# Itinerary Recommendation for User Groups in Temporary Social Network

Jing Xia[1,2], Yu Li[1,2], and Yuyu Yin[1,2](✉)

[1] School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China
[2] Key Laboratory of Complex Systems Modeling and Simulation of Ministry of Education, Hangzhou 310018, China
yinyuyu@hdu.edu.cn

**Abstract.** Temporary social network has been a increasing popular field in the last few years where people form a temporary social group for a short time period with common interests or purposes in the same area. When a user attends an event or conference in a new city, he/she can join the temporary social networks with his/her social account. Users who attend the same academic conference or activity may have similar interests and time schedules, so they are willing to travel together. Recently, renting cars to travel has become very common, since it helps improve user experience as well as save travel costs (e.g., renting and oil costs). Thus, we propose a group-wise itinerary planning framework to minimize the travel costs for each user in a temporary social network. Experimental results conducted on real-world data sets confirm the efficiency and effectiveness of our proposed framework.

**Keywords:** Travel planning · Spatial temporary social network · Mobile computing

## 1 Introduction

Temporary social network has been a increasing popular field in the last few years where people form a temporary social group for a short time period with common interests or purposes in the same area. When a user attends an event or conference in a new city, he/she can join the temporary social networks with his/her social account. They can send message to all group members, share locations and pictures, set up sub-groups, etc. All the records of her actions in the temporary social network will be deleted after the guest checks out of the hotel.

Supported by the National Key Research and Development Program of China (No. 2017YFB1400601), National Natural Science Foundation of China (No. 61802098), Natural Science Foundation of Zhejiang Province (No. LY16F020017), Shaanxi Province (No. 2018JQ6050), and Fundamental Research Funds for Central Universities (JBX171007).
The two authors Jing Xia and Yu Li contribute equally to this paper, and they are co-first authors.

Travel is a core function of temporary social networks. For instance, in a hotel social network, guests may attend the same conference in a new city, but they did not know each other before. Guests may have common interests and thus be willing to travel together with their new friends, thereby improving user satisfaction. More importantly, guests prefer to travel together to share travel costs (e.g., car-renting and oil costs). However, organizing guests into temporary social groups may negatively affect user experience, thus driving the users away from the application. Because, although the guests are coming to the same conference/business activity/concert, their time schedules and interest preferences may vary greatly.

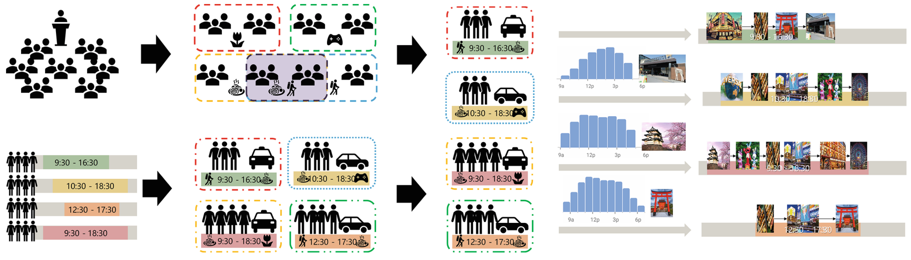


**Fig. 1.** Example of the framework

In this paper, we propose a framework for temporary social networks to group users and recommend group-wise itineraries. When a guest checks in at a commercial activity/academic conference, we will obtain his or her interest preferences and available time periods for traveling via his/her social network account. As showed in Fig. 1, we first group similar guests with similar hobbies, then we allocate users into car groups whose size are limited by the car capacity (e.g., four). Then, we recommend an itinerary to each group with crowdedness-aware. This itinerary will meet all group members' interest preferences and time constraints. To measure the similarity of two guests, we consider the overlapping of their available time slots and their common interest preferences.

Some research has been conducted to examine similar problems to those presented herein. Our problem formulation presents major differences from those current studies. For comparison, the differences between our system and the earlier works are illustrated in Table 1.

In this paper, we consider the deadline and interest preference of each user while grouping users, as well as the group size (car capacity). Then we also take the popularity and crowdedness of each POI into consideration during itinerary recommendation. In this way, we improve the quality of the generated groups and recommended itineraries. To our knowledge, no paper has studied this same problem.

This paper has the following four contributions.

