A personalized route recommendation service for theme parks using RFID information and tourist behavior

Chieh-Yuan Tsai a,*, Shang-Hsuan Chung b

a Department of Industrial Engineering and Management, Yuan Ze University, Taiwan, ROC
b Department of System and Engineering Management, ChipMOS Technologies Corp., Taiwan, ROC

A R T I C L E  I N F O
Article history:
Received 15 November 2010
Received in revised form 20 September 2011
Accepted 11 October 2011
Available online 18 October 2011

Keywords:
Theme parks
RFID
Recommendation systems
Visiting sequences

A B S T R A C T
Like any other industry, theme parks are now facing severe challenges from other entertainment competitors. To survive in a rapidly changing environment, creating high quality products/services in terms of consumer preference has become a critical issue for theme park managers. To fulfill these needs, this paper develops a route recommendation system that supplies theme park tourists with the facilities they should visit and in what order. In the proposed system, tourist behaviors (i.e. visiting sequences and corresponding timestamps) are persistently collected through a Radio-Frequency Identification (RFID) system and stored in a route database. The database is then segmented into sub-groups based on the similarity among tourists’ visiting sequences and time lengths. Whenever a visitor requests a route recommendation service, the system identifies the sub-group most similar to that visitor’s personal preferences and intended visitation time. Based on the retrieved visiting behavior data and current facility queuing situation identified by the RFID system, the proposed system generates a proper route suggestion for the visitor. A simulation case is implemented to show the feasibility of the proposed system. Based on the experimental results, it is clear that the recommended route satisfies visitor requirements using previous tourists’ favorite experiences.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction
A theme park, applying themes to provide visitors with interesting experiences different from daily life, is an aggregation of attractions including architecture, landscape, rides, shows, food services, costumed personnel and retail shops [18]. Theme parks, especially regional parks, spread endemic across the US in the late 1960s and 1970s. In the 1980s and into the 1990s, most parks were developed as destination parks [4]. Well-known examples include Disney World, Disneyland, Universal Studios and Six Flags. Although the theme park industry has enjoyed steady attendance growth in the past several decades, the theme park market has entered a mature stage and is no longer experiencing high growth in terms of new development [8]. The mature theme park business became highly dependent on a higher proportion of return visitors and faced new competition from other leisure and tourism products [3].

To survive in a rapidly changing environment, theme parks need to provide high quality services in terms of consumer tastes and preferences [13]. Understanding the spatial and temporal behavior of tourists could enhance the management of attractions and contribute to extending the geographical distribution of tourists and tourist expenditures within regions [21]. Knowing which rides have been taken, which shows have been attended and which shops and squares have attracted the attention of tourists, could lead to radical improvement in satisfaction performance.

Recently, the recommender technique has been regarded as a popular technique for recommending interesting items for visitors in the tourism industry [12]. Personalized tourism services are aimed at helping the user find what they are looking for by comparing the user profile to some reference characteristics without spending much time and effort. Therefore, a variety of approaches have been used to perform recommendations in these domains, including content-based, collaborative, demographic, knowledge-based or hybrid approaches [2, 11, 15]. Abowd et al. [1] proposed a mobile context aware tour guide, called CyberGuide, which allows its users to leave messages to exhibit owners and send reports about his/her location to some central service that others can access. This system detects the user’s location using GPS for an outdoor environment and RF for the indoor version. Fleck et al. [6] developed a “guidebook” system for the Exploratorium in San Francisco. The guidebook prototype provides two communication functions, named rememberer and communicator. The rememberer provides visitors with the means to build a record of their experiences. The communicator function helps visitors communicate using electronic bulletin boards for individual exhibits, instant-messaging and/or beaming information between handheld devices. Huang and Chuang [10] utilized the association
rules mining technique to form a set of recommended exhibits. Based on the mining results, this guide system shows related exhibits via the active model and location-aware service when users check for a specific exhibit.

Niakiki and Kim [17] presented a generic ontology-based architecture using a multi-criteria decision making technique to design a personalized route planning system. An ontology-based knowledge modeling technique uses an analytical hierarchical process to determine the choice of criteria for applying an impedance function in the route finding algorithm. Wang et al. [25] presented semantic web technologies for providing personalized access to digital museum collections. These technologies collect the user’s profile information in context including the user’s personal information, objects that the user has interacted with, user activities over the objects and corresponding contextual information. A content-based filtering method is then employed to recommend artworks and art-history topics to cope with the typical user modeling problems. Huang and Bian [9] proposed an intelligent recommendation system that offers personalized recommendations for tourist attractions at a given destination. Through a tourism ontology application, Bayesian network technique and analytic hierarchy process method, this system provides recommendations to a user by taking into account the travel behavior of the current user and previous users. This system leads to more intelligence, collaboration and personalization in tourist attraction recommendations.

Schiaffino and Amandi [20] presented an expert software agent in the tourism and travel domain, named Traveler. This agent combines collaborative filtering with content-based recommendations and demographic information about customers to make recommendations. The agent assists users by suggesting a package of holidays that are presumably interesting for the user according to his/her profile. The results revealed that a combination of these three approaches overcomes the difficulties in each method used in isolation. García-Crespo et al. [7] presented the SPETA system, which uses knowledge of the user’s current location, preferences, as well as a history of past locations to provide the type of recommender services that tourists expect from a real tour guide. The SPETA framework, taking advantage of Web 3.0 technologies, provides a fully-fledged platform, architecture and a proof-of-concept implementation that presents, on one hand, a combination of the mentioned cutting-edge technologies and emerging standards; and on the other hand, reflects research efforts and innovation.

The above tourism recommendation systems have demonstrated themselves efficient tools by designing user interfaces that can smoothly interact with the environment, providing convenient information query tools, or suggesting a set of associated products (or services). However, few studies focused on how to offer tourists a customized visiting itinerary that guides them completely through their trip [24]. For example, most previous tourism recommendation systems suggest that places A, B, C, and D are worth visiting, but they do not provide information that route A→B→C→D is better than C→B→A→D. However, the visiting sequence is a very important factor that helps tourists complete their trips on time. Without a personalized route suggestion, tourists tend to make an inefficient trip or even get lost in the complex theme park environment.

