For office use only	Team Control Number	For office use only
T1 ____________	**72973**	F1 ____________
T2 ____________		F2 ____________
T3 ____________	Problem Chosen	F3 ____________
T4 ____________	**B**	F4 ____________

2018
MCM/ICM
Summary Sheet

PTPs: E-Guide Helping Plan a Tour

Summary

When travelling in a new city, it is important but challenging to choose Points of Interests (POIs) to visit and design the route. The quality of a tour is highly dependent on tour planning. However, planning a tour takes much time and effort.

To free the tourist from tour planning, our paper provides a personalized tour planning system (PTPs), which can offer the user a tour plan based on his/her preference, budget and time constraint. Moreover, we revise our PTPs to deal with “super POIs”, namely, ones whose scale and structure cannot be omitted.

We first set up a criterion to characterize each POI’s type and popularity. Then we build our PTPs model and examine its effectiveness by applying it to New York City (NYC).

PTPs consists of three sub-models: Evaluation Model, Route Planning Model, and Hotel & Restaurant Recommendation Model. There is a figure showing the connection between all parts.

Evaluation Model gives each POI a “satisfactory score”. It measures the closeness between the type of a POI and the user’s preference, meanwhile considers the POI’s popularity, and produce a score to describe how satisfactory a POI is for the user.

Route Planning Model generates an optimal touring route under the constraints of time and budget. We apply a genetic algorithm to find a route with highest total satisfactory score.

Hotel & Restaurant Recommendation Model

To acquire the data we need,

Keywords: tour planning, POI selection, route design, tourist satisfaction

Contents

1 Introduction

1.1 Background

Travelling to another city is a delightful experience for most people. There are so many places of interest to visit, such as museums, parks and so on. With limited time, tourists have to choose among those places and design their own route. Therefore, tour planning is a important task for tourists. However, it takes a lot of time and massive information to make a perfect plan, and several constraints(e.g. cost, transportation time, etc.) should be considered.

Figure 1: View of a city.

1.2 Restatement of problem

As required by the question, we are supposed to determine a tour route among Points of Interests (POIs), which include spots, hotels and restaurants. The route we provide should cover the popular POIs, meanwhile cater for the user's personal preference, namely, maximize the user's satisfaction[1], and meet the constraints.

Furthermore, for "super POIs" (those with large scales) we should consider their inner structures, positions of entrances and exits and give a more subtle tour plan.

1.3 Literature review

Existing study on tourism products recommendation applies one of the following four methods[2]:

- **Content-based recommendation,** which recommends tourism products similar to products chosen by the user.

- **Collaborative filtering recommendation,** which recommends the choice of users with similar preference.
- **Knowledge-based recommendation,** including constraint-based[3] and case-based, makes recommendation according to Knowledge-based rules.
- **Social-based recommendation,** which applies users' relationship in social media.

Besides, in other works, factors such as distance[4], user interests[5], and popularity[6] are taken into consideration. However, some of these methods are dependent on a large amount of historical datas, therefore have difficulties in cold-start. Moreover, datas on travelling are sparse, and getting these datas from users might be viewed as an invasion of privacy.

There is also study on travelling route planning[7]. The majority of current works need users to input the origin and destination of travelling, and cannot deal with accommodation arrangement, which limits their application. Therefore, we combine spot recommendation and route planning in our model, and get rid of reliance on historical datas.

2 Assumptions

To simplify the real life situation, we will make the following assumptions at the start of constructing our models.

- **Sightseeing is in higher priorty than accommodation and meals.** Meals can spread all over the world, but spots can't. Therefore, the most significant embodiment of cities' characteristic are their spots. As a result, we consider sightseeing as the most important activity during a trip.
- **The popularity of POIs can totally reflects their quality.** Thus we only use the popularity of POIs and their conformity with tourists' preference to mearsure their Satisfactory Score.
- **Entrance tickets and traffic fares are ignorable compared to expenses of hotels and restaurants.** Entrance tickets and traffic fares are usually very cheap in comparison with expenses of hotels and restaurants. We view the cost of visiting a spot and transportation between POIs as zero.
- **Road distance between any two POIs are proportional to their great-circle distance.** Calculating accurate road distance requires extremely large quantity of data, and is unnecessary for tour planning. Therefore, we ignore specific road distribution in cities.
- **Speeds of each transportation methods are identical in every roads in a city repectively.** We ignore specific traffic volume distribution of time and roads. Because they also require large quantity of data and are not important for tour planning.
- **POIs in the scale of cities and subspots in the scale of spots are points without structures and volumes.**