- We propose a group-wise itinerary planning framework, which can be used to improve users' travel experience in a temporal social network.

**Table 1.** Difference between prior work and our system

| Paper | User's deadline | User's preference | POI's category | POI's popularity | POI's crowdedness | Group recommen-dation | Group size limit |
|---|---|---|---|---|---|---|---|
| [20] | ✓ | ✓ | × | × | × | × | × |
| [5] | × | ✓ | ✓ | × | × | ✓ | × |
| [13] | ✓ | ✓ | ✓ | ✓ | × | ✓ | × |
| [19] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × |
| [2] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × |
| [11] | ✓ | ✓ | ✓ | ✓ | ✓ | × | × |
| ours | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

- We design relevance measure functions to group users according to their preferences and available times.
- We combine multiple objectives and discuss different methods in itinerary recommendation that achieve a balance between conflicting objectives such as preferences of group-wise users, POI popularity and crowdedness.
- We evaluate the efficiency and effectiveness of our proposed framework and methods by extensive experiments using real road network data sets.

For the remaining paper: Sect. 2 formally formulates the preliminary definitions and propose our three-step framework. Section 3.1 trains the city model by cluster method. Section 3.2 presents a greedy-based user allocating algorithm. Section 4.2, we compare two algorithms to recommend itinerary avoiding crowdedness. Section 5 gives the experimental study. Section 6 discusses related work; and Sect. 7 concludes this paper.

## 2 Problem Statement

In this section, we first introduce the main symbols used in this paper and then formally formulate the framework proposed in this paper.

$U = \{u_1, \ldots, u_n\}$ be the set of users and $V = \{v_1, \ldots, v_l\}$ be the set of POIs. Given that $C = \{c_1, \ldots, c_o\}$ as POI category set, each POI $v$ belongs to one category. The popularity of POI $v$ is defined as $Pop(v)$.

We assume $Int_u(c)$ be the user's interest preference for category $c$. In our implementation, we estimate user interest preference using the total number of check-ins from user history data. The user interest preference of every user is defined as $\boldsymbol{I_u} = \langle Int_u(c_i), \ldots, Int_u(c_o) \rangle$.

We aim to address the problem in terms of its two sub-problems of user grouping and route recommendation. As illustrated in Fig. 2, our three-step framework includes offline city model train, car group allocation and itinerary recommendation with crowdedness. We use the cluster method train the different city travel pattern then part users into pattern groups and then each group query runs car group allocation method in parallel. For each small groups, we unified group members' interests and time schedules. Finally, we use route recommender calculate the itinerary for each group to satisfy the group's preferences and time constraints, at the same time, help users avoid crowdedness. The objective of *group-wise itinerary planning* is to recommend traveling partners and traveling routes for users in the temporal social network, such that: ($\mathcal{O}1$) all users' time constraints are satisfied, ($\mathcal{O}2$) users' preferences are satisfied, and ($\mathcal{O}3$) each user's traveling cost is minimized.

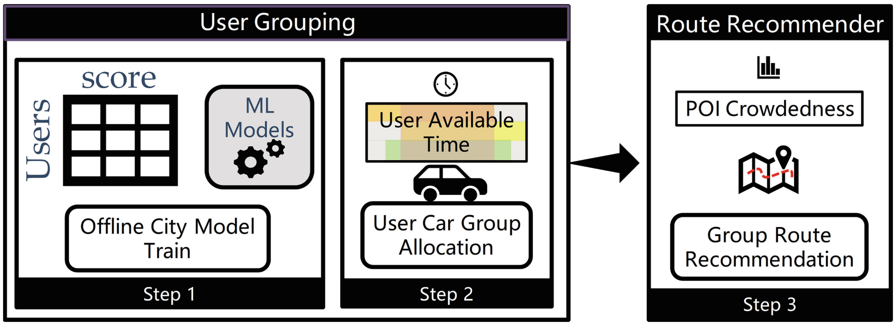


**Fig. 2.** Framework

## 3 User Grouping

In this section, we illustrate how to group users in a temporal social network. First, we train offline city model to find the distinguished interest patterns in the city so as to divide users into pattern groups according to these interest patterns. Then, we further divide each pattern group based on users' available time periods and car capacities. As the time complexity of car group allocation is quite high, we obtain the user groups by the greedy method at the end of this section.

