended route in the second stage 1: *pBestRoute* ← *InitPBestRoute*(*Route*, *obj*) 2: gBestLoc ← FindPBestLoc(pBestRoute,obj) 3: if *gBestLoc* \geq 0 then 4: $gBestRoute \leftarrow pBestRoute[gBestLoc]$ 5: while *iterator* < *secondThreshold* do $tempRoute \leftarrow Roulette(pBestRoute)$ 6: 7: anotherRoute \leftarrow Roulette(pBestRoute) **if** *Route*[*curPar*] > 0 **then** 8: 9. $tempRoute \leftarrow Route[curPar]$ 10: end if *list* \leftarrow *Crossover*(*tempRoute*, *anotherRoute*) 11: 12: $list \leftarrow Mutation(tempRoute, gBestRoute)$ 13: if Judge(A, res, list) = false then 14: Start a new iteration 15: end if $fitness \leftarrow CalFitness(A, tempRoute, obj)$ 16: Update(Route, tempRoute, pBestRoute, 17: fitness, curPar, obj) 18: if $gBestLoc \ge 0$ then 19: $gBestRoute \leftarrow pBestRoute[gBestLoc]$ 20: end if 21: end while 22: end if 23: $Route' \leftarrow Merge(Route)$ 24: $TopKRoute \leftarrow FNS(Route')$

4.3. Algorithm Complexity Analysis

The complexity of the proposed MOVNS algorithm mainly involves executing the algorithms of VNS and HPSG. Suppose the number of attractions is *n*. The complexity of VNS is mainly divided into *numN* neighbors from scenic spots. The complexity of this part is O(An), where *A* is a constant. Next is the complexity of iteratively generating neighbors and sifting out *numR* routes, which is O(firstThreshold * numR). The total complexity of these two parts is O(firstThreshold * numR + An). The HPSG algorithm consists of the following parts: The complexity of finding the best parent is O(BnumR), where *B* is a constant. Then it iteratively generates the complexity of single-objective candidate routes in *numR* seed routes, which is O(secondThreshold * numR). Finally, the complexity

of generating Top-K recommended routes by FNS sorting algorithm is $O(numR \log_2^{numR})$. Hence, the complexity of HPSG is $O(secondThreshold * numR + BnumR + numR \log_2^{numR})$ and the complexity of the whole algorithm is $O(firstThreshold * numR + secondThreshol * numR + BnumR + numR \log_2^{numR} + An)$

5. Experiment

5.1. Experimental Datasets

Our experimental datasets were derived from two major domestic portal sites, Ctrip and Dianping. The scenic spot data comes from Ctrip, which has attributes of name, rating, opening time, type, recommended playing time, and so on. We have collected real data from 200 scenic spots, 8171 restaurants, and 2528 hotels in Beijing before August 2017, which were used for our experiments [34].

5.2. Experimental Setup

Although tour route recommendation algorithms have been intensively studied, there have been few research efforts on multi-objective and multi-constrained tour route recommendations. To the best of our knowledge, only the ATP algorithm [25] can achieve multi-objective Top-k route recommendation. The ATP algorithm adopts the ant colony optimization method to obtain an approximately optimal solution which can balance trip quality and computational complexity. Therefore, we only choose the ATP algorithm to compare with our proposed MOVNS in this paper.

The experiment was run on a workstation with a 3.4 GHz Core i7 processor, 16 GB of memory, and a Windows 10 operating system. The radius of crowd sensing location score was set to 2 km. The default value of balance parameter α in the comprehensive crowd sensing score was set to 0.5, and θ was set to 0.5. In the MOVNS algorithm, the default value of *numN* is 4, *numR* is 100, the iteration number *firstThreshold* is 100, and *secondTheshold* is 2000.