3 Nomenclature

Notation	Definition	Unit
$\overrightarrow{cat}$	Vector that describes a spot's category	
$\overrightarrow{prf}$	Vector that describes the user's preference of spots' category	
$Intr$	Rate of user's interest to a certain spot	
Pop	Rate of POI's popularity among the public	
Sat	Score of satisfaction of a spot	
t_{dur}	Time duration of touring in a spot	hour
t_{tra}	Time duration of traffic between two spots	hour

4 Model I: Personalized Tour Planning System (PTPs)

In this section, we will discuss our models. Before diving into details, briefly explain the whole idea (as figure 2 shows). Firstly, we give each spot a score based on its popularity and conformity with user's preference. Then we construct a graph, containing scores, traffic time and tour duration. We apply a genetic algorithm to find a way maximizing the score under the constraints of time. Furthermore, we analysis the relation between final satisfactory score and some properties.

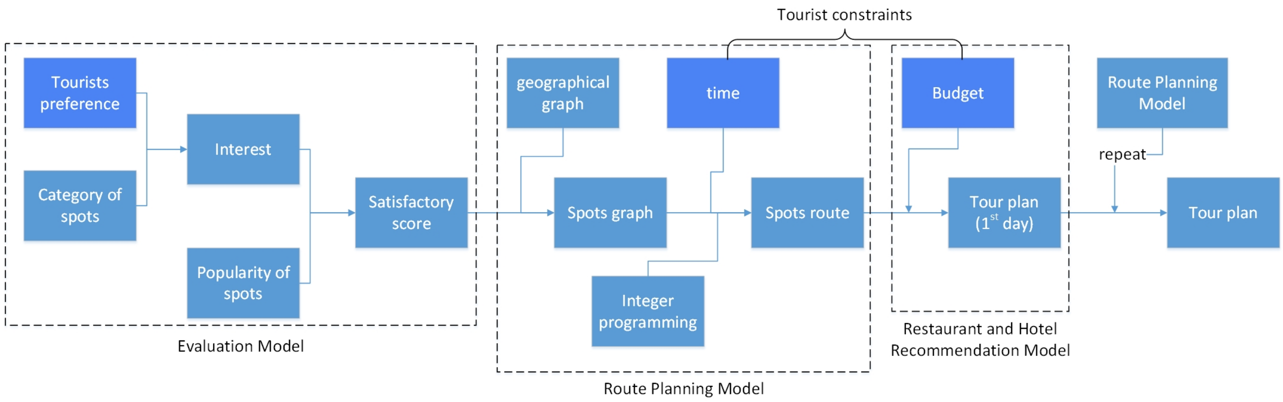

Figure 2: Overview of Personalized Tour Planning System

4.1 Rate the POIs: Evaluation Model

We define a series of dimensions to classify the POIs(e.g. landmark, museum, historical site, etc.) A spot is assigned a value ranging from 0 to 1 for each dimension. Thus we build a category vector $\overrightarrow{cat} = [r_1, r_2, \ldots, r_n]$, where r_i denotes the rate of i^{th} category, to classify each spot. Through a user inerface, the user is asked to input the following values:

- His/her "interest rate" to each category of POI;
- Total travelling time limit;
- Highest budget acceptable.

We normalize the data, thus getting the user's preference vector $\overrightarrow{prf}$, which is a unit vector. Then the rate of user's interest to a certain spot() can be calculated:

$$Intr(s1) = cos < \overrightarrow{cat}(s1), \overrightarrow{prf} >$$

To quantitify the each POI's popularity Pop, we acquire the datas of search volume and rating. We normalize the datas using a linear mapping, in which the maximum and minimum are converted to 1 and 0, respectively. Then we calculate the POI's satisfactory score by the following fomula:

$$Sat = \alpha Intr + (1-\alpha)Pop$$

where α is a positive weighting coefficient.

4.2 Construction of the graph

Data acquirement

After giving each POI a score, we calculate the distance between each two POIs by their latitudes and longitudes using Haversine fomula:

$$hav(\frac{d}{r}) = hav(\varphi_2 - \varphi_1) + cos(\varphi_1)cos(\varphi_2)hav(\lambda_2 - \lambda_1)$$

where:

- hav is the haversine function: $hav(\theta) = sin^2(\frac{\theta}{2})$;
- d is the distance between two spots along the great circle of the sphere;
- r is the radius of the sphere, namely the radius of earth in this case;
- φ and λ represent the latitude and longitude respectively.