### 3.1 Step 1: Offline City Model Train

According to the history travel itinerary, the kinds of different interest preferences among users are limited. A natural approach to group by interest preferences in a city is to use some clustering method. We train the offline city model by clustering history users interest preferences in the city to find all key interest patterns.

In our case, however, the number of clusters is not known in advance. In the following discussion, we assume there are $k$ interest patterns in a city, i.e., $P = \{p_1, \ldots, p_k\}$, where $p_i$ represents a city interest preference pattern. Each interest preference pattern $p_k$ consists of a cluster of users $p_k = \{u_1, \ldots, u_q\}$, and users in the same pattern have similar interest preferences. The $k$ patterns are as follows:

$$p_k = \frac{1}{|p_k|} \sum_{u \in p_k} I_u, \forall p \in P. \tag{1}$$

In this paper, we represent discrete travel patterns by Gaussian mixture model(GMM) [15] which is a set of continuous density functions. Effective clustering minimizes inter-cluster and maximizes intra-cluster similarities [3]. We use cosine similarity $Cos(u_i, u_j)$ of interest preference measure the interest preference similarity between two users.

Intra-cluster similarity is the average cosine similarity of all pair-wise combinations of users in a cluster $p$.

$$Intra(p_k) = \frac{1}{(|p_k| \cdot (|p_k| - 1))} \sum_{u_i \in p_k} \sum_{u_j \in p_k, u_j \neq u_i} Cos(u_i, u_j) \tag{2}$$

The inter-pattern similarity is the similarity between patterns, which is defined as follows:

$$Inter(p_i, p_j) = \frac{\boldsymbol{p_i} \cdot \boldsymbol{p_j}}{\|\boldsymbol{p_i}\| \cdot \|\boldsymbol{p_j}\|}, \forall p_i, p_j \in P \tag{3}$$

Formally, we find a best set of $k$ patterns $P$ such that:

$$Max \frac{\frac{1}{k}\sum_1^k Intra(p_k)}{\frac{1}{(k \cdot (k-1))} \sum_{p_i \in P} \sum_{p_j \in P, p_j \neq p_i} Inter(p_i, p_j)} \tag{4}$$

After we model those $k$ patterns, and part users' queries into $k$ pattern groups. In subsequent steps, users in the same pattern group can be allocated by their available time in parallel. In this way, we can reduce the server-side response time and improve system efficiency.

## 3.2 Step 2: Car Group Allocation

We allocate similar users into car groups, help them to save costs and find travel partners, even if they need to adjust their travel schedules slightly. We need to group users according to interest preferences and the overlap of their available time, as well as the group size which is limited by car capacity.

**User's query:** $u = \{I, l, u, v_s\}$, where $u.I$ is the user's interest preference. $u.t = [t_s, t_e]$ is the user's available time period. $u.v_s$ is the user's start position. We assign the users into $k$ pattern groups by comparing $u.I$ with each $k$ pattern center and then mark them with the label $u.l$.

**Time Similarity:** The time period similarity between users $u_i$ and $u_j$ calculated by the Jaccard similarity, which is also known as the intersection over the union.

$$TS(u_i, u_j) = \frac{u_i.t \cap u_j.t}{u_i.t \cup u_j.t} \tag{5}$$

**User Similarity:** Then, we define the similarity between $u_i, u_j$ using

$$Sim(u_i, u_j) = \mu TS(u_i, u_j) + (1 - \mu)\, Cos(u_i, u_j) \tag{6}$$

**Thresholds:** $DT$ be the maximum start/end time difference between two users, $TT$ be the lowest time overlap and $CT$ be the lowest interest similarity of group members based on Cosine.

**Occupancy Rate:** Car Capacity $CA$. The lowest car occupancy rate $OR$ means the group size is least $OR \times CA$ users in a group, otherwise renting a car is too wasteful.