Actually, four important considerations should be included when making a theme park route recommendation. First, most visitors do not have plenty of time to visit the whole theme park and wish to finish their trip close to their given intended-visited time. If the time constraint is not taken into account in the suggestion, tourists will feel very rushed or even have no time to visit their favorite rides. Second, tourists usually have a set of “must-play” rides in mind before starting their trips in the theme park. Tourists will feel disappointed if these favorite rides are not included in their visiting itinerary. Third, instead of considering the route recommendation problem as an optimization problem that minimizes the total visiting time, route suggestions should be based on the visiting behaviors of previous tourists. Finally, quite often in theme parks visitors congregate in certain areas while at the same time other adjacent areas are vacant. If the recommendation system can take the crowd situation into consideration and provide a less congested route, the service quality of the theme parks should be higher [14].

To achieve the above goals, this research developed a route recommendation system that provides personalized visiting routes for tourists in theme parks that consider a set of visiting constraints. The remainder of this paper is organized as follows. Section 2 introduces the system architecture and assumptions for the discussed theme parks. Section 3 details the proposed route recommendation system. Section 4 provides an example to show the feasibility and performance of the proposed system. Section 5 highlights the contribution and limitation of this research. Section 6 summarizes this research and points out future research directions.

2. System architecture and assumption

Recently, radio-frequency identification (RFID) systems have been successfully applied in many theme parks such as Legoland amusement park in Denmark, Steamboat Ski Resort (CO), Wild Rivers (CA), Dolly’s (TN) and Keylme Cove Water Resort (IL) in the USA [5, 19]. In this research, a RFID system is assumed to be available in the studied theme parks. When tourists enter a theme park, they are provided with a wristband embedded with a RFID tag with a unique electronic product code (EPC) [16, 26]. RFID readers are installed in the entrance and exit of each recreation facility (ride). Whenever a tourist enters the navigating area of a RFID reader, the reader will record the corresponding EPC code and time and transfer this information to the Ride Information Server and the Route Database Server, as shown in Fig. 1. The recording process is repeated until the tourists leave the theme.
park. Note that, with this process, the number of tourists in the queue of each ride is monitored in real time by the RFID system and stored into the Ride Information Server.

Public information booths which contain RFID readers are placed at the locations of rides, food courts, souvenir shops and information centers to provide route recommendation service to visitors. The visitor can initiate this service by approaching his/her wristband and input his/her personal preference at the booth. The booth will transfer the visitor’s EPC code, booth location, current time and personal preference to the Route Recommendation System. The Route Recommendation System, which is the core component of the proposed system, generates an appropriate route recommendation based on the queue information of each ride from the Ride Information Server and visiting sequences from the Route Database Server. The route recommendation will be transferred back to the booth for the visitor’s reference. Fig. 2 illustrates the route recommendation service request process.

3. Route recommendation system

The Route Recommendation System consists of three major modules, as shown in Fig. 3. The first module, the tourist clustering module, segments all tourists’ visiting sequences in the route database into sub-groups based on the dissimilarity among the tourists’ visiting sequences and time lengths. The second module, the sub-group retrieval module, finds the sub-group that is most similar to a visitor’s input preference. The personal preference includes the intended departure time, favorite thematic regions with preferred order and wished staying time length, and favorite rides. The last module, the route generation module, takes the visiting behavior data identified in the second module and the queuing information from each ride identified by the RFID system to generate an appropriate visiting recommendation for the visitor.

3.1. Data preparation and preprocessing

In general, a theme park might contain more than a hundred recreation facilities (also called rides), and each of them belongs to a specific thematic region. Let $R = \{R_1, R_2, ..., R_n\}$ be the set of rides in the theme park and $T = \{T_1, T_2, ..., T_m\}$ be the set of thematic regions in the theme park. Based on theme park practices, a transformation function that describes the hierarchical relationship between $R_i \in R$ and $T_j \in T$ is:

$$f : R_i \rightarrow T_j.$$  

For example, based on the hierarchical structure defined in Fig. 4, the transformation function is defined as $f(R_1) = f(R_2) = f(R_3) = T_1$, $f(R_4) = f(R_5) = f(R_6) = f(R_7) = T_2$, and so on.

![Fig. 3. The three modules in the Route Recommendation System.](image-url)
Let a visiting record in the route database $RD$ be represented by $<cid, vs>$, where $cid$ is a record identifier and $vs$ is a visiting sequence. The visiting sequence $vs$ is represented as $\{(r_1, t_{s_1}, \tau_{e_1}), (r_2, t_{s_2}, \tau_{e_2}), \ldots, (r_m, t_{s_m}, \tau_{e_m})\}$ where $r_i \in R$, and $t_{s_i}$ is the arrival time to $r_i$, $\tau_{e_i}$ is the departure time from $r_i$, and $t_{s_i} < \tau_{e_i}$ for $i = 1, \ldots, n$. Table 1 illustrates an example visiting sequence in a route database $RD$.

In practice, visiting sequences collected by the RFID system might contain many inappropriate sequences. For example, in a theme park, tourists tend to traverse popular rides more than one time. However, a recommendation system should not suggest the same activities again and again. Therefore, a redundancy check should be conducted. That is, if a ride is visited more than once, the redundancy check procedure will keep the first ride in the sequence and remove the recurring visits to that ride since the ride taken in the front position of the visiting sequence is considered as having higher preference. For example, the visiting sequence $cid$ in Table 1 should be remedied as $((R_1, 10,30), (R_2, 40,80), (R_3, 85,95), (R_4, 100,120), (R_5, 126,146), (R_6, 150,190), (R_7, 200,250))$ since redundancies $(R_6, 125, 180)$ and $(R_7, 200, 300)$ appear in the sequence. In addition, if a visiting sequence contains only one ride, it should be eliminated. For instance, $cid$ 5 in Table 1 poses only one ride experience and should be removed. Table 2 shows the visiting sequences after taking data preprocessing for the route database in Table 1.

### 3.2. Tourist clustering module

The tourist clustering module segments all visiting sequences in $RD$ into a set of sub-groups in which each sub-group contains a set of similar visiting sequences. However, clustering visiting sequences at the ride level might produce two problems. First, it is difficult and trivial to find similar visiting sequences at the ride level because there are a great number of rides in a visiting sequence. Second, tourists at a theme park tend to visit thematic regions one by one. That is, they select a target thematic region first and then move to the next thematic region after completing the activities in the first region. Therefore, instead of clustering similar visiting sequences at the ride level, this research transforms the visiting sequences represented at the ride level into sequences represented at the theme level. The ride sequences are then clustered at the theme level.

