Theme Park Personalized Intelligent Route Guidance

Research Engineer: Chen HuaXing
Student Intern: Gabriel Yee Qi Ming

Lead Project Investigator: Prof. Lau Hoong Chuin
Summary

• Motivation and Background
• Problem Definition
• Literature Review
• Features of Intelligent Guidance
• Solution Approach
  – Utility Mapping
  – Orienteering Problem Solving
• System Architecture
• Experiment
  – Test in Simulation
• Reference
Motivation and Background

• Enhance theme park visitors’ experiences
  Impossible to visit all within time limit
  – Issues of guidebook (outdated, listing info only, generic route and etc)
  – Issues of following the crowd (extra queue time, ignore part of personal interests and etc)
  – Issues of ad-hoc planning on the spot, and etc

• Leverage on networked communication devices (mobile devices and PC)
Problem Definition

• Information given
  – A set of attributes of the patron (preference, health issues, physical limitations and etc)
  – A set of attributes of each attraction in the park (operation status, queue time profile, scale, user’s rating and etc)
  – A duration the patron will stay inside the park

• Expected outcome
  – a personalized route which maximizes user’s experience subject to time constraint
Problem Definition

Purpose of this intelligent guidance

– is not to construct a social coordination which is perhaps better theoretically studied in terms of game theory.

– is to pursue single’s/single group’s interests by assumption over the validity of certain historical info (queue times, operation status and etc).
Literature Review

• Kawamura et al. (2003) proposes a simulation approach to coordinate many visiting agents behavior by some reactive algorithms.
• Matsuo et al. (2003) develops a story-based planning to make a tour plan based on certain plot, and interpret it with some side stories that increase visitor’s experiences.
• Souffriau et al. (2008) addresses a tourist trip design problems by using vector space model to calculate the score, and solving the resultant orienteering problem (prize collecting TSP).
• Fischeti et al. (1998) Solves the orienteering problem optimally via branch-and-cut.
Literature Review

• The orienteering problem could be solved by a collection of metaheuristic methods such as genetic algorithm in Tasgetiren (2001), harmony search in Geem et al. (2005), variable neighborhood search in Sevkli et al. (2006) and Archetti et al. (2007), tabu search in Archetti et al. (2007), a guided local search in Vansteenwegen et al. (2009), iterated local search in Vansteenwegen et al. (2009), path relinking in Souffriau et al. (2010),

• A recent survey on orienteering problem is given in Vansteenwegen et al. (2011).
Intelligent Guidance Features

• Taking Personalized Information
  – Personal preference (wet, dark, thrill and etc)
  – Theme based visit (family trip, exciting trip and etc)
  – Health issues (heart, pregnancy and etc)
  – Physical limitation (height, weight and etc)

• Taking Dynamic Information
  – Weather, breakdown and etc
Intelligent Guidance Features

• Holistic plan for the route
  – Sequence dependent relationship (e.g., no meal immediately before/after rollercoaster)
  – Route will be generated to maximize the overall utility (experience) of the patron

• Real-time replan for the route (Dynamic)
  – Given changes on weather, breakdown, dynamically replan (IP)
Solution Approach

• Solution Approach
  – Utility Mapping
    • Convert a collection of info (user personalized info, historic info of attraction) into a time dependent utility(experience) value for each attraction
  – Orienteering Problem
    • Find a route which maximizes the overall utility
    • Solve heuristically here for prompt user response

• Outcome-A route
  – that is personalized
  – that is holistically planned, yet able to be updated dynamically
Utility Mapping in a Nutshell

• Utility Mapping

Attributes $J$ affecting user’s experience could be categorized into 3 groups: critical subset, quantitative subset, and subjective subset.