**Car Group:** $\mathcal{G} = \{g_1, \ldots, g_m\}$ is the set of car groups, whose size $|g|$ is smaller than $CA$ and larger than $OR \times CA$. Each query is defined by $g = (I, t_s, t_e, v_s)$, where $g.I$ is the group's interest preference.

$$\boldsymbol{I_g} = \langle Int_g(c_1), \ldots, Int_g(c_o) \rangle, \forall \{c_1, \ldots, c_o\} \in C \tag{7}$$

$g.t_s$ is the group's start time which is the start time of the latest start user in the group. $g.t_e$ is group's end time which is the end time of the earliest end user in the group. $g.v_s$ is the group's start position. We assume users are in the same hotel.

**Group's Interest Preference:** One major challenge in the group-wise itinerary recommendation is the diverse interest preferences among group members. We define a collective group interest preference to meet all members demand. The group interest preference for category $c$ as follows:

$$Int_g(c) = \omega \cdot rel(u, c) + (1 - \omega) \cdot (a_{max} - dis(u, c)), \forall u \in g \tag{8}$$

where $a_{max}$ is maximum interest preference among group members for category $c$ and $rel(u, c)$ is the average preference score among group members for category $c$; $dis(u, c)$ is the pairwise interest preference difference of group members for category $c$.

**Problem Define:** The (1) car capacity $CA$, (2) occupancy rate $OR$, and (3) users' queries $U$ are given. Allocating users into different car groups whose size is limited by $CA$ and $OR$. Our goal is to group more people to travel together, and the maximum total average user similarity in each group includes people who should travel alone.

## 3.3 Method

Since the optimal solution requires the assessment of all possible combinations with very high complexity, we design a greedy strategy with following rules to allocate each pattern group users into small car groups. We use *queries* queue to receive more user queries.

**Rule 1.** If $u$ and $u'$ both send a query, and the start time of $u$ is earlier than that of $u'$, $u$ is serviced first. We sort the users' queries by start time and end time.

**Algorithm 1.** Car Group Allocation

```
Input: a level-1 group $g$, $CA$,$OR$,$DT$,$TT$,$CT$
Output: $\mathcal{G} = \{g_1, g_2, \cdots, g_m\}$
 1: $queries \leftarrow \emptyset$ , $\mathcal{G} \leftarrow \emptyset$, $single \leftarrow \emptyset$
 2: $used \leftarrow \emptyset$
 3: while $g \setminus used \neq \emptyset$ do
 4:    $u' \leftarrow$ first $u \in g \setminus used$, $candi \leftarrow$ empty $MaxHeap(CA - 1)$
 5:    for $u'' \in g \setminus \{used \cup u'\}$ do
 6:       if $isGroup(u', u'')$ then
 7:          $candi \leftarrow candi \cup u''$
 8:       end if
 9:    end for
10:    if $OR \times CA - 1 \leq |candi| \leq CA - 1$ then
11:       $g' \leftarrow u' \cup candi$, $\mathcal{G} \leftarrow \mathcal{G} \cup g'$, $used \leftarrow used \cup u' \cup candi$
12:    else
13:       $used \leftarrow used \cup u'$, $queries \leftarrow u'$
14:    end if
15: end while
16: $g \leftarrow queries$ go to line 2
17: for $u \in queries$ do
18:    $single \leftarrow single \cup u$
19: end for
20: return $\mathcal{G} \cup single$
```

**Lemma.** Suppose $u_1.t_s = 9:00, u_2.t_s = 9:10, u_3.t_s = 9:20, u_4.t_s = 9:30, u_5.t_s = 9:40$, and $u_1$ is the earliest user according to the start times among users. We find similar users to $u_1$ starting from 9:00, and then find similar users to $u_2$ starting from 9:10, excluding $u_1$ with a start time earlier than himself or herself. If $u_2$ can be grouped with $u_1$, after the previous identification of similar users to $u_1$, we will group $u_1$ with $u_2$. Then when we consider $u_2$, $u_1$ and $u_2$ are already in the same group, so there is no need to consider $u_2$ with $u_1$.

**Rule 2.** For $u$ and $u'$, we use function *isGroup*$(u, u')$ to validate if $u$ and $u'$ could travel together. *isGroup*$(u, u')$ returns *true* if $u$ and $u'$ satisfy $TT$, $CT$ and $DT$.

Furthermore, the number of similar users is too little to generate a group, and similar itineraries are recommended for them. Instead of querying *exact* itinerary recommendation for every single person, which may generate a lot of computational overhead, we send their queries as a batch and response a same recommendation.