#### 3.2.1. Transformation process

The transformation process transforms a visiting sequence represented at the ride level into a sequence represented at the theme level according to the following four steps. First, ride $r_i$ in a visiting sequence is replaced by the corresponding theme $\rho_i$ for all visiting sequences in $RD$ according to the transformation function in Eq. (1). The visiting sequence $((R_1, t_{s_1}, t_{e_1}), (R_2, t_{s_2}, t_{e_2}), \ldots, (R_m, t_{s_m}, t_{e_m}))$ is represented at the ride level where $r_i \in R$ is transformed to the visiting sequence represented at the theme level as $((t_1, t_{s_1}, t_{e_1}), (t_2, t_{s_2}, t_{e_2}), \ldots, (t_n, t_{s_n}, t_{e_n}))$ where $t_j \in T$. Second, in a visiting sequence, if the number of rides consecutively taken within the same theme $t_j$ is less than a user-defined consecutive ride threshold, $\rho_j$, those rides will be removed from the sequence because those rides are not significant enough to be represented in a visiting experience at the theme level. For simplicity, $\rho_j$ can be set as the average number of rides consecutively taken by all tourists in theme $t_j$.

Third, rides consecutively taken within the same thematic region are aggregated into their corresponding thematic region. After aggregation, a sequence is represented as $((t_1, t_{s_1}, t_{e_1}), (t_2, t_{s_2}, t_{e_2}), \ldots, (t_n, t_{s_n}, t_{e_n}))$ where $t_{s_j}$ is the starting time of the first ride belonging to theme $t_j$ and $t_{e_j}$ is the departure time for the last ride belonging to theme $t_j$. If the time length staying at a theme $t_j$ is less than a user-defined minimum time threshold, $\varphi_j$, that theme experience should be ignored and eliminated from the sequence. In this study, $\varphi_j$ can be set as the average time length spent by all tourists for theme $t_j$ and the time length at theme $t_j$ denoted as $t_{l_j}$, is calculated according to Eq. (2). After completing steps one to four, a visiting sequence is then represented as $((t_1, t_{l_1}), (t_2, t_{l_2}), \ldots, (t_n, t_{l_n}))$.

\[
\begin{align*}
\forall j & \in \{1,2,\ldots,m\} \\
t_{l_j} &= \begin{cases} 
    \left(\frac{t_{e_j} + (t_{s_{j+1}} - t_{e_j})}{2} - t_{s_j}\right) & \text{if } j = 1 \\
    \left(\frac{t_{e_j} + (t_{s_{j+1}} - t_{e_j})}{2} - \frac{t_{s_j} - (t_{s_{j-1}} - t_{e_{j-1}})}{2}\right) & \text{if } 1 < j < m \\
    \left(\frac{te_{j-1} - te_j}{2}\right) & \text{if } j = m
    \end{cases}
\end{align*}
\]

Let’s take the route database in Table 2 and the hierarchical structure in Fig. 4 as an example. The visiting sequence for $cid$ 1 $((R_1,10, 30), (R_2, 40,80), (R_3, 85,95), (R_4, 100,120), (R_5, 126,146), (R_6, 150,190), (R_7, 200,250))$ is transferred as $((T_1, 10, 30), (T_2, 40,80), (T_3, 85,95), (T_4, 100,120), (T_5, 126,146), (T_6, 150,190), (T_7, 200, 250))$ in step one. If the consecutive ride threshold $\rho_j$ is set as 2 for all themes, $T_3, 85,95$ will be removed because only one ride is taken in theme $T_3$. Therefore,

<table>
<thead>
<tr>
<th>Table 1</th>
<th>An example route database $RD$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cid</td>
<td>Visiting sequence</td>
</tr>
<tr>
<td>1</td>
<td>((R_1,10,30), (R_2,40,80), (R_3,85,95), (R_4,100,120), (R_5,126,146), (R_6,150,190), (R_7,200,250))</td>
</tr>
<tr>
<td>2</td>
<td>((R_1,10,45), (R_2,35,45), (R_3,70,95), (R_4,100,135), (R_5,140,180), (R_6,190,250))</td>
</tr>
<tr>
<td>3</td>
<td>((R_1,15,50), (R_2,60,65), (R_3,85,125), (R_4,210,330))</td>
</tr>
<tr>
<td>4</td>
<td>((R_1,5,50), (R_2,45,65), (R_3,70,95), (R_4,100,140))</td>
</tr>
<tr>
<td>5</td>
<td>((R_1,50,100))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>The modified route database after conducting data preprocessing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cid</td>
<td>Visiting sequence</td>
</tr>
<tr>
<td>1</td>
<td>((R_1,10,30), (R_2,40,80), (R_3,85,95), (R_4,100,120), (R_5,126,146), (R_6,150,190), (R_7,200,250))</td>
</tr>
<tr>
<td>2</td>
<td>((R_1,10,45), (R_2,35,45), (R_3,70,95), (R_4,100,135), (R_5,140,180))</td>
</tr>
<tr>
<td>3</td>
<td>((R_1,15,50), (R_2,60,65), (R_3,85,125))</td>
</tr>
<tr>
<td>4</td>
<td>((R_1,5,50), (R_2,45,65), (R_3,70,95), (R_4,100,140))</td>
</tr>
</tbody>
</table>
the visiting sequence for cid 1 will be modified as ((T1, 10, 30), (T1, 40, 80), (T2, 100, 120), (T2, 126, 146), (T4, 150, 190), (T4, 200, 250)) in step two. cid 1 is then aggregated as ((T1, 10, 80), (T2, 100, 120), (T4, 250, 250)) in step three. Based on Eq. (2), $t_{d1} = [100 - (100 - 80)/2] = 10 = 80$, $t_{d2} = [146 + (150 - 146)/2] = 152$, $t_{d4} = 250 - [150 - (150 - 146)/2] = 102$. If the minimum time threshold $\phi$ is set at 70 for all themes, the visiting sequence cid 1 is transformed as (((T1, 10, 80), (T4, 102))) in step four since the time staying in theme T2 is not significant enough.