For each user profile the utility $S_i$ of an attraction $i$ could be expressed as,

$$S_i = CS_i[\alpha * QS_i + (1 - \alpha) * SS_i]$$

- Critical Score $CS_i$

  $$CS_i = \prod CS_{i,j}$$
  The individual critical score $CS_{i,j} \in \{0,1\}$ is evaluated on each attribute $j$ of critical subset (height limit, weight limit and etc).

- Objective (Quantitative) Score $QS_i$

  $$QS_i = \sum QS_{i,j} / M$$
  The individual quantitative score $QS_{i,j}$ is evaluated on each attribute $j$ of quantitative subset (queueing time, ride duration and etc) and the sum is normalized by a constant $M$.

- Subjective Score $QS_i$

  $$SS_i = \sum w_j SS_{i,j}$$
  The individual subjective score $SS_{i,j} \in (0,1)$ is evaluated on each attribute $j$ (wetness, darkness, thrill and etc).
Utility Mapping in a Nutshell

• Sequence Dependent Relationship

For each pair of activities (i.e. attractions visited) $i_1, i_2 \in I$, there will be an assigned relationship value $P$ between 0 and 1, which could be denoted as $(i_1, i_2, P_{i1i2})$. E.g. if eating lunch and activity of intensive movement are not desired to be done one immediately after another, a value close to 1 could be assigned for the tuple. Similar, 0 is assigned to $P$ for any tuple of one activity does not impact the other.

E.g. exponential-corrected utility of attraction $i \in R$ is expressed as $S_i[1 - \sum_j P_{ji}^{\Delta T_j}]$, $j \in R$ (note that $j$ is visited before $i$ and $\Delta T_j$ is the time difference between visiting $j$ and visiting $i$). We could easily tweak the formulation to model the phenomenon that one activity enhances the utility of the other.
Algorithm for Orienteering Problem

• Construction heuristic + Local Search
• Construction heuristic

\[ \text{Do} \]
\[ \text{For} \text{ each non included attraction} \]
\[ \text{Determine the best possible insertion by looking at} \]
\[ (K \times \text{UtilityGained/Timeincrement}) \]
\[ \text{End For} \]
\[ \text{Insert the attraction at position of the best value} \]
\[ \text{Update current route’s schedule} \]
\[ \text{Update the inserted attraction’s status} \]
\[ \text{If no more insertion gaining positive utility} \]
\[ \text{return route;} \]
\[ \text{End If} \]
\[ \text{While (True)} \]
Algorithm for Orienteering Problem

- Local Search(IP)
  - Shake(remove random no of attractions, insert)
  - Path relink (visit virtual paths between starting solution and guiding solution)
  - 2-opt, 3-opt, 4-opt
Implementation System Architecture

- Access from Web browser
- Access from Phone Application (Android, Apple, Windows phones)
Experiments

- Import queue time profiles from simulator
- Suggest a route based on queue time profiles inserted into the intelligent guidance system
- Run suggested route on the simulator
Test in the Simulation

• Test Scenario 1 [TestRoute1.pdf]
  – Suggested Route is from 10am to 6pm
  – Actual Schedule from simulation is from 10am to 6:06pm, 3 attractions skipped (one is not open, the other two are shows already start when arriving)

<table>
<thead>
<tr>
<th>ID</th>
<th>Amount of location visited</th>
<th>Order of visits</th>
<th>Duration Of Stay (Hours)</th>
<th>Time Entered</th>
<th>Time Exit</th>
<th>Amount of Attractions</th>
<th>KPI</th>
<th>Amount of Shops</th>
<th>Amount of Restaurants</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>31</td>
<td>[7 11 10 1 3 2 21 12 13 16 14 15 8 6 17 22 23 9 24 25 26 27 28 29 30 31 32 33 34 35 36]</td>
<td>8.1</td>
<td>10am</td>
<td>6.06pm</td>
<td>15</td>
<td>1.85185</td>
<td>1852</td>
<td>12</td>
</tr>
</tbody>
</table>
A Version of the System

• http://202.161.45.164/USS/
Reference