## 4 Step 3: Itinerary Recommendation with Crowdedness

In this section, we use route recommender calculate the itinerary for each group to satisfy the group's preferences and time constraints, at the same time, help users avoid crowdedness.

## 4.1 POI Crowdedness

The crowdedness $Crd(v, t)$ of POI $v$ at time $t$ is the number of users visiting POI $v$ at time $t$ normalized to between zero and one. We can model the POI crowdedness in the future prediction as a time series forecasting problem. In our work, we use the average number of people per hour at each POI from data set as a prediction.

## 4.2 Itinerary Profit and Constraints

For each group $g$ with a category preference $g.I$, start time $g.t_s$, end time $g.t_e$ and start point $g.v_s$, we recommend an itinerary $R_g = \{v_1, v_2, \ldots, v_n\}$ which maximize the following object function:

$$Score(R_g) = \sum_{v \in R_g} pr_g(v) \tag{9}$$

where the $pr_g(v, t)$ evaluate the profit that gained when group members $g$ visit POI $v$(category $c$) at time $t$, which is defined as:

$$pr_g(v, t) = \frac{(\rho \cdot Pop(v) + (1 - \rho) \cdot I_g(c))^\gamma}{Crd(v, t)} \tag{10}$$

The itinerary constraints are as follows:

**Rule 1.** Each itinerary can visit the same POI only once, avoiding blind searches back and forth among POIs or rounding in the cycle, which is time-consuming and not beneficial.

**Rule 2.** The itinerary time cost $\sum_{v_i \in R} (Dur(v_i) + Dist(v_{i+1})) + Dist(v_s, v_1) + Dist(v_n, v_s)$ should not exceed $g.t_e - g.t_s$, where $Dur(v_i)$ is the average visit duration for a POI $v$. $Sat(v_i, v_j, t)$ returns *true* if leaving from $v_i$ at time $t$ to visit $v_j$ provides sufficient time to reach destination $v_s$ within the budget time.

## 4.3 Methods

Due to the irregular profit changes with time, then backtrack with pruning approach is to perform an exhaustive search which finds the optimal itinerary. To avoid the expensive backtrack with pruning search, the greedy method also proposed.

**Backtrack with Pruning.** We use depth-first backtrack to enumerate all the feasible arrangements of POI within time budget. And we use *visited* set to record the POI which is already visited in current partial itinerary. During the search we update the current best itinerary score.

**Algorithm 2.** Backtrack with Pruning

```
1: $R \leftarrow \emptyset$, $r' \leftarrow \langle v_s \rightarrow \emptyset \rightarrow v_s \rangle$, $visited \leftarrow \emptyset$
2: for $v_j \in V \setminus visited$ do                                    // Rule 1
3:     if $Sat(v_i, v_j, \pi_i)$ then                                      // Rule 2
4:         $r' \leftarrow r' \cup v_j$; $visited \leftarrow visited \cup v_j$;
5:         if $Score(r') > Score(R)$ then
6:             $R \leftarrow r'$;
7:         end if
8:         $backtrack(r', visited)$;
9:         $r' \leftarrow r' \setminus v_j$; $visited \leftarrow visited \setminus v_j$;
10:    end if
11: end for
12: return $R$
```

**Greedy.** Backtrack with Pruning search has too much time complexity, we use greedy method to select next POI $v_j$ iteratively appending to current partial itinerary until the time budget is not enough. We use a strategy function $f(v_j) = \frac{pr(v_j,t)}{Dist(v_i,v_j)+Dur(v_j)+Dist(v_j,v_s)}$ where $v_i$ is last visited POI and $t$ is the access time at $v_j$ to choose the next POI. The next POI should have maximal $f(v_j)$ and users can back to $v_s$ within budget.

## 5 Experiment and Evaluation

### 5.1 Experimental Settings

We evaluate our framework on a real-life datasets extracted from Flickr photos [18] in Toronto, Osaka, and Edinburgh, with the statistics shown in Table 2. All datasets are provided by Lim et al. [12,13,19].

We implement our framework and algorithms by Python sklearn packages and Java. All experiments are run on a 3.6 GHz Intel i7 Quad-Core and 16 GB of RAM PC.