3.2.2 Dissimilarity measurement

Let a visiting sequence $S_a$ be $((t_{la1}, t_{la2}), (t_{la3}, t_{la4}), \ldots)$ and $S_b$ be $((t_{lb1}, t_{lb2}), (t_{lb3}, t_{lb4}), \ldots)$ where $t_{laq} \in T_l$ and $t_{lbq}$ are the corresponding time lengths. The dissimilarity measure between $S_a$ and $S_b$, denoted as $DSim(S_a, S_b)$, is defined as:

$$DSim(S_a, S_b) = \sum_{k=1}^{l} \left( w_k \times Cost_{a,b,k} \right) / \sum_{k=1}^{l} w_k$$

where $l = \text{Max} (-S_{a,b}, |S_a|)$ is the maximal sequence length among $S_a$ and $S_b$, and $w_k = (l - k + 1)/l$ is the penalty weight in position $k$. In addition, $Cost_{a,b,k}$ is the cost that changes ($t_{laq}, t_{lbq}$) in $S_a$ to ($t_{lbq}, t_{lbq}$) in $S_b$ according to the edit distance (or Levenshtein distance) concept which is based on the following rule:

$$Cost_{a,b,k} = \begin{cases} \psi_{a,b,k} & \text{if no change between } t_{laq} \text{ and } t_{lbq} \\ 1 & \text{if deletion, insertion, or substitution} \end{cases}$$

where $\psi_{a,b,k}$ is the time dissimilarity between $t_{laq}$ and $t_{lbq}$ which is defined as:

$$\psi_{a,b,k} = \left| \frac{t_{lbq} - t_{laq}}{\text{Max}(t_{lbq}, t_{laq})} \right|$$

If the two sequences are exactly the same, the dissimilarity between the two sequences is 0. Conversely, if both sequences have no match, the dissimilarity is 1. The detailed cost evaluation algorithm can be found in [22, 23].

3.2.3 K-Medoids clustering algorithm

When the dissimilarities between all pairs of visiting sequences are derived using Eq. (3), the K-Medoids clustering algorithm is applied to cluster all visiting sequences in $RD$. The K-Medoids algorithm is derived from the K-Means algorithm and is more robust in handling noisy data and outliers. A typical K-Means algorithm randomly chooses $K$ initial centroids, where a centroid is a virtual point. In contrast, the K-Medoids algorithm begins by randomly selecting an actual data point, called a medoid. In this study, the number of $K$ sub-groups and dissimilarities between all pairs of visiting sequences are the input to the K-Medoids algorithm, while the output is the $K$ sub-groups ($RD_i | i = 1, \ldots, K$) where $RD_i$ contains the visiting sequences for the $i$th sub-group. Fig. 5 illustrates the pseudo-code for the K-medoids algorithm for clustering all visiting sequences into $K$ sub-groups.

3.3 Sub-group retrieval module

When a visitor wants to request a route recommendation, he/she simply holds his/her wristband to the RFID reader in a public information booth and enters their personal preferences to the booth. The visitor’s preference includes (1) the intended departure time, LT; (2) a set of wished visiting sequences at the theme level FT where $FT = ((t_{11}, t_{12}), (t_{21}, t_{22}), \ldots)$ in which $t_{ij} \in T_l$ is the $i$th thematic region the visitor wishes to visit and $t_l$ is the time length the visitor wished to stay in $t_l$; and (3) a set of favorite rides FR where $FR=\{r_{1j}\}_{j=1}^{T_l}$. In addition, the time a route recommendation is requested is denoted as CT and the public information booth location denoted as BL are automatically collected by the system. The request information (RI) vector sent to the Route Recommendation System is:

$$RI = CT, BL, LT, FT, FR$$

For example, assume that the theme park opens at 8:00 am. If a visitor requests a route recommendation from booth B05 at 8:30 am and enters his intended departure time as 6:30 pm, $CT = 30, BL = B05$, and $LT = 630$ will be collected. The visitor wishes to spend 4 h for his/her first priority thematic region - theme 3 (T3) and 3 h for his/her second priority area - theme 2 (T2), so that $FT = ((T3, 240), (T2, 180))$. Finally, he/she inputs five favorite rides (rides 1, 2, 3, 5, and 6) that he/she wishes to play, which results in $FR = \{r_{11}, r_{12}, r_{13}, r_{15}, r_{16}\}$. Based on the above information, $RI = <30, 805, 630, ((T3, 240), (T2, 180)), \{R_1, R_2, R_3, R_4, R_5\}>$. The sub-group retrieval module first evaluates the dissimilarity values between all medoids for sub-groups and the set of wished visiting sequences at the theme level FT of the RI vector using Eq. (3). The module then returns all visiting sequences in the sub-group $RD_{\text{retreived}}$ in which the medoid of $RD_{\text{retreived}}$ is most similar to FT among all $RD_i$ where $i = 1, \ldots, K$. If more than one sub-group has the same dissimilarity value, the sub-group having the largest data size is selected.

K-Medoids algorithm:

**Input:** the number of sub-groups $K$ and the route database $RD$

**Output:** $K$ sub-groups $\{RC_i | i=1, \ldots, K\}$

1. Begin
2. Randomly choose $K$ data points from $RD$ as medoids $m_1, m_2, \ldots, m_K$.
3. Repeat
4. Assign remaining non-medoid data points to its closest medoid;
5. Compute total distance of $RC_i$, $TD_i$, between $m_i$ and non-medoid $r_j, j=1, \ldots, |RC_i|, i=1, \ldots, K$;
6. For each medoid $m_i$ do
7. Select the non-medoid $r_j$ for which $TD_i$ is minimal;
8. Compute $TD_i(r_j \rightarrow m_i)$;
9. If $TD_i(r_j \rightarrow m_i)$ is smaller than the current $TD_i$;
10. Swap $m_i$ and $r_j$;
11. End if
12. End for
13. Until no $m$ changes;
14. End

Fig. 5. The pseudo-code for the K-medoids algorithm.
often a ride is visited and how long tourists stay at a ride. Let 
be two rides in the theme park where 


Support count 


3.4.1. Recommendation matrix
generate a proper visiting sequence for the visitor’s reference.


Input: a route generation algorithm is then developed to


Based on the recommendation matrix and the visitor’s personal preference, a route generation algorithm is then developed to generate a proper visiting sequence for the visitor’s reference.