We estimate the speed of each kind of transportation, and with the distance between spots d assigned, we can calculate the traffic time taken between each two spots.

Construct the graph With the datas prepared, we construct a weighted directed graph (figure 3), of which the nodes are spots. Every node is weighted by its score, and has touring duration as another property. The edge, which denotes road between the two spots it connects, is weighted by the traffic time.

4.3 Plan the Tour Route

With all the spots rated and a weighted graph constructed, we apply a non-linear programming to design the route. Since the exact optimum is hard to reach, we use genetic algorithm (as figure 4 shows) with non-linear constraints to generate a near-optimal route:

$$\max \sum_{i=1}^{n} Sat(s_i)\ (fitnessfunction) \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} t_{dur}(s_i) < t_{max} \tag{2}$$

$$s_i\ is\ only\ toured\ one\ time, (i = 1, 2, \dots, n) \tag{3}$$

where:

- s_i is the i^{th} spot, $i = 1, 2, \dots, n$, n is the total amount of spots;
- t_{max} is the longest total travelling time duration that the tourist can accept.

we use path coding method, represent a path starting from spot s_{k0} via $s_{k1}, s_{k2}, \dots, s_{km}$ with $k0 - k1 - \cdots - km$. Applying the *ga* funtion in MATLAB, we find out a near-optimal route and the corresponding value of fitness function.

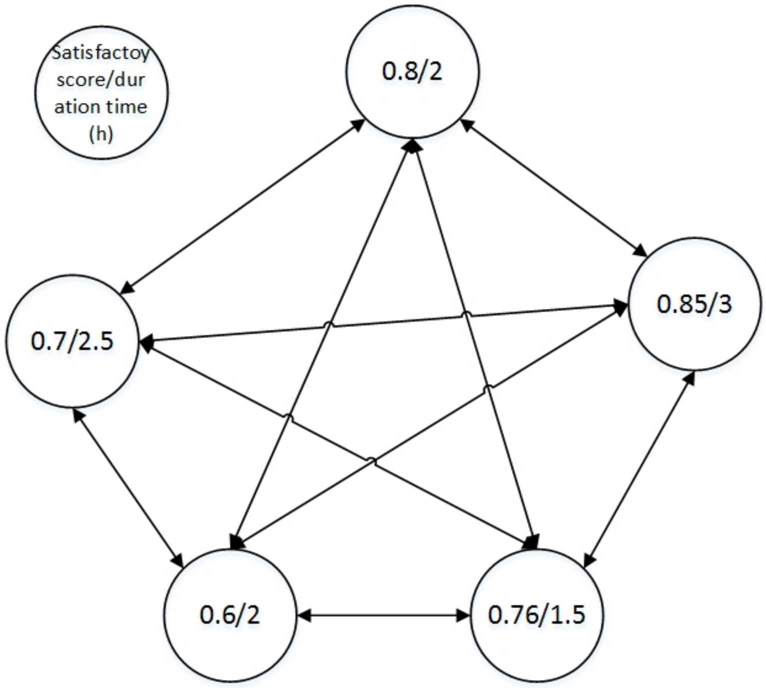

Figure 3: Structure of Weighted Graph Model

4.4 Restaurant & Hotel Recommendation

After providing a route to travel around spots, we work on ways to recommend hotels and restaurant to consummate our tour planning system. Since hotels and restaurants are different in many ways, we use different methods in recommendation.

Restaurants

From the route planning process, a travelling sequence $\mathscr{S} = [s_1, s_2, \ldots, s_n]$ is generated. We can also generate a sequence of starting times and ending times of touring the spots, $\mathscr{T} = [t_1, t_2, \ldots, t_{2n}]$. If we set a meal time(e.g. 12am for lunch and 6pm for supper), we can find a t_m in $\mathscr{T}$ that is closest to the meal time, and find the corresponding s_k of t_m.

Then we seek for restaurants that suits the constraints in a circle with s_k as center and a radius of 500 meters (might be adjusted due to the user's preference). With the similar method we apply in rating the spots, we give each restaurant a score according to its category, popularity and the user's preference. The one with highest score is what we recommend to the user.

Hotels

As stated in 2, the tourist stay in the same hotel during the whole tour. In our model, we first list the spots in the first day's schedule, and select from the hotels economically suitable in a 500m-radius-circle around each spot. The hotel with best popularity among those selected ones will be our recommendation.