**Table 2.** A summary of the datasets

| City | No. of POIs | No. of photos | No. of users | POI visits | Travel sequences |
|---|---|---|---|---|---|
| Toronto | 30 | 157,505 | 1,395 | 39,419 | 6,057 |
| Edinburgh | 29 | 82,060 | 1,454 | 33,944 | 5,028 |
| Osaka | 29 | 392,420 | 450 | 7,747 | 1,115 |

## 5.2 Effect of Offline City Model Train

We use users' geo-photos as the approximation as a user visit to each POI in real-life. First, we construct the interest preference vector $I$ for each user $u \in U$ by counting each user's photo number in different POI category and the POI popularity $Pop(v)$ by counting total photo number at different POI.

The best $k$ represents the goal in Eq. 3 is maximum when we use the GMM model to identify the best cluster partition based on historical data. The $Inter(P)$ line illustrates the average inter-cluster similarity of the $k$ pattern pairwise combinations. Table 3 shows the results. The intra-cluster similarity is very high, suggesting good cluster compactness. The inter-cluster similarity is low which shows the groups are not redundant. In this way, we get $k$ travel pattern in a city and people are parted to pattern groups with label $u_l$.

**Table 3.** Evaluation of Step 1

| | Toronto | Edinburgh | Osaka |
|---|---|---|---|
| best $k$ | 7 | 6 | 5 |
| $Intra(P)$ | 0.9172 | 0.9179 | 0.9109 |
| $Inter(P)$ | 0.1101 | 0.1357 | 0.1809 |
| $CT$ | 0.4203 | 0.5077 | 0.5547 |

**Table 4.** Average evaluation result of Step 2

| | Toronto | Edinburgh | Osaka |
|---|---|---|---|
| $Tu(\mathcal{G})$ | 0.9372 | 0.9439 | 0.9436 |
| $Cu(\mathcal{G})$ | 0.9699 | 0.9505 | 0.8889 |
| $Cos(\mathcal{G})$ | 0.9534 | 0.9590 | 0.9727 |
| single person | 8/1395 | 14/1454 | 10/450 |
| runtime(s) | 9.6000 | 12.4001 | 1.0024 |

## 5.3 Effect of Car Group Allocation

We use average cosine similarity ,time utilization, and car utilization of all generated groups to evaluate the effect of car group allocation.

**Time utilization:** $Tu(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{1}{|g|} \sum_{u \in g} \frac{g.t_e - g.t_s}{u.t_e - u.t_s}$

**Car utilization:** $Cu(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{|g|}{CA}$

Users' check-in timestamps are scattered in the datasets, and their travel are too short to simulate our problem. We generated users' start times $u.t_s$ and end times $u.t_e$ from 9:00 to 11:00 and 16:00 to 18:00 randomly. We set $Ca = 5$, $OR = 0.8$, $DT = 60$ min, $TT = 0.75$, $\mu = 0.5$, $\omega = 0.5$. We use the minimum intra-similarity in city models as $(CT)$.

Figure 3 shows the evaluation results for each small car group on each dataset. Table 4 shows the average evaluation results on each dataset. Our car group allocation algorithm is a greedy solution, the average time utilization $Tu(\mathcal{G})$ is greater than 90% on each dataset, indicating that the each user waits up to one hour (less than $DT$) for other users. Additionally, the average cosine similarity $Cos(\mathcal{G})$ is higher than the minimum lowest cosine similarity in pattern groups (Step 1). $Cu(\mathcal{G})$ and the single person shows our method group most people. In addition, the runtime on single-threaded shows the efficiency.

