International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Volume 5, Issue 7 - January 2017 - Pages 154-164

TRAVEL GUIDE FOR BACKPACK TRAVELLERS

Pavitra M Gadhar[1], Shweta M Nirmanik[2], Carmel Prabha[3]

[1,2,3]Department of Computer Science and Engineering R. T. E. Society's Rural Engineering College Hulkoti, Gadag, Karnataka, India-582205

Abstract--Travel agencies always provide some predefined itineraries which are not suitable for backpack traveler, as they are not meant for particular customer. Existing solutions provide automatic planning services in which the Points of Interest (POIs) are organized into customized itinerary. As the search space of all itineraries is too costly to explore entirely, assumptions are made that the trip is limited to only important POIs. In order to overcome the above limitation, in this project, we aim to design a planning service, where multiday itineraries are generated for the users. In this service, all POIs are given by users based on reviews. A two-stage planning scheme is proposed in which precomputing of single-day itineraries via MapReduce jobs are done in preprocessing stage and in second stage some approximation search algorithm is used for combining single day itineraries and finally entire itinerary for specified number of days is generated.

Keywords : *- Map reduce, itinerary planning, Points of Interest (POIs).*

1. INTRODUCTION

Traveling market is divided into two parts. For casual customers, they will pick a package from local travel agents. The package, in fact, represents a pre-generated itinerary. The agency will help the customer book the hotels, arrange the transportations, and preorder the tickets of museums/parks. It prevents the customers from constructing their personalized itineraries, which is very time consuming and inefficient. Although the travel agencies provide efficient and convenient services, for experienced travelers, the itineraries provided by the travel agents lack customization and cannot satisfy individual requirements. Some interested POIs are missing in the itineraries and the packages are too expensive for a backpack traveler.

Therefore, to attract more customers, travel agency should allow the users to customize their itineraries and still enjoy the same services as the predefined itineraries. However, it is impossible to list all possible itineraries for users. A practical solution is to provide an automatic itinerary planning service. The user lists a set of interested POIs and specifies the time and money budget. The itinerary planning service returns top-K trip plans satisfying the requirements. In the ideal case, the user selects one of the returned itineraries as his plan and notifies the agent. Planning an itinerary consumes more time. For a popular city or tourist spot, it involves investigating the many number of Points of Interest (POIs), selection of the POIs that one is interested, to work out the order in which POIs are required to be visited, and to safeguard

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

that the time it takes to visit each of them, and to make movement from one POI to other POI, satisfying the user's time budget [1].

Therefore, to attract more customers, travel agency should allow the users to customize their itineraries and still enjoy the same services as the predefined itineraries. However, it is impossible to list all possible itineraries for users. A practical solution is to provide an automatic itinerary planning service. The user lists a set of interested POIs and specifies the time and money budget. The itinerary planning service returns top-K trip plans satisfying the requirements. In the ideal case, the user selects one of the returned itineraries as his plan and notifies the agent.

2. METHODOLOGY

First, current planning algorithms only consider a single day's trip, while in real cases, most users will schedule an n-day itinerary (e.g., the one shown in Fig. 1). Generating an n-day itinerary is more complex than generating a single day one. It is not equal to constructing n single-day itineraries and combining them together, as POI can only appear once in the itinerary. It is tricky to group POIs into different days. One possible solution is to exploit the relocations, for example, nearby POIs are put in the same day's itinerary. Alternatively, we can also rank POIs by their importance and use a priority queue to schedule the trip.

Second, the travel agents tend to favor the popular POIs. Even for a city with a large number of POIs, the travel agents always provide the same set of trip plans, composed with top POIs. However, those popular POIs may not be attractive for the users, who have visited the city for several times or have limited time budget. It is impossible for a user to get his personal trip plan. The travel agent's service cannot cover the whole POI set, leading to few choices for the users. In our algorithm, we adopt a different approach by giving high priorities to the selected POIs and generating a customized trip plan on the fly.

Third, suppose we have N available POIs and there are m POIs in each single day's itinerary averagely. We will end up with candidate itineraries. It is costly to evaluate the benefit of every itinerary and select the optimal one.

Therefore, in some heuristic approaches are adopted to simplify the computation. However, the heuristic approaches are based on some assumptions (e.g., popular POIs are selected with a higher probability). They only provide limited number of itineraries and are not optimized for the backpack traveler, who plans to have a unique journey with his own customized itinerary.

Last but not the least, handling new emerging POIs was tricky in previous approaches. The model needs to be rebuilt to evaluate the benefit of including the new POIs into the itinerary. For systems based on the user's feedback, we need to collect the comments for the new POIs from the users, which is very time-consuming.