5 Model II: PTPs with Super POIs

In 4, we regard the POIs as points without shape, inner structure and volume. For the Super POIs, inner information should be take into consideration, therefore we revise our model (figure 5). To begin with, we evaluate the POIs and generate a optimal route using PTPs model. Then we

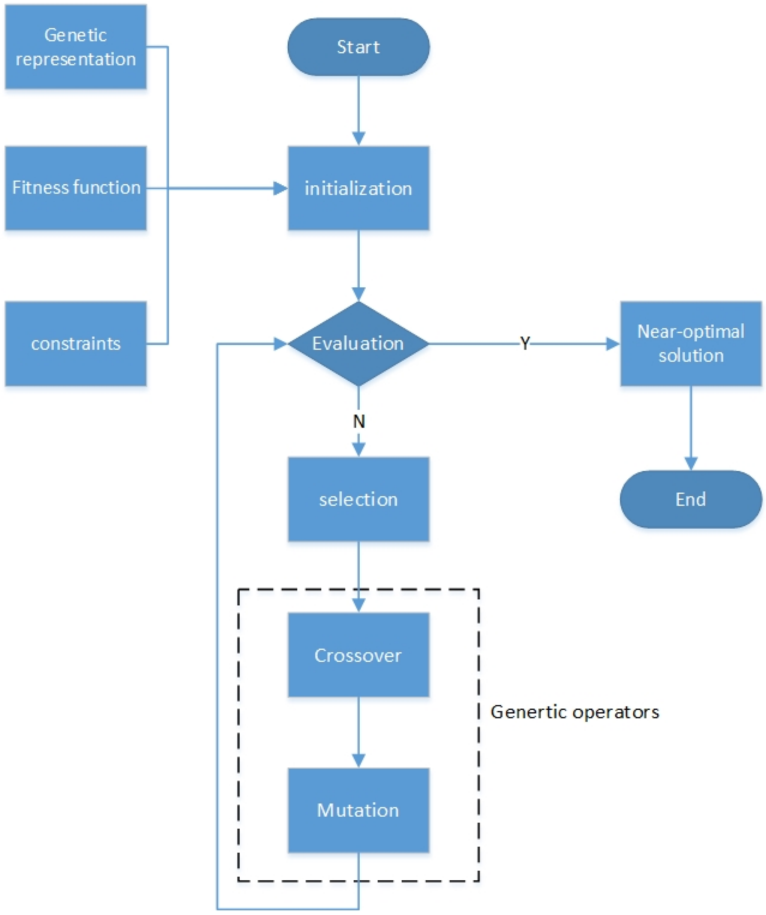

Figure 4: Sketch of genetic algorithm

transform each super POI to a weighted graph, in which the nodes are gates and subspots to view. Each subspot has a tour duration as its property.

In the tour route generated, each super POI has two adjacent nodes (two spots). We select two gates with least distance to the adjacent nodes and decide the entrance and exit according to the tour direction. After the selection of gates is done, we design the tour route inside the super POI. Using the evaluation model we give each subspot a satisfactory score. And because the traffic time between subspots is ignorable, we just need to maximize the total satisfactory score under the constraint of time.

To choose among all subspots in a super POI, we apply a greedy algorithm: Pick the highest-scored subspot, and then pick the highest-scored one from the left ones $\cdots$ until total tour duration reaches the time limit. Planning a route to tour around the chosen subspots starting from the entrance and ending at the exit is a travelling salesman problem (TSP).

6 Inplementation

To test our PTPs model, we first apply it to New York City (NYC). Using the information we gathered, we apply the models and design a tour plan in NYC.

6.1 Data

Because there is no existing dataset about NYC's POIs, in this paper we ourselves analyze many datas from reliable sources and experiment with them. We acquire a list of POIs in NYC from the *Trip*