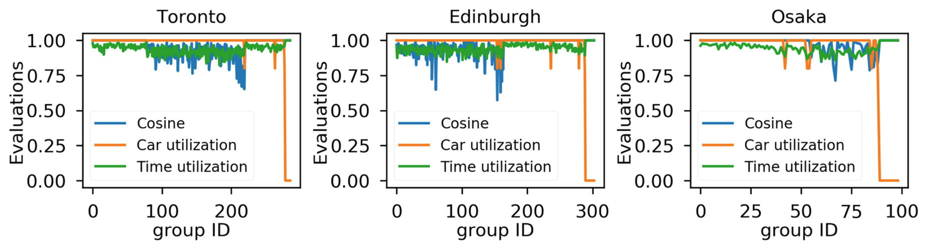


**Fig. 3.** Evaluation results of all groups in Step 2

## 5.4 Effect of Itinerary Recommendation

We use the time difference between the photos taken by the user for the first and last visit at a POI as user's visit duration and use the average time of all users as $Dur(v)$. For the cold spots, we replace the duration with the average duration of all POI in the city. We assume that all users' start locations (hotel $v_s$) are one of the most popular attractions in the city. We use euclidean distance estimate $Dis(v_i, v_j)$ between two POIs. If the distance between the two POI is less than 1 km, we use the walking speed (4 km/hour), consistent with literature [12]; otherwise the driving speed will be used. We estimate the time cost using Google Maps[1].

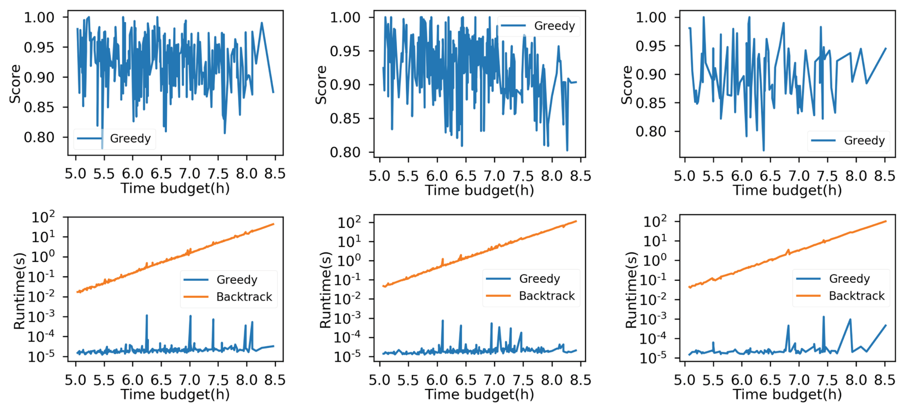
Figure 4 The upper row shows the recommended itinerary score by Backtrack and Greedy. The $y$-axis score of 1 represents the optimal result of the backtracking method. The line in the figure shows the ratio of the profit score obtained by the greedy method to the optimal solution. $x$-axis is the time budget(hours) of each group. As can be seen from the figure, the greedy method can achieve approximately 90% of the optimal solution.

The lower row is the runtime of the two algorithms. The $y$-axis is the run time in seconds. $x$-axis is the time budget(hours) of each group. Greedy has a fast and stable runtime. When the available POI is more than 29, the backtrack algorithm running for more than 300 seconds. However, backtracking is still suitable for finding the optimal itinerary in a large attraction or in a city with a small number of POI.

## 6 Related Work

**Group-wise Itinerary Planning.** Recently, the study of group-wise itinerary planning has gained growing attention. The problem is looking for a path that covers the set of user input points and minimizes the travel distance of the group. For instance, author in [1] analyzes the needs of group recommendation and proposes a formula that considers the correlation between individuals and the group, as well as the differences between members. [4,5] find the route with the maximum group interest preference and recommend it to multiple users. In [17], a

[1] https://www.google.com/maps.



(a) Toronto (b) Edinburgh (c) Osaka

**Fig. 4.** Effect of itinerary recommendation

novel group path query problem is proposed, in which users can dynamically join a new group or leave the original group while traveling. Lim et al. [13] clustered similar tourists into tour groups. Then recommend routes to each group. Finally, assigned an appropriate tour guide to each group.

**Route Recommendation.** There have also been numerous studies associated with recommending itineraries for a single tourist. [2,10] consider the orienteering problem with time windows(OPTW). These works are aimed at finding the best itinerary to maximize the user's experience under a traffic-conscious time budget constraint. And [11,20] consider the congestion of peak hours and queue time to choose the best time to access the POI. In addition, [6,8,9,16] provided other forms of search problems including optimal route for crowdsourcing works, finding a optimal access sequence. Those works calculate the shortest route passing through specific categories of POIs. Recent works [7,14] analyze interest preferences in user history data to recommend paths for them.