The running time, route scores, interest abundance (IA), and number of POIs in route were used to measure the performance of the comparison algorithms. Route score refers to the total score of each POI in the recommendation route. A high score indicates a good quality route. IA is a comprehensive evaluation index for recommended routes, including the travel time, the route score, the number of POIs, and the type number of scenic spots. The number of POIs refers to the number of POIs included in the recommendation route, which is of great significance to measure whether the route recommendation algorithm is reasonable. The running time is an important index with which to measure the efficiency of the algorithm. From the perspective of the quality of recommendation routes, we used route score, interest abundance, and the number of POIs en route to measure. From the perspective of the running efficiency of the algorithms, we used the running time to measure the efficiency of the comparison algorithms. Therefore, the advantages and disadvantages of each algorithm were analyzed. The specific mathematical definition of IA denoted by Φ in this paper is as follows.

$$\Phi = \begin{cases} \frac{\sqrt{KS}}{T} \cdot \sum_{i=1}^{M} G_i^{PA} \times \frac{(M^{up} - M)}{MM^{best}} & M^{best} < M\\ \frac{\sqrt{KS}}{T} \cdot \sum_{i=1}^{M} G_i^{PA} \times \frac{1}{M} & M \le M^{best} \end{cases}$$
(10)

where *T* represents the total route time, *M* represents the number of attractions included in the route, M^{best} is the best number of scenic spots in the recommended route, *K* is the number of types of attractions included in the route, and *S* is the proportion of labels of all attractions in the recommended route that match user-provided interest labels.

5.3. Experimental Results

5.3.1. Sensitivity Analysis

This section focuses on the sensitivity study of the balance coefficient α between crowd sensing score and interest tag matching score, the balance coefficient θ between space-time accessibility value and the comprehensive crowd sensing score of scenic spots, the number of neighborhoods *numN*, the number of iterations *firstThreshold* of the VNS algorithm, and the iteration number *secondThreshold* of the HPSG algorithm in our proposed MOVNS method. We analyze the effects of their values on the performance of the MOVNS algorithm.

As shown in Figure 1a, the IA reached a superior value when α was between 0.1 and 0.5; the best performance was when α reached 0.5. From Figure 1a, we can see that the balance coefficient was α about 0.5 to reach the optimum. The results show that the mechanism of crowd sensing can effectively improve the recommendation performance of tour routes.

As is depicted in Figure 1b, when θ was between 0 and 0.4, the IA was on the rise and reached its maximum at 0.4. IA remained stable when θ was greater than 0.5. The results indicate that the space-time accessibility value can enhance the recommendation effect on the performance of recommendation and improve recommendation efficiency when θ is in [0.4,1].



Figure 1. The influences of parameters on the MOVNS algorithm. (a) The influence of parameter α on IA. (b) The influence of parameter θ on IA. (c) The influence of neighborhood number on IA. (d) The influence of parameter *firstThreshold* on IA. (e) The influence of parameter *secondThreshold* on IA. (f) The influence of particle number on IA.

Figure 1c shows that when the number of neighborhoods was 2, 4, or 5, the IA was higher. With the increase in the number of neighborhoods, the IA fluctuated and decreased. Therefore, the number of neighborhoods should not be set too high. We set it to four in this experiment.

As shown in Figure 1d, when the number of iterations *firstThreshold* of VNS increased to 10, the IA reached the optimum and began to converge. The experimental results show that VNS has good convergence.

According to Figure 1e, when the number of iterations *secondThreshold* in HPSG was less than 600, the IA tended to increase, but its range was large and very unstable. As the iterations increased to 2000, the IA became much more stable than before. From Figure 1f, it can be seen that the IA fluctuated and converged when the number of particles was greater than 60, and reached the optimal state.

5.3.2. Comparative Experiments on Different Metrics

This section compares our MOVNS algorithm with the multi-objective constraint route recommendation algorithm ATP, proposed by Luan [25]. The running time, route scores, IA, and number of scenic spots produced by MOVNS were compared ten times in each experiment with ATP.