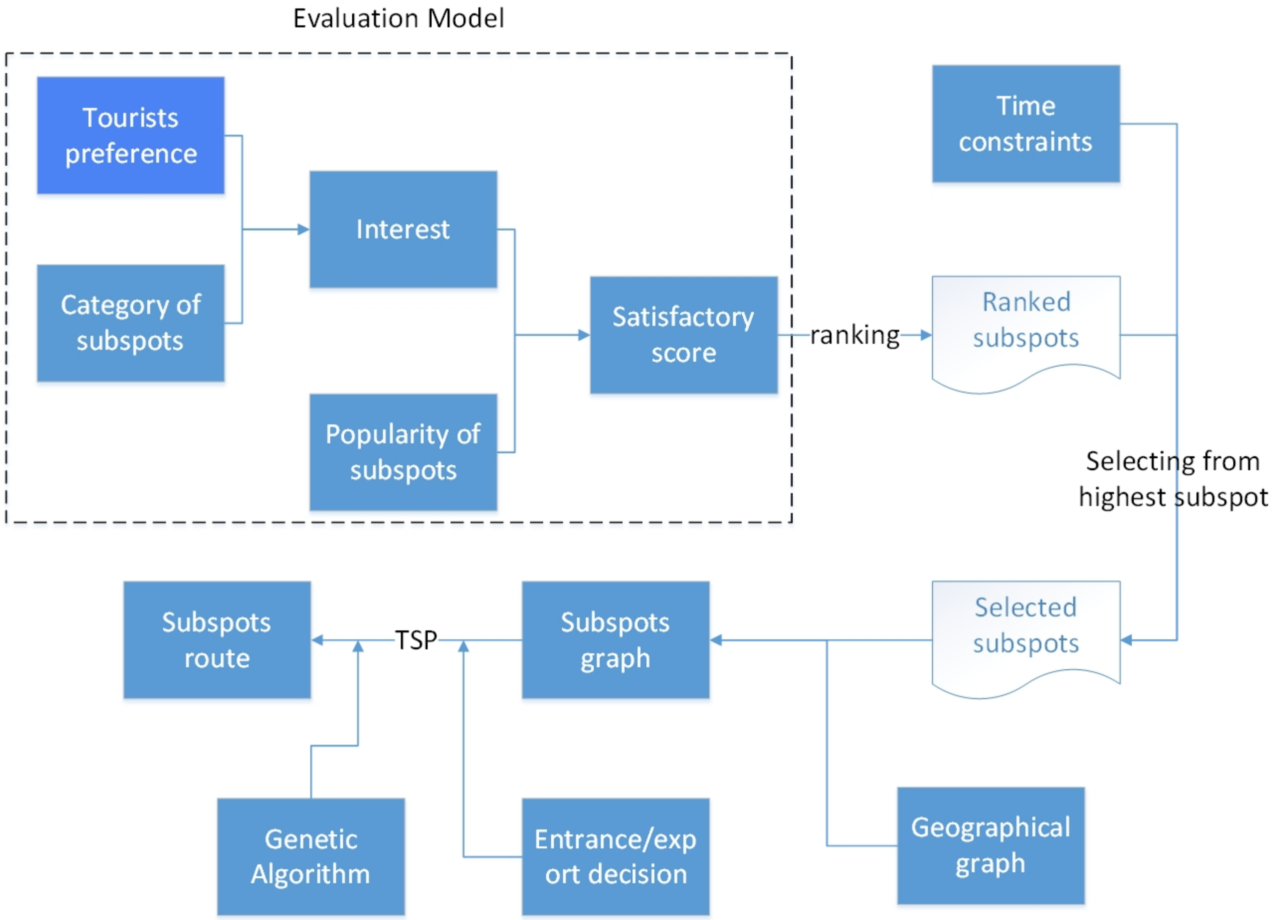

Figure 5: Overview of the revised model

Advisor website[8]. Then we gather detailed information from Google's Places API Web Service[9], including:

- The position of each POI, namely its latitude and longitude.
- The ticket price of each POI;
- Search volume and public rate of each POI.

Applying the methods mentioned in 4, we get the distance between each two POIs in NYC, and each spot's price and satisfactory score.

6.2 Analysis & Results

Tour plan For the spots in NYC, we define five categories to classify each spot:

- Architectural building
- Landmark
- Monument / statue
- Historical site
- Church

We simulate some "users" and input their requests (available time, preference of POIs, etc.) to our PTPs and generate personalized tour plans for them. For example, a hypothetical user inputs a preference $\overrightarrow{prf} = [2, 3, 1, 4, 5]$, which means he/she prefers historical sites and churches most among all spots. Figure 6 shows the route generated by PTPs. Another hypothetical user (who is enthusiastic