## 7 Conclusion

In this paper, we propose a group-wise itinerary planning framework which including offline city model train, car group allocation and itinerary recommendation with crowdedness. For improving happiness of the group members and reducing travel costs in the temporary social network, we design relevance measure functions to group users in terms of their interest preferences, time schedule and the group size limit. We use the cluster method to part users into pattern groups and then each group runs car group allocation in parallel. For each group, we design an itinerary score function which combines the group interest preference, the crowdedness and the popularity of each POI in order to achieves a balance between conflicting objectives. An recommended itinerary achieve maximum group satisfaction. The experiment based on a real-world social network data set shows the effectiveness of our framework.

## References

1. Amer-Yahia, S., Roy, S.B., Chawlat, A., Das, G., Yu, C.: Group recommendation: semantics and efficiency. Proc. VLDB Endowment **2**(1), 754–765 (2009)
2. 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. **16**(3), 1259–1273 (2015)
3. Chen, M.S., Han, J., Yu, P.S.: Data mining: an overview from a database perspective. IEEE Trans. Knowl. Data Eng. **8**(6), 866–883 (1996)
4. Fan, L., Bonomi, L., Shahabi, C., Xiong, L.: Multi-user itinerary planning for optimal group preference. In: Gertz, M., et al. (eds.) SSTD 2017. LNCS, vol. 10411, pp. 3–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64367-0_1
5. Fan, L., Bonomi, L., Shahabi, C., Xiong, L.: Optimal group route query: finding itinerary for group of users in spatial databases. GeoInformatica **22**(4), 845–867 (2018)
6. Khan, M.U.S., et al.: MacroServ: a route recommendation service for large-scale evacuations. IEEE Trans. Serv. Comput. pp. 589–602 (2017)
7. Kurashima, T., Iwata, T., Irie, G., Fujimura, K.: Travel route recommendation using geotags in photo sharing sites. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 579–588. ACM (2010)
8. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.H.: On trip planning queries in spatial databases. In: SSTD, pp. 273–290 (2005)
9. Li, W., Guan, J., Lian, X., Zhou, S., Cao, J.: Probabilistic time-constrained paths search over uncertain road networks. IEEE Trans. Serv. Comput. **11**(2), 399–414 (2018)
10. Li, Y., Yiu, M.L., Xu, W.: Oriented online route recommendation for spatial crowdsourcing task workers. In: Claramunt, C., et al. (eds.) SSTD 2015. LNCS, vol. 9239, pp. 137–156. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22363-6_8
11. Lim, K.H., Chan, J., Karunasekera, S., Leckie, C.: Personalized itinerary recommendation with queuing time awareness. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017, pp. 325–334 (2017)
12. Lim, K.H., Chan, J., Leckie, C., Karunasekera, S.: Personalized tour recommendation based on user interests and points of interest visit durations. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015, pp. 1778–1784 (2015)
13. Lim, K.H., Chan, J., Leckie, C., Karunasekera, S.: Towards next generation touring: Personalized group tours. In: Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, 12–17 June 2016, pp. 412–420 (2016)
14. Lu, X., Wang, C., Yang, J.M., Pang, Y., Zhang, L.: Photo2trip: generating travel routes from geo-tagged photos for trip planning. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 143–152. ACM (2010)
15. Nasrabadi, N.M.: Pattern recognition and machine learning. J. Electron. Imaging **16**(4), 049901 (2007)
16. Sharifzadeh, M., Kolahdouzan, M.R., Shahabi, C.: The optimal sequenced route query. VLDB J. **17**(4), 765–787 (2008)
17. Tabassum, A., Barua, S., Hashem, T., Chowdhury, T.: Dynamic group trip planning queries in spatial databases. In: Proceedings of the 29th International Conference on Scientific and Statistical Database Management. SSDBM 2017, pp. 38:1–38:6. ACM, New York (2017). https://doi.org/10.1145/3085504.3085584

18. Thomee, B., et al.: The new data and new challenges in multimedia research. CoRR abs/1503.01817 (2015)
19. Wang, X., Leckie, C., Chan, J., Lim, K.H., Vaithianathan, T.: Improving personalized trip recommendation by avoiding crowds. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, 24–28 October 2016, pp. 25–34 (2016)
20. Zhang, C., Liang, H., Wang, K., Sun, J.: Personalized trip recommendation with POI availability and uncertain traveling time. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, 19–23 October 2015, pp. 911–920 (2015)