3.4.1. Recommendation matrix
In general, if the ride utilization is high or the staying time at a ride is long, the ride is popular and interesting to tourists. Therefore, the previous tourists’ experience can be evaluated in terms of how often a ride is visited and how long tourists stay at a ride. Let X and Y be two rides in the theme park where X and Y ∈ R. The selection preference \( SP(X, Y) \) is defined as the occurrence frequency that tourists visit ride Y right after visiting ride X, and is evaluated as:

\[
SP(X, Y) = \frac{S_{(X \rightarrow Y)}}{S_X}
\]  
(7)

where support count \( S_{(X \rightarrow Y)} \) is the total number of sequences in \( RD_{\text{retrieved}} \) where any ride is visited right after X is visited. The \( SP(X, Y) \) value is within [0, 1]. In addition, the time preference \( TP(X, Y) \) is defined as the time length staying at ride Y if its immediate preceding ride is X, and is evaluated as:

\[
TP(X, Y) = \frac{T_{(X \rightarrow Y)}}{T_X}
\]  
(8)

where \( T_{(X \rightarrow Y)} \) is the sum of all discretized stay time lengths at Y when tourists visit Y right after visiting X; \( T_X \) is the sum of all discretized stay time lengths for any ride right after X is visited. The value of \( TP(X, Y) \) is within [0, 1]. If a visiting sequence \( v_s \) is represented as \( \{(r_1, t_{s_1}, t_{e_1}), (r_2, t_{s_2}, t_{e_2}), ..., (r_n, t_{s_n}, t_{e_n})\} \) where \( r_i \in R \), the time length \( t_{l_i} \) of ride \( r_i \) can be derived by \( t_{s_i} - t_{e_i} \). The discretized time length of \( t_{l_i} \) can be obtained by:

\[
\text{Discrete}(t_{l_i}) = \begin{cases} j, & \text{if } IW \times (j-1) < t_{l_i} \leq IW \times j \text{ for } j \in \{1, ..., N-1\} \\ N, & \text{if } t_{l_i} > IW \times (N-1) \end{cases}
\]  
(9)

The discretization function enforces time length \( t_{l_i} \) falls into be one of N intervals where \( IW \) is a user-defined time interval width. The discretization function can reduce the number of continuous values and make the analysis more meaningful.

Based on Eqs. (7) to (9), an integrated preference that stands for the previous behavior in taking ride Y right after taking ride X, denoted as \( IP(X, Y) \), is defined as:

\[
IP(X, Y) = SP(X, Y) \times TP(X, Y) = \frac{S_{(X \rightarrow Y)} \times T_{(X \rightarrow Y)}}{S_X \times T_X}
\]  
(10)

The integrated preference value is within [0, 1]. If tourists do not prefer the experience \( X \rightarrow Y \), the integrated preference value should be low. Based on Eq. (10), the integrated preference values between all pairs of rides are calculated and stored in a previous tourist preference (PTE) matrix as shown in Eq. (11). Notes that PTE matrix is a

### Table 3
The visiting sequences after taking discretization.

<table>
<thead>
<tr>
<th>Cid</th>
<th>Visiting sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((R_2, R_3, R_4, R_5))</td>
</tr>
<tr>
<td>2</td>
<td>((R_1, R_2, R_3, R_4, R_5))</td>
</tr>
<tr>
<td>3</td>
<td>((R_1, R_2, R_3, R_4))</td>
</tr>
<tr>
<td>4</td>
<td>((R_3, R_4, R_5))</td>
</tr>
</tbody>
</table>

### Route generation module

To generate a proper route recommendation, both of the previous tourist visiting experiences and real-time queuing situation of each ride should be considered. The previous tourist visiting experience can be derived from the visiting sequences in the sub-group \( RD_{\text{retrieved}} \) while the real-time queuing situation of each ride can be obtained by the RFID system in the theme park. Therefore, a recommendation matrix containing the above information is constructed first in the route generation module. Based on the recommendation matrix and the visitor’s personal preference, a route generation algorithm is then developed to generate a proper visiting sequence for the visitor’s reference.

3.4.1. Recommendation matrix

To generate a proper route recommendation, both of the previous tourist visiting experiences and real-time queuing situation of each ride should be considered. The previous tourist visiting experience can be derived from the visiting sequences in the sub-group \( RD_{\text{retrieved}} \) while the real-time queuing situation of each ride can be obtained by the RFID system in the theme park. Therefore, a recommendation matrix containing the above information is constructed first in the route generation module. Based on the recommendation matrix and the visitor’s personal preference, a route generation algorithm is then developed to generate a proper visiting sequence for the visitor’s reference.

3.4.1. Recommendation matrix

To generate a proper route recommendation, both of the previous tourist visiting experiences and real-time queuing situation of each ride should be considered. The previous tourist visiting experience can be derived from the visiting sequences in the sub-group \( RD_{\text{retrieved}} \) while the real-time queuing situation of each ride can be obtained by the RFID system in the theme park. Therefore, a recommendation matrix containing the above information is constructed first in the route generation module. Based on the recommendation matrix and the visitor’s personal preference, a route generation algorithm is then developed to generate a proper visiting sequence for the visitor’s reference.

### Route generation algorithm:

**Input:** The recommendation matrix (RM) and the request information vector RI = \(<CT, BL, LT, FT, FR>_\rangle.

**Output:** a visiting sequence recommendation (RecRt).

1 Begin
2 \( TC = LT - CT; \)
3 \( MVist = FR; \)
4 \( UnVist = R - FR; \)
5 Assign BL to RecRt(0);
6 \( i = 1; \)
7 While \( MVist == \{ \} \) Do
8 \( RecRt[i] = \text{Search\_Next\_Ride}(RecRt[i-1], MVist, RM); \)
9 Recalculate \( TotVisT \) of \( RecRt; \)
10 Remove \( RecRt[i] \) from \( MVist; \)
11 If \( TotVisT > TC \) Then
12 Remove \( RecRt[i] \) from \( RecRt; \)
13 Output \( RecRt \) and Stop;
14 Else
15 \( i = i+1; \)
16 End if
17 End while
18 While \( TotVisT < TC \) Do
19 If \( UnVist \) is empty Then
20 Output \( RecRt \) and Stop;
21 Else
22 \( r = \text{Search\_Inserted\_Ride}(RecRt, UnVist, RM, \&Position); \)
23 Insert \( r \) into \( RecRt[\text{Position} \rangle\) end \( RecRt[\text{Position} + 1]\); \)
24 Recalculate \( TotVisT \) of \( RecRt; \)
25 Remove \( r \) from \( UnVist; \)
26 End while
27 Remove \( \text{r} \) from \( RecRt; \)
28 Output \( RecRt \) and Stop;
29 End

**Fig. 6.** The pseudo-code for the route generation algorithm.
non-symmetric matrix recording the previous tourist visiting preference from ride \( X \) to ride \( Y \):

\[
PTE = \begin{bmatrix}
0 & IP(R_1, R_2) & IP(R_1, R_3) & \cdots & IP(R_1, R_N) \\
IP(R_2, R_1) & 0 & IP(R_2, R_3) & \cdots & IP(R_2, R_N) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
IP(R_N, R_1) & IP(R_N, R_2) & \cdots & \cdots & 0
\end{bmatrix}_{N \times N}
\]

(11)

Taking Table 2 as an example, if the time interval width \( h \) in Eq. (9) is set as 10 min and the largest time length in a ride is 40 min, 5 time intervals are obtained. Therefore, the discretization function can be defined as: \( \text{Discre}(t) = 1 \) if \( 0 < t \leq 10 \); \( \text{Discre}(t) = 2 \) if \( 10 < t \leq 20 \); \( \ldots \); \( \text{Discre}(t) = 5 \) if \( t > 40 \). After taking discretization, the visiting sequences are displayed in Table 3. Based on Eqs. (7) to (10), the integrated preference \( IP(R_1, R_2) = (2 \times 2)/(3 \times 6) = 0.2222 \) since \( \rho = 2 \), \( \tau_{(1,2)} = 1 + 1 = 2 \), \( \rho = 3 \), and \( \tau_{(R_N)} = 4 + 1 + 1 = 6 \).