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

To address the above problems, a novel itinerary planning approach is proposed. The design philosophy of our approach is to generate itineraries that narrow the gap between the agents and travelers.

Preprocessing:

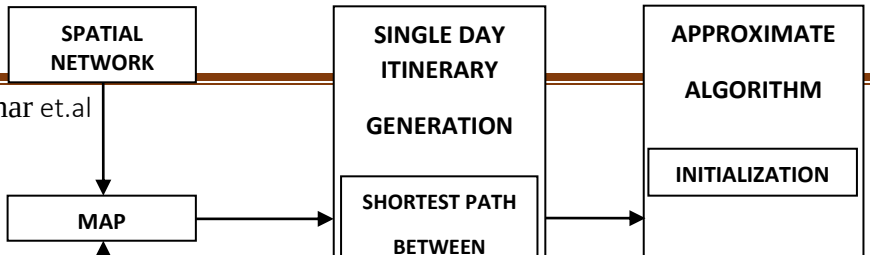
In the preprocessing, POIs are organized into an undirected graph. The distance of two POIs is evaluated by Google Map's APIs.1 Given a request, the system provides interfaces for the user to select preferred POIs explicitly, while the rest POIs are assumed to be the optional POIs. Different ranking functions are applied to different types of POIs. The automatic itinerary planning service needs to return an itinerary with the highest ranking. Searching the optimal itinerary can be transformed into the team orienteering problem (TOP), which is an NP-complete problem without polynomial approximations. Therefore, a two stage scheme is applied.

In the preprocessing stage, we iterate all candidate single-day itineraries using a parallel processing framework, MapReduce. The results are maintained in the distributed file system (DFS) and an inverted index is built for efficient itinerary retrieval. To construct a multiday itinerary, we need to selectively combine the single itineraries. The preprocessing stage, in fact, transforms the TOP into a set-packing problem, which has well-known approximated algorithms.

Approximation Algorithms:

In the online stage, we design approximate algorithms to generate the optimal itineraries. The approximate algorithm adopts the initialization-adjustment model and a theoretic bound is given for the quality of the approximate result. To evaluate the proposed approach, we use the real data from Yahoo Travel2. The experiments show that our approach can efficiently return high-quality customized itineraries. The remainder of this methodology is organized as follows: we formalize the problem and give an overview of our approach. Then, present the preprocessing stage and online stage of our approach, respectively.

3. SYSTEM DESIGN



International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
Volume 5, Issue 7 - January 2017 - Pages 154-164
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Fig3.1 Architecture of a trip planning system

Fig. 3.1 shows the proposed architecture for trip-planning system. A two stage scheme is proposed.

1) Pre-Processing
2) Algorithm approximation

Pre-processing stage:

In this stage candidate single-day itineraries are iterated using a MapReduce. The results of this are stored in the distributed file system.

Algorithm Approximation stage:

In this stage algorithms are designed which approximates algorithm that generate the optimal itineraries. These approximation algorithms take on initialization-adjustment model so that new solutions are searched.

3.2 POI Graph

3.2.1 Definition of POI Graph:

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

In the POI graph G= (V, E), for each of the POI a vertex is created and each pair of vertices are connected through undirected edge in E. In the graph G, every vertex and edge has the following properties:

1. "$\forall vi \in$ V, w (vi) denotes the weight (importance) of the POI and t (vi), is the average time that tourists will spend on the POI.

2. $\forall$ ($ex = vi \rightarrow vj$) $\in$ E, t (ex) is the cost of the edge, computed as the average travelling time fromvi to vj ."

In this planning system, the user chooses an interested set of POIs and requests the system to generate k-day itinerary. As shown in figure 4.1, the first thing to be carried out is the POI graph construction. Once the user provides the POIs, the graph is constructed which is shown in below figure.3.2 and the Google Map API is utilized to calculate the distance between two different POIs in the graph.

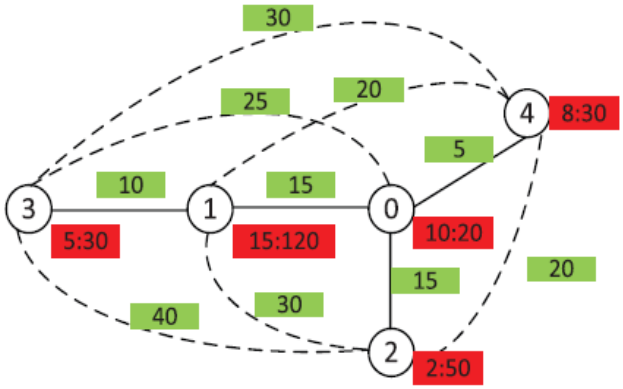


Fig3.2 POI graph

In the figure 3.2 shown above each node is considered as POI and bears two properties: weight and travel time. All nodes are connected through weighted edges. There are two types of edges. First is one which shows that in the map two nodes are directly connected (other POI is not existing in their shortest path e.g., path from 0 to 1). Second is which contains no direct path but has many shortest paths in the map. The cost of the edges are assumed to be symmetric in the definition of POI graph i.e. the travelling time from vi to vj are equal to the time from vj to vi , but in this approach the case of non symmetric is directly applied e.g., traffics varies for vi to vj and vj to vi .