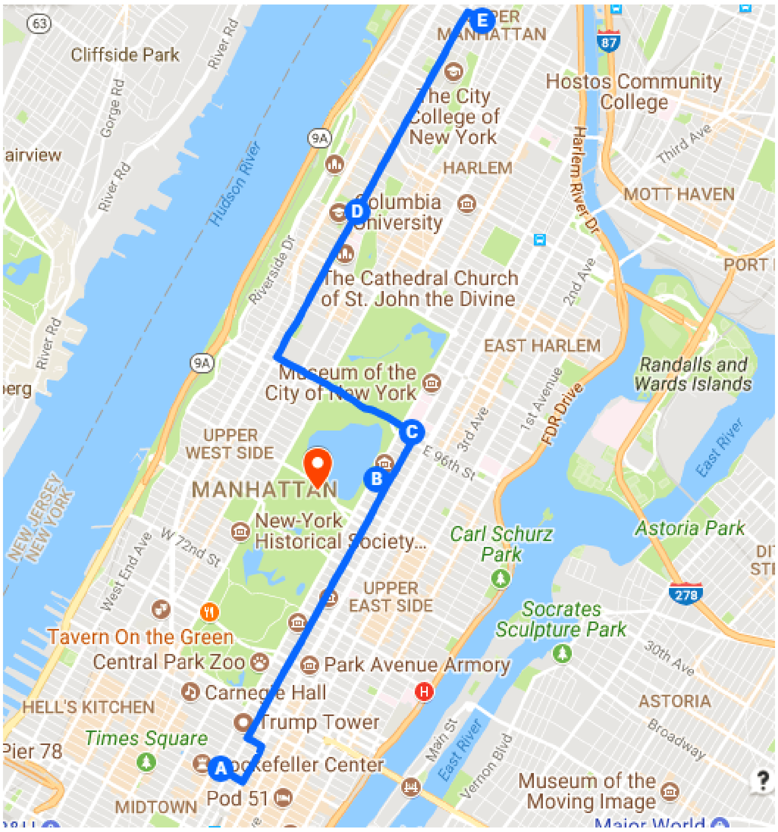

Figure 6: PTPs-generated tour route.

about statues and architectures) inputs a preference $\overrightarrow{prf} = [4, 1, 5, 3, 2]$. Many statues and monuments are in Central Park, which is a super POI, so the tour plan includes an inner route in Central Park. The tour plan PTPs offers is presented in table7. As we can see from the table, most of spots recommended

Morning	Time	Afternoon	Time
Empire State Building	8:00 - 9:30	**Super POI : Central Park**	
The Kneeling Fireman	9:41 - 10:01	USS Maine Monument	12:50 - 13:30
Chrysler Building	10:07 - 10:40	Balto Statue	13:35 - 14:05
The Museum of Mordern Art	10:51 - 11:50	Jose de San Martin Statue	14:10 - 14:40
		Alice in Wonderland Statue	14:45 - 15:15
		Group of Bears Statue	15:25 - 15:55
		Alexander Hamilton Statue	16:00 - 16:30

Figure 7: Tour plan for the hypothetical user

are monuments and statues. And although the hypothetical user is not so interested in landmarks, the tour plan includes a visit to Empire State Building, because it is too popular to miss. Figure 8 shows the plan of route in a visable way.

Figure 8: Tour route with a super POI.

Analysis of result

To examine the effectiveness of our PTPs, for each "user" we generate a series of alternative tour plans using a randomized algorithm. We calculate the total satisfactory score for each tour plan, including the PTPs-recommended one and the randomized ones. We find out that the tour plan PTPs generates has great advantages in comparison to other plans, that is to say, has the highest satisfactory score among the plans.

For further discussion, we calculate the satisfactory scores (fitness values) under different t_{max}, and visualize the result in figure 9.