Figure 2a shows that the running time of MOVNS was about 1 s, which was much less than that of ATP. Figure 2b shows that the IA index of the proposed algorithm fell in the range [0.15, 0.2], which is higher than that of ATP. Figure 2c shows that the recommended route score of MOVNS ranged in [3.6, 3.9]. Compared with ATP, the route score of MOVNS algorithm was higher. As shown in Figure 2d, ATP generated routes consisting of up to six scenic spots, which cannot meet the needs of users to visit more scenic spots within a limited time. MOVNS is more flexible in the number of scenic spots recommended by the route. The time complexity of ATP algorithm compared in our paper is O(RNKn), where *R* is the total number of iterations, *N* is the total number of ants, *K* is the number of POIs in the dataset, and *n* denotes the number of elements in the POI set. The ATP algorithm is higher in time complexity compared with our proposed algorithm. In addition, the ATP algorithm is a route recommendation algorithm based on ant colony optimization, and its pheromone update is a positive feedback process which rapidly expands the initial pheromone difference, so it can easily fall into a local optimum and fail to reach the global optimal. The advantage of the algorithm proposed in our paper is generating globally diverse solutions conforming to constraints through the first stage of the variable nearest neighbor algorithm. In addition, it makes the diversity of solutions better, and does not easily fall into local optima through cross-variations of the genetic algorithm.



Figure 2. Performance comparison between MOVNS and ATP. (**a**) Running time comparison. (**b**) AI comparison. (**c**) Route score on comparison. (**d**) Number of scenic spots comparison.

5.3.3. Comparative Experiments with Multiple Objectives

We set four objectives for the algorithm, namely, user cost, play time, route score, and number of scenic spots; and compare the route score and number of scenic spots of the TOP-K Pareto optimal set produced by MOVNS and the routes generated by ATP. The experiments showed that MOVNS outperforms ATP. The value of K in this experiment was set to 10. As Pareto's optimal set can balance multiple objectives, it reached a higher score for routes and a higher number of scenic spots; see Figure 3.

5.3.4. Tourist Route Simulation Experiment

This experiment assumed the user's interest preferences and targets, and used ArcGIS to simulate the routes generated by ATP and MOVNS to compare the effectiveness of the recommendations. As shown in Figure 4, our proposed algorithm has better route planning and scenic spot recommendation quality when setting multiple objectives. The main findings are as follows. Firstly, MOVNS achieved shorter travel distance, and there was almost no bypassing. The route arrangement is reasonable, which shows that the dynamic distance score plays a very good regulating role. Secondly, the types of scenic spots recommended by MOVNS are much richer than those of ATP, which indicates that the idea of crowd sensing method can provide better decision support for tourists. Finally, under the multi-objective and multi-constraint settings, the route recommended by MOVNS is more in line with the real needs of users.



Figure 3. Comparison of TOP-K route recommendations. (a) Route Score. (b) Number of scenic spots.



Figure 4. Comparison of route simulation. (a) ATP route. (b) MOVNS route.

6. Conclusions

We proposed a multi-objective and multi-constrained travel route recommendation algorithm. It uses the idea of crowd sensing to get the social and location scores of scenic spots, and calculates the matching similarity between the user's labels and POI labels, and incorporates the user's personal space-time accessibility value into the above score. Based on the multi-constraints model, the MOVNS method includes the variable neighborhood search algorithm and a hybrid particle swarm genetic optimization algorithm, which is designed to recommend the optimal tour route to meet the user's personalized needs. The experimental results of the algorithm proposed in our paper on a real dataset of Beijing showed that it is better than the ATP algorithm in terms of scores for the recommended routes, interest abundance, and number of POIs in the route, so our proposed approach has a better recommendation effect. Moreover, our proposed algorithm is more efficient than ATP in running time. However, our proposed algorithm also has some problems, such as slow convergence and complex parameter settings. The explainability of recommendation results is very important for applications to recommender systems. Therefore, we plan to study how to construct the semantic model of tourism POIs by using a knowledge graph, and further construct the tour route recommendation method based on a knowledge graph, so as to enhance the loyalty and satisfaction of users.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z. and Y.L.; software, X.Z.; validation, X.Z., Y.L., J.Z. and S.C.; formal analysis, X.Z.; investigation, X.Z.; resources, X.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z., L.S., Q.Y. and Y.L.; visualization, X.Z.; supervision, J.Z. and Y.L.; project administration, Y.L. and X.Z.; funding acquisition, Y.L. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of China (61972439), the Natural Science Foundation of Anhui Province (1808085MF172), and the Key Program in the Youth Elite Support Plan in Universities of Anhui Province(gxyqZD2019010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available online from https://ieee-dataport.org/documents/beijing-poi-datasets-geographical-coordinates-and-ratings accessed on 9 October 2021.