In addition to the previous tourist preference, tourists prefer the ride with shorter waiting time because most tourists wish to take more rides within a limited time. Therefore, to avoid a long queue waiting time, the ride queuing situation should be used to adjust the preference value for each ride. In the proposed system, the number of tourists in the queue for ride \( r \), denoted as \( CQ(r) \), can be monitored in real-time through the RFID system. Therefore, the current ride preference (CRP) vector can be represented as:

\[
\text{CRP} = \frac{1}{CQ(r_1)}, \frac{1}{CQ(r_2)}, \ldots, \frac{1}{CQ(r_1)}, \frac{1}{CQ(r_2)}
\]

(12)

If the number of tourists in the queue for ride \( r \) is large, the ride is less preferred. Note that the min-max normalization is applied for CRP so that each element in the vector is within \( [0, 1] \).

Based on previous tourist preference PTE matrix and current ride preference CRP vector, the recommendation matrix is constructed as:

\[
RM = \text{CRP} \times \text{PTE}
\]

(13)

3.4.2. The route generation algorithm

The route generation algorithm is proposed in this research to generate a route recommendation for the visitor. The input to the algorithm includes the recommendation matrix \( RM \) of Eq. (13) and the \( R \) vector \( \langle BL, LT, FT, FR \rangle \) of Eq. (6). The algorithm output is a visiting sequence recommendation, \( \text{RecRt} \). The time constraint \( TC \) is set as \( LT-CT \) to ensure the recommended route can be finished in time. In addition, the MtVist list stores the rides the visitor wishes to visit, while the UnVist list stores the rides the visitor has not visited yet. Initially, the MtVist list is set as \( FR \) and the UnVist list is set as \( R-\text{FR} \) where \( R \) is the set of rides in the theme park. The pseudo-code of the proposed route generation algorithm is shown in Fig. 6.

This algorithm consists of two major phases. The first phase, shown from lines 5 to 18, constructs a primary visiting sequence based on the MtVist list. The first position in the primary sequence is the booth location \( \langle BL \rangle \) where the visitor makes a request. The next ride is then selected from the ride in the MtVist list that has the largest recommendation value using the proposed Search_Next_Ride function. Once the ride is added into the sequence, the total visiting time \( \text{TotVisT} \) for the new sequence is recalculated and the ride is removed from the

Function Search_Inserted_Ride (RecRt, UnVist_list, RM, Position)

1. \( \text{MaxRecV} = 0; \)
2. \( \text{For each CurrentRide in RecRt do} \)
3. \( \text{For each PotentialRide in UnVist_list do} \)
4. \( \text{If Index(CurrentRide) == Index(RecRt) do} \)
5. \( \text{RecValue} = \text{RM}[\text{Index(RecRt)}, \text{Index(PotentialRide)}]; \)
6. \( \text{Else} \)
7. \( \text{RecValue} = \text{RM}[\text{Index(RecRt)}, \text{Index(PotentialRide)}] + \text{RM}[\text{Index(PotentialRide)}, \text{Index(RecRt)}]; \)
8. \( \text{End if} \)
9. \( \text{If RecValue > MaxRecV do} \)
10. \( \text{Assign RecValue to MaxRecV;} \)
11. \( \text{Assign PotentialRide as r;} \)
12. \( \text{Assign Index(PotentialRide) to Position;} \)
13. \( \text{End if} \)
14. \( \text{End for} \)
15. \( \text{End for} \)
16. \( \text{Return r;} \)
17. \( \text{End function} \)

Fig. 8. The pseudo-code for the Search_Inserted_Ride function.
MtVist list. If TotVisT is larger than TC, the ride is removed and the primary visiting sequence is output as a route recommendation. Otherwise, the algorithm will repeat until there are no rides in the MtVist list. In this case, the algorithm will move to the second phase. The second phase of the algorithm as shown from lines 19 to 30 and expands the primary visiting sequence until TotVisT of the recommended sequence is greater than or equal to TC. To fulfill this requirement, the Search_Inserted_Ride function is proposed to find the ride in the UnVisT list that has the largest recommendation value if the ride is inserted between two adjacent rides in the sequence. The position of the insertion into the sequence is returned by the &Position variable. In this way, rides in the UnVisT list are inserted into the primary visiting sequence gradually until TotVisT is greater than TC. The algorithm stops when the sequence fulfills the time constraint or no rides remain in the UnVisT list.

Fig. 7 illustrates the pseudo-code for the Search_Next_Ride function. This function searches the ride in MtVist list having the largest recommendation value based on the recommendation matrix RM. Note that the Index(r) function returns the index number of ride r in the RM matrix. Fig. 8 shows the pseudo-code for the Search_Inserted_Ride function that searches for ride r from UnVisT list and returns its inserted position. Because the ride is inserted between the current ride and the next ride in the sequence, the sum of the recommendation values from the current ride to the inserted ride and from the inserted ride to the next ride should be the largest.

Let us use an example to explain the second phase of the algorithm. Assume that a primary visiting sequence RecSt = R1 → R2 → R3 → R4 is derived from the first phase and UnVisT list = [R4, R5, R6]. The Search_Inserted_Ride function will evaluate the recommendation values of R3 → R6 → R1, R3 → R7 → R1, R3 → R8 → R1, R1 → R6 → R2, R1 → R7 → R2, R1 → R8 → R2, and so on. This process is repeated until all possible combinations are checked. If R1 → R2 → R3 has the largest summarized recommendation value among all, for example, Position will be 2 and r will be R7 at the end of the function. The visiting sequence R3 → R1 → R7 → R2 → R6 → R3 is then generated.