Let w (v_i) represent the weight or importance of POIv_i . The initial weight of v_i is considered from the user"s reviews i.e. user can define score 0 to 5 for each POIs. User selectsset of POIs, and the weight of the POIs chosen are increased purposely by$\alpha(w(v_i)$ + i), where α is set to arbitrary integer. A single-day itinerary is as shown below

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

$L=v_0 \rightarrow \cdots \rightarrow v_n \rightarrow h_j$ -------- (1)

hj is hotel POI. The time is calculated as

$$t(L) = \sum_{i=0}^{n} t(vi) + \sum_{i=0}^{n-1} t(vi \rightarrow vi+1) + t(vn \rightarrow hj) \text{----------(2)}$$

Pre-Processing

Once POI graph is constructed next stage is pre-processing which involves two stages.

1. Single-day Itinerary Generation
2. Building Index

In first stage a set of MapReduce jobs are given way to construct all feasible or viable single-day itineraries. By making use of parallel processing mechanism such MapReduce all itineraries can be generated efficiently. In the second stage an itinerary index is built and these indexes are just reorganization of all single-day itineraries, which assists in itinerary search to be systematic or well structured.

Single-day Itinerary

The basic plan or scheme is iteration of all single-day itineraries, which is done using a MapReduce jobs. In each itinerary initially consists of only one POI. MapReduce job makes an effort to add one extra POI to the itinerary and checks whether the two POIs can be visited in the same day and the same process continues. If no single-day itineraries can be generated then the process halts. This grand design is based on conclusion that user cannot visit many number of POIs in single-day. The algorithms for MapReduce used here are mapper and reducer. The mapper tries to attach new POI to the existing itineraries. A test is conducted for each new pathto test that a path can be completed within a single day, if cannot be completed then the new pathis dropped out or discarded. And if the old path does not produce any new path, then the old pathis considered as a output. In mapper, in order to calculate the cost and weight of latest itinerary POI graph is loaded. POI graph is in the form of table and table's plot is as follows

(S_POI, E_POI, S_weight, E_weight, S_cost, E_cost, cost) -------- (3)

Where E_POI and S_POI represent the two POIs connected by a particular edge, cost is the travelling cost from E_POI to S_POI. Once all itineraries are generation is completed, a clean process is called to eliminate the duplicates.

Itinerary Index

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

The index is built using MapReduce jobs. In the algorithm mapper, a key-value pairs are generated for each POI that involved and the algorithm reducer on the other hand accumulates all itineraries of a particular POI and classifies or categorizes POI on the basis of their weights before index file is created. In real case size of the index file may differ as some of the POI may have a exceptionally huge index file, owing to its short visit time and popularity.

Approximation Algorithm

Once the itinerary indexes are built, the user request can be dealt with by choosing k best itineraries from the indexes. There are three processes

1 Initialization

2 Adjustment

3 Hotel Selection

Initialization

Whenever the user requests, the weight of POIs are adjusted in that set in order to highlight the user's selection. The weight of POI is increased by $\alpha(w(v_i) + 1)$, where α is integer which greater than 0 and $w(v_i)$ is original or actual weight of POI v_i.

Adjustment

In order to improve the weights of the itineraries that are obtained, there is adaptation of the adjustment phase. In this phase there is hunting of new solution and this process is iterated until no betterment can be procured.

Hotel Selection

Hotel selection process can be taken into account as the special type of POIs. This step appears as the last stage in the itinerary planning system. Depending on user's choice there are two stages of processing: multiple hotels, single hotel.

Multiple Hotels

If the user is not interested to stay in the same hotel, then the pre-processing algorithm can be extended to manage the hotel selection. In the MapReduce jobs, a test is conducted to test each hotel and tried to attach it to the terminating part of L_i. $L_i|h_j$ is reviewed as a single-day itinerary, if

1. "The complete travelling time of $L_i|h_j$, is less than H. H is the average travelling time per day.