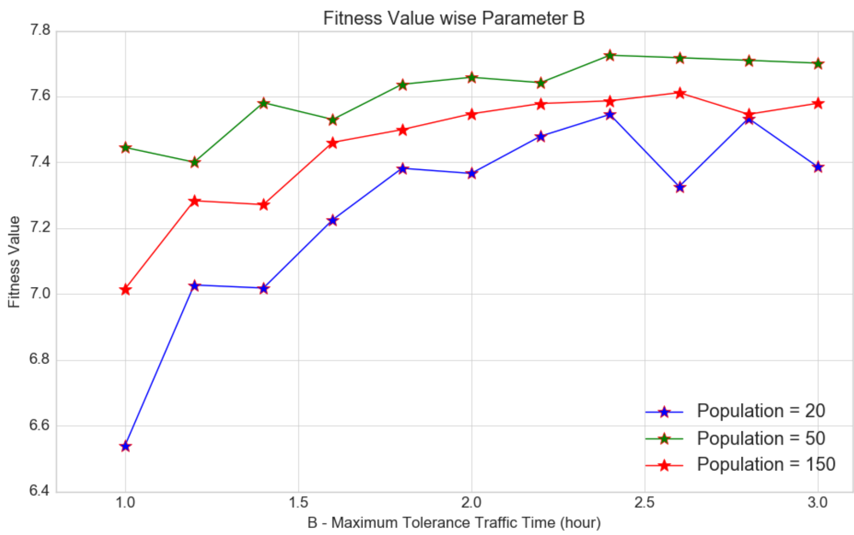

Figure 9: Relationship between satisfactory score and traffic time limit.

We then set different duration (days) of tour and respectively calculate the total satisfactory score Sat. Figure 10 shows the relationship between Sat and tour duration.

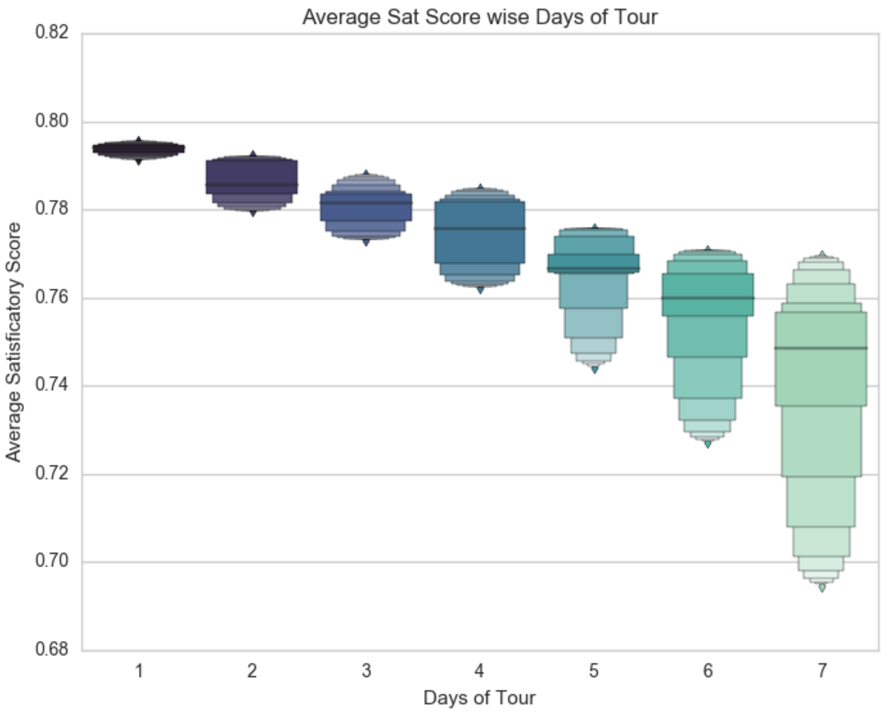

Figure 10: Sat-tour duration relationship

7 Model Analysis

7.1 Sensitivity Analysis

7.2 Strengths & Weaknesses

7.2.1 Strengths

1. Have the ability of cold-start. Our model can give personalized tour plan without the user's historical data.
2. Combine the processes of choosing which POIs to go and designing the route in one app/website.
3. Use massive accurate data of New York City. Only with these dependable data can we validate that our model is built appropriately.
4.

7.2.2 Weaknesses

1. Roads between POIs are simplified to spherical lines, while real roads can include detours, slopes and even bridges across rivers.
2. Complex traffic conditions in cities are omitted. We simplify the traffic network to weighted graph. However, in the real condition there are much more factors influencing the traffic time.
3. Tourist's satisfaction is assumed only related to his/her interest and the POI's popularity. The impact of complex subjective and emotional factors during the tour are not considered.

8 Conclusion

References

[1] E. E. Cicerali, L. K. Cicerali, and A. Saldaml, "Linking psycho-environmental comfort factors to tourist satisfaction levels: Application of a psychology theory to tourism research," *Journal of Hospitality Marketing & Management*, no. 8, 2017.

[2] Y. Wen, Y. Shi, G. Cai, Y. Miao, and G. Long, "A survey of personalized travel recommendation," *Computer Science*, no. Z11, 2014.

[3] X. Wang, M. Chen, and X. Li, "Research and design of constraint-based travel recommendation system," *Computer Technology and Development*, pp. 83–125, 2012.