4. Experimental illustration

An example theme park as illustrated in Fig. 9 to demonstrate the feasibility of the proposed route recommendation system. The theme park includes 7 theme regions, 40 recreation facilities (rides) and 1 entrance/exit. The theme park operating hours are from 9 a.m. to 5 p.m.

![Fig. 9. The layout of the example theme park.](image-url)

Table 5
The visiting sequences collected from tourists.

<table>
<thead>
<tr>
<th>Cid</th>
<th>Visiting sequence</th>
<th>Total time (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R4 R6 R5 R2</td>
<td>496.80</td>
</tr>
<tr>
<td></td>
<td>R6 R4 R1 R3 R2 R5</td>
<td>446.60</td>
</tr>
<tr>
<td>5</td>
<td>R6 R4 R5 R2</td>
<td>187.88</td>
</tr>
<tr>
<td>16</td>
<td>R6 R4 R5 R2</td>
<td>658.21</td>
</tr>
</tbody>
</table>

Note: minimum total time: 187.88 (min.), maximum total time: 658.21 (min.), average total time: 547.24 (min.)
Table 4 shows the hierarchical structure between themes and rides in this theme park.

However, the RFID system is not deployed in this example theme park right now. Thus, a web questionnaire accompanying the theme park map is developed to collect the tourist visiting sequences. When a tourist (tester) moves a mouse over the ride position in the theme park map, further information such as the thrill rating, ride time and the rides in this theme park. Table 5 summarizes the number of clusters \(K\) is set as 4 in this implemented, so that four sub-groups whose medoid is \(\text{cid} 4\) (called \(RD_1\)), \(\text{cid} 7\) (called \(RD_2\)), \(\text{cid} 17\) (called \(RD_3\)) and \(\text{cid} 54\) (called \(RD_4\)) are constructed as shown in Table 6.

After the transformation process, 12 visiting sequences are eliminated and 42 sequences are left. The dissimilarity values between all pairs of visiting sequences represented at the theme level are transformed into the sequences represented at the ride level for evaluation. The dissimilarity values between all pairs of visiting sequences at the theme level as (\(T_4\), \(T_9\), \(T_{15}\)) and the set of favorite rides as \(\{R_{38}, R_{24}, R_{15}, R_9, R_{28}\}\). Therefore, the \(RI\) vector is \(\{90, 720, \{T_4, 45\}, \{T_9, 60\}, \{T_{15}, 100\}, \{T_{28}, 140\}, \{T_{38}, 150\}\}\). With the \(RI\) vector, the sub-group retrieval module identifies \(RD_1\) as the sub-group which is the most similar to the visitor’s personal preference. Since the dissimilarity value between the input data and \(RD_1\) is 0.491, \(RD_2\) is 1.000, \(RD_3\) is 0.519, and \(RD_4\) is 0.793. This result is expected according to Table 7 since there is a similarity order matched with the input data \((T_4, 45), (T_9, 60), (T_{15}, 100), (T_{28}, 140), (T_{38}, 150)\). Based on the visiting sequences in \(RD_1\), a 40 by 40 previous tourist preference \(\text{PTE}\) matrix is constructed using Eqs. (7) to (11). Note that the time interval width \(IW\) in Eq. (9) is set as 30 min in this implementation. The number of tourists in the queue of rides \(r_i\), denoted as \(CQ(r_i)\), is generated by a data generation program to simulate the RFID system.

### 4.1. Route recommendation generation

The proposed route recommendation system is implemented in C++ with STL (Standard Template Library) and tested on a PC with Core 2 Duo 2.20 GHz CPU and 2 GB memory. Before executing the data preprocessing is conducted including a redundancy check and singleton check. Fortunately, no inappropriate sequence in the route database was detected. Therefore, after conducting the data preprocessing procedure, the route database remained the same.

In the tourist clustering module, all visiting sequences represented at the ride level are transformed into the sequences represented at the theme level based on the hierarchical structure in Table 4. To exclude meaningless theme experiences in a sequence, \(\rho_j\) is set as the average number of consecutive rides taken by all tourists in theme \(t_j\) and \(\psi_j\) is set as 80% of the average time length spent by all tourists in theme \(t_j\). According to the data in Table 5, the average number of consecutive rides taken by all tourists and the average time length spent by all tourists in each theme area are derived and shown in Table 6.

After the transformation process, 12 visiting sequences are eliminated and 42 sequences are left. The dissimilarity values between all pairs of visiting sequences represented at the theme level is then evaluated. The K-Medoids algorithm is applied to divide the 42 visiting sequences into \(K\) sub-groups. To obtain a meaningful result, the number of clusters \(K\) is set as 4 in this implemented, so that four sub-groups whose medoid is \(\text{cid} 4\) (called \(RD_1\)), \(\text{cid} 7\) (called \(RD_2\)), \(\text{cid} 17\) (called \(RD_3\)) and \(\text{cid} 54\) (called \(RD_4\)) are constructed as shown in Table 7.

We assumed that a visitor requests a route recommendation at booth location \(R_1\) at 10:30 am. He inputs the intended departure time as 9:00 pm, the set of wished visiting sequences at the theme level as \((T_4, 45), (T_9, 60), (T_{15}, 100), (T_{28}, 140)\), and the set of favorite rides as \(\{R_{38}, R_{24}, R_{15}, R_9, R_{28}\}\). Therefore, the \(RI\) vector is \(\{90, 720, \{T_4, 45\}, \{T_9, 60\}, \{T_{15}, 100\}, \{T_{28}, 140\}\}\). With the \(RI\) vector, the sub-group retrieval module identifies \(RD_1\) as the sub-group which is the most similar to the visitor’s personal preference. Since the dissimilarity value between the input data and \(RD_1\) is 0.491, \(RD_2\) is 1.000, \(RD_3\) is 0.519, and \(RD_4\) is 0.793. This result is expected according to Table 7 since the medoid of \(RD_1\) has three common themes \((T_4, T_9, T_{28})\) in the same appearance order matched with the input data \((T_4, 45), (T_9, 60), (T_{28}, 140)\).

Based on the visiting sequences in \(RD_1\), a 40 by 40 previous tourist preference \(\text{PTE}\) matrix is constructed using Eqs. (7) to (11). Note that the time interval width \(IW\) in Eq. (9) is set as 30 min in this implementation. The number of tourists in the queue of rides \(r_i\), denoted as \(CQ(r_i)\), is generated by a data generation program to simulate the RFID system.