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

References

- 1. Khan, M.U.S.; Khalid, O.; Huang, Y.; Ranjan, R.; Zhang, F.; Cao, J.; Veeravalli, B.; Khan, S.U.; Li, K.; Zomaya, A.Y. Macroserv: A route recommendation service for large-scale evacuations. *IEEE Trans. Serv. Comput.* **2017**, *10*, 589–602. [CrossRef]
- Sengvong, S.; Bai, Q. Persuasive public-friendly route recommendation with flexible rewards. In Proceedings of the IEEE International Conference on Agents (ICA), Beijing, China, 6–9 July 2017; pp. 109–114.
- Ji, S.; Wang, Z.; Li, T.; Zheng, Y. Spatio-temporal feature fusion for dynamic taxi route recommendation via deep reinforcement learning. *Knowl.-Based Syst.* 2020, 205, 106302. [CrossRef]
- 4. Chen, C.; Zhang, D.; Guo, B.; Ma, X.; Pan, G.; Wu, Z. Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1259–1273. [CrossRef]
- Hua, Y.; Cao, J.; Gu, Q.; Tan, Y. Pd-trp: A service composition approach for the personalized and optimized door-to-door travel plan recommendation. In Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017; pp. 768–775.
- Hu, G.; Qin, Y.; Shao, J. Personalized travel route recommendation from multi-source social media data. *Multimed. Tools Appl.* 2020, 79, 33365–33380. [CrossRef]
- Kurashima, T.; Iwata, T.; Irie, G.; Fujimura, K. Travel route recommendation using geotagged photos. *Knowl. Inf. Syst.* 2013, 16, 37–60. [CrossRef]
- Wang, J.; Wu, N.; Zhao, W.X.; Peng, F.; Lin, X. Empowering A* search algorithms with neural networks for personalized route recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 539–547.