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

2. For any other nonhotel POI $\bar{v}$ which is not considered by L_i, $L_i|\bar{v}|h_j$ cannot be completed within H time."

4. IMPLIMENTATION

The figure 3.1 shows the proposed architecture for trip-planning system. A two stagescheme is proposed.

1. Pre-Processing
2. Algorithm approximation

POI Graph

In this planning system, the user chooses interested set of POIs and requests the system to generate k-day itinerary. The first thing to be carried out is the POI graph construction. Once the user provides the POIs the graph is constructed and the graph is stored in the form of table, the table scheme is as shown below

(S_POI, E_POI, S_weight, E_weight, S_cost, E_cost, cost),

Where E_POI and S_POI represent the two POIs connected by a particular edge, cost is the travelling cost from E_POI to S_POI. The Google Map API is utilized to calculate the distance between two different POIs in the graph. The POI graph is read by function as shown below:

readPOIGraph (String fname)

Pre-Processing

Once POI graph is constructed next stage is pre-processing where the input to this stageis POI table and the single-day itineraries are generated using MapReduce method. The algorithm used for MapReduce is mapper and reduce. The mapper and reduce are invoked by functions as shown below respectively:

map (Object key, Text value, Context context)
reduce (Key key, Iterable values, Context context)

In mapper in order to calculate the cost and weight of latest itinerary POI graph which is in the form of table is loaded and the computation begins for generation of single-day itineraries. Once all itinerary generations are completed, a clean process is called to eliminate the duplicate POIs and the function is as shown below:

Path removeDuplicate (Path x, Vector<Path> rev)

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

Initialization

The initialization process is invoked by

Initialization (POIList L, Day k).

In this step grouping is done and before the grouping process, the POIs are sorted first dependingon their weights by

sortByWeight(L)

In this grouping process, it holds the subsets of POIs that can be visited within a day.

```
if i < k and L.size() > 0 then
poi= L.nextPOI();
Set group=new Set()
group.add (poi)
 lastpoi= poi
```

The POI with the shortest distance is greedily selected and is added into the group and this isinvoked by:

```
newpoi =getNearest  (lastpoi,  L)
time= getTravelTime (group, newpoi)
if time ≤ one day then
group.add(newpoi)
L.remove(newpoi)
Lastpoi= newpoi
```

Adjustment

The adjustment process is invoked by:

Adjustment (Set S, double P, int step)

Hotel Selection

In some cases user may prefer to stay in different hotel or in same hotel. The hotel selectionprocess is invoked by:

HotelSelection (Set hotels, Set itinerarySet).

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 5, Issue 7 - January 2017 - Pages 154-164

The main idea is at the end of each itinerary dropout some POI and tries to attach the hotel POI,as shown below

if getTravelTime (L_j , h_i) > H then

L_j .removelast ()

The above mentioned is a pseudo code to continuously remove the last POI.

5. CONCLUSION

In this project, we present an automatically itinerary generation service for the backpack travelers. The service creates a customized multiday itinerary based on the user's preference. This problem is a famous NP-complete problem, team orienting problem, which has no polynomial time approximate algorithm. To search for the optimal solution, a two-stage scheme is adopted. In the preprocessing stage, we iterate and index the candidate single-day itineraries using the MapReduce framework. The parallel processing engine allows us to scan the whole dataset and index as many itineraries as possible. After the preprocessing stage, the TOP is transformed into the weighted set-packing problem, which has efficient approximate algorithms. In the next stage, we simulate the approximate algorithm for the set-packing problem. The algorithm follows the initialization-adjustment model and can generate a result, which is at most worse than the optimal result. Experiments on real data set from travelling website show the proposed approach can efficiently generate high-quality customized itineraries.

REFERENCES

[1] Gang chen, Sai Wu, Jingbo Zhou, and Anthony K.H Tung "Atomatic Itinerary Planning for Travelling Services" IEEE Knoweledge And Data Engineeringvol.26 Mar 2014.

[2] Tom white., "Hadoop: The Definitive guide".

[3] S Singh., N Singh, "Big Data Analytics", International Conference on Communication, Information & Computing Technology (ICCICT), pp.208-214, 2012

[4] Apache Hadoop, 2013, http://hadoop.apache.org

[5] S.B. Roy, G. Das, S. Amer-Yahia, and C. Yu, "Interactive Itinerary Planning," Proc. IEEE 27th Int"l Conf. Data Eng. (ICDE), pp. 15-26,2011.

[6] C. Archetti, A. Hertz, and M.G. Speranza, "Metaheuristics for the Team Orienteering Problem," J. Heuristics, vol. 13, pp. 49-76, Feb. 2007.