![Fig. 10. The suggested route and visiting sequence of cid 16.](image-url)
system real-time data collection process. Therefore, the current ride preference (CRP) vector can be obtained. Based on the PTE matrix and the CRP vector, a recommendation matrix is constructed using Eq. (13).

According to the RL vector, the route generation algorithm recognizes that the time constraint (TC) is 630.00 min, the must-visit list (MtVist) is rides 38, 24, 13, 5 and 28, and the un-visit list (UnVist) is the remaining 35 rides. After running the algorithm, the first phase of the algorithm generates a primary visiting sequence as $R_1 \rightarrow R_5 \rightarrow R_{13} \rightarrow R_{24} \rightarrow R_{38} \rightarrow R_{38}$. Since the total visiting time ($TotVisT$) of the primary sequence (282.70 min) is less than TC, the algorithm is enforced to the second phrase. Finally, a route recommendation (RecRt) is generated as $R_1 \rightarrow R_3 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6 \rightarrow R_7 \rightarrow R_8 \rightarrow R_9 \rightarrow R_{10} \rightarrow R_{11} \rightarrow R_{12} \rightarrow R_{13} \rightarrow R_{24} \rightarrow R_{31} \rightarrow R_{32} \rightarrow R_{36} \rightarrow R_{38} \rightarrow R_{38}$. The RecRt suggests 21 rides which cross 7 themes, and its $TotVisT$ is 588.04 min which is less than but close to TC. It is clear that the RecRt not only meets the requirement of the visitor’s five wished thematic regions in the desired order ($T_2 \rightarrow T_3 \rightarrow T_5 \rightarrow T_6 \rightarrow T_7$) but also the visitor’s five favorite rides ($R_3$, $R_{13}$, $R_{24}$, $R_{38}$ and $R_{38}$).

4.2. System validation

It is difficult to compare the proposed route recommendation system with previous recommendation systems, since no route suggestion function is found in previous works. However, to show the benefit of the proposed system, the route suggestion generated by the proposed methods is compared with the original visiting sequence obtained from tourists. In this case, we compared the route suggested for the visitor in Section 4.1 with the $cid$ 16 visiting sequence obtained from the route database in Table 5, since the $cid$ 16 has the identical personal preference with the visitor discussed in Section 4.1 (actually the personal preference of the visitor is created by mimicking the visiting sequence of $cid$ 16). The route suggested for the visitor is visually displayed in Fig. 10 with a solid line, while the $cid$ 16 visiting sequence is displayed with a dashed line. Based on the figure it is clear that the route suggested for the visitor and the $cid$ 16 visiting sequence have the same visiting order at the theme level ($T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow T_6 \rightarrow T_7$). However, the route for the visitor suggests 21 rides while the original visiting sequence of $cid$ 16 contains only 17 rides. Four more rides were suggested for the visitor, the expected time to finish the suggested route is 630.00 min which is less than the total $cid$ 16 visiting time (658.21 min). The result shows that the proposed system can fulfill the visitor’s requirements and provide a richer visitation route.

4.3. The influence of personal preferences

Booth location (BL) where a visitor requests a route recommendation is an important factor for the proposed system. To know how BL affects the result, the booth location at rides 1, 4, 17, and 40 are tested while other settings are kept the same as the ones in Section 4.1. Fig. 11(a) to 11(d) visually illustrate the RecRts generated by the four booth locations $R_1$, $R_4$, $R_{17}$, and $R_{40}$, respectively. When BL is at $R_1$ of theme 1, $R_4$ of theme 2, $R_{17}$ of theme 5, and $R_{40}$ of theme 7, the number of suggested rides is 21, 20, 18, and 18 respectively. In addition, all the four RecRts cover 7 themes and their routes tend to follow a clockwise direction.

The intended departure time (LT) is another important factor that a visitor is concerned about. To know the influence of LT to the RecRt, we assume that the visitor intends to leave the theme park at 1:30 pm, 3:30 pm, 5:30 pm, and 9:00 pm. That is, the visitor wishes to spend 3, 5, and 7, 10.5 h to visit the theme park since $CT$ is 10:30 am. Therefore, LT is 270, 390, 510, and 720 while other settings are kept the same. Fig. 12(a) to 12(b) visually display the RecRts for the four cases. Fig. 12(a) indicates that 4 rides in 4 different themes are suggested ($R_1 \rightarrow R_5 \rightarrow R_{13} \rightarrow R_{24}$), since only 3 h is allowed in the theme park. For the rest of the three cases, all favorite rides provided

![Fig. 11. The route recommendations when BL is at $R_1$, $R_4$, $R_{17}$, and $R_{40}$.](image-url)
by the user \{R_{38}, R_{24}, R_{13}, R_5, R_{28}\} are in their RecRts. Fig. 12(b) indicates that \(R_1 \rightarrow R_6 \rightarrow R_{12} \rightarrow R_{24} \rightarrow R_{29} \rightarrow R_{28}\) is recommended if the visitor can spend 5 h in the theme park. Similarly, Fig. 12(c) and (d) shows that 13 rides and 21 rides are suggested when the visitor can spend 7 and 10.5 h respectively.

4.4. System parameter analysis

A set of system parameters might affect the performance of the proposed system. These parameters include the consecutive ride threshold \(\rho_j\), the minimum time threshold \(\phi_j\), the number of subgroups \(K\), and the time interval width \(IW\). Therefore, a set of experiments are conducted to observe the affection caused by these parameters. In the following discussion, the initial system parameter and visitor preference settings are summarized in Table 8.

4.4.1. The consecutive ride threshold

As discussed in Section 3.2.1, the transformation process transforms a visiting sequence represented at the ride level to the sequence represented at the theme level according to the four steps. In step two, if the number of rides consecutively taken within theme \(t_j\) is less than user-defined consecutive ride threshold, \(\rho_j\), those rides will be removed from the sequence since they are considered not significant enough to represent a visiting experience at the theme level. To observe how \(\rho_j\) impacts the system, \(\rho_j\) is set as the value ranging from 40% to 220% of \(\mu_{CR_j}\) where \(\mu_{CR_j}\) is the average number of rides consecutively taken by all customers in theme \(t_j\), while other settings remain the same as the ones in Table 8. Table 9 shows a set of \(\rho_j\) values when different \(\mu_{CR_j}\) values are applied. For example, when \(\rho_j\) is set as 180% of \(\mu_{CR_j}\), \(\rho_1 = 4\), \(\rho_2 = 5\), \(\rho_3 = 5\), \(\rho_4 