- 9. Zuo, Y.; Gong, M.; Zeng, J.; Ma, L.; Jiao, L. Personalized recommendation based on evolutionary multi-objective optimization. *IEEE Comput. Intell. Mag.* **2015**, *16*,52–62. [CrossRef]
- 10. Wang, S.; Gong, M.; Li, H.; Yang, J. Multi-objective optimization for long tail recommendation. *Knowl.-Based Syst.* 2016, 104, 145–155. [CrossRef]
- 11. Zhou, Z.; Yang, M.; Sun, F.; Wang, Z.; Wang, B. A Continuous Transportation Network Design Problem with the Consideration of Road Congestion Charging. *Sustainability* **2021**, *13*, 7008. [CrossRef]
- Pop, M.D.; Prostean, O.; Prostean, G. Prediction of Route Choosing Behavior Based on Genetic Algorithm Approach. In Proceedings of the 2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES), Las Palmas de Gran Canaria, Spain, 21–23 June 2018; pp. 193–198.
- 13. Yu, Z.; Xu, H.; Yang, Z.; Guo, B. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Trans.-Hum.-Mach. Syst.* 2016, *46*, 151–158. [CrossRef]
- 14. Campigotto, P.; Rudloff, C.; Leodolter, M.; Bauer, D. Personalized and situation-aware multimodal route recommendations: The favour algorithm. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 92–102. [CrossRef]
- 15. Chiang, H.-S.; Huang, T.-C. User-adapted travel planning system for personalized schedule recommendation. *Inf. Fusion* 2015, 21,3–17. [CrossRef]
- Jaimes, L.G.; Vergara-Laurens, I.J.; Raij, A. A survey of incentive techniques for mobile crowd sensing. *IEEE Internet Things J.* 2015, 2, 370–380. [CrossRef]
- 17. Zhu, X.; Luo, Y.; Liu, A.; Tang, W.; Ma, Z. A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility. *IEEE Trans. Intell. Transp. Syst.* **2021**, 22, 4648–4659. [CrossRef]
- 18. Dasari, V.S.; Kantarci, B.; Pouryazdan, M.; Foschini, L.; Girolami, M. Game theory in mobile crowdsensing: A comprehensive survey. *Sensors* 2020, *20*, 2055. [CrossRef]
- Noor, R.M.; Rasyidi, N.B.G.; Nandy, T.; Kolandaisamy, R. Campus Shuttle Bus Route Optimization Using Machine Learning Predictive Analysis: A Case Study. Sustainability 2021, 13, 225. [CrossRef]
- 20. Liu, F.; Liu, C.; Zhao, Q.; He, C. A Hybrid Teaching-Learning-Based Optimization Algorithm for the Travel Route Optimization Problem alongside the Urban Railway Line. *Sustainability* **2021**, *13*, 1408. [CrossRef]
- 21. Pitakaso, R.; Sethanan, K.; Srijaroon, N. Modified differential evolution algorithms for multi-vehicle allocation and route optimization for employee transportation. *Eng. Optim.* **2020**, *52*, 1225–1243. [CrossRef]
- Boonsri, P.; Kaewkanha, T.; Namwiset, J.; Puntheeranurak, S. Pathly: The integrated system for finding diverse journey methods. In Proceedings of the 2016 Fifth ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, Thailand, 27–28 May 2016; pp. 117–120.
- Wang, J.; Chi, W.; Li, C.; Wang, C.; Meng, M.Q.H. Neural RRT*: Learning-based optimal path planning. *IEEE Trans. Autom. Sci.* Eng. 2020, 17,1748–1758.
- Yen, J.Y. Vergara-Laurens and A. Raij. Finding the k shortest loopless paths in a network. Manag. Sci. 1971, 17,712–716. [CrossRef]
- 24. Cao, X.; Chen, L.; Cong, G.; Xiao, X. Keyword-aware optimal route search. Proc. Vldb Endow. 2012, 5, 1136–1147. [CrossRef]
- 25. Luan, W.; Liu, G.; Jiang, C.; Zhou, M. Mptr: A maximal-marginal-relevance-based personalized trip recommendation method. *IEEE Trans. Intell. Transp. Syst.* 2018, 19, 3461–3474. [CrossRef]
- Rahimi, S.M.; Far, B.; Wang, X. Behavior-based location recommendation on location-based social networks. *GeoInformatica* 2020, 24, 477–504. [CrossRef]
- 27. Zhu, X.; Hao, R.; Chi, H.; Du, X. Personalized location recommendations with local feature awareness. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
- Zhu, X.; Hao, R.; Chi, H.; Du, X. Fineroute: Personalized and time-aware route recommendation based on check-ins. *IEEE Trans. Veh. Technol.* 2017, 66, 10461–10469. [CrossRef]
- 29. Kargar, M.; Lin, Z. A socially motivating and environmentally friendly tour recommendation framework for tourist groups. *Expert Syst. Appl.* **2021**, *180*, 115083. [CrossRef]
- Bao, S.; Yanagisawa, M.; Togawa, N. Personalized one-day travel with multi-nearby-landmark recommendation. In Proceedings
 of the 2017 IEEE 7th International Conference on Consumer Electronics, Berlin, Germany, 3–6 September 2017; pp. 239–242.
- Su, H.; Zheng, K.; Huang, J.; Jeung, H.; Chen, L.; Zhou, X. Crowdplanner: A crowd-based route recommendation system. In Proceedings of the 2014 IEEE 30th International Conference on Data Engineering, Chicago, IL, USA, 31 March–4 April 2014; pp. 1144–1155.
- 32. Luo, W.; Tan, H.; Chen, L.; Ni, L.M. Finding time period-based most frequent path in big trajectory data. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 22–27 June 2013; pp. 713–724.
- 33. Yao, Y.; Peng, Z.; Xiao, B. Parallel hyper-heuristic algorithm for multi-objective route planning in a smart city. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10307–10318. [CrossRef]
- Beijing POI Datasets with Geographical Coordinates and Ratings. IEEE Dataport. Available online: https://ieee-dataport.org/ documents/beijing-poi-datasets-geographical-coordinates-and-ratings (accessed on 29 October 2021).