[4] W. Zhang, L. Han, D. Zhang, P. Ren, J. Ma, and Z. Chen, "Geopmf:a distance-aware tour recommendation model," *Journal of Computer Research & Development*, 2017.

[5] Q. Wu, Y. Zhou, D. Wen, and Z. He, "Personalized trip itinerary recommendation based on user interests and points of interest popularity," *Journal of Computer Applications*, 2016.

[6] Q. Li, W. Xu, and H. Jiang, "Personalized recommendation for traveling planning based on online word-of-mouth," *Business Review*, vol. 28, no. 6, pp. 113–118, 2016.

[7] B. Hao, Y. Tan, W. Wang, and J. Jia, "Intellegent tourism route planning system based on ant colony optimization," *Journal of Jinggangshan University (Natural Science)*, no. 1, pp. 8–13, 2015.

[8] "New york city travel guide," Trip Advisor. [Online]. Available: https://www.tripadvisor.com/Travel_Guide-g60763-New_York_City_New_York.html

[9] "Places api web service," Google. [Online]. Available: https://maps.google.com

Get FREE from Tour Planning!

Want to explore a new city? Hard to decide where to go?

Ooooh, a tour should be relaxed. Don't let tour planning burden you!

Why not experience the convenience *Where to GO* provides!

Personalized tour planning, a touch and all done.

Appendices

Appendix A Implemented Genetic Algorithm

```
global score distance B
[score distance] = load_data();
B = 0;
%-------------constrain of GA algorithm--------------------
n = length(score);
%num of POI
l = 10
LB = ones(l,1);
UB = ones(l,1)*n;
intcon = [1:l];
tol = 0.1;
%----------------------------------------------------------

%------------Set Training Time and interval----------------
k = 11; %Num of B value
Fval = zeros(k,1);
Bval = linspace(1,3,k);
Pval = [10,20]
FResult = zeros(2,k);
%----------------------------------------------------------

%--------------------Training------------------------------
for m = 1:2
    P = Pval(m)
for i = 1:k
    B = Bval(i)
    options=optimoptions(@ga,'MaxGenerations',1000,'ConstraintTolerance',tol,'Populationsize',P);
    Sum = 0;
    for j = 1:5
        [x,fval,exitflag,output,population,scores] = ga(@fitness_func,l,[],[],[],[],LB,UB,@constrain
        Sum = Sum + fval;
    end
    Fval(i) = -Sum/5;
end
    FResult(m,:) = Fval
end
%-------------------Visualization--------------------------
plot(Bval,Fval);
hold on;
```

Appendix B Fitness Function

```
function [ans] = fitness_func(x)
%length(x) = 110
%total_time_bound
global score startpoint;
B = 0.5;
v = 20;%velocity
Num_of_POI = length(x) - sum(x==0);
ans = 0;%fitness value
now = 0;
time = 0;
visit = zeros(110,1);
for i = 1:length(x)
    now = uint32(x(i));
    if now ~= 0
        ans = ans - score(now,1);
    end
end
end
```

Appendix C Calculation of satisfactory score

```
function [score,distance] = load_data(alpha,beta,gamma,pref)
%alpha ¿řţžñ
%beta  ¿řţ7
%gamma ¿řţĺůčl'ś
%score = alpha*map(type) + beta*rating + gamma*Viewer
alpha = 0.5
beta = 0.3
gamma = 0.2
% pref = [3,1,4,5,2]
pref = [5,4,2,1,3];
```

```
    data = xlsread('C:\Users\lenovo\Desktop\MONI\spider\data\Satisfactory_Score\score.xlsx');
    data2 = xlsread('C:\Users\lenovo\Desktop\MONI\spider\data\Traffic\Distance_Matrix_red.csv');
    distance = data2(2:111,2:111);
% Regularize
    total_num = size(data,1);
    Rating = data(:,3);
    Rating = (Rating - min(Rating))/(max(Rating)-min(Rating))
    Viewer = data(:,4);
    Viewer = (Viewer - min(Viewer))/(max(Viewer)-min(Viewer))
    Type = data(:,2);
    Map = zeros(5);
    for i = 1:5
        Map(pref(i)) = 6 - i;
    end
    for j = 1:total_num
        Type(j) = Map(Type(j) + 1)/5;
    end
    Type
    score = alpha*Type + beta*Rating + gamma*Viewer;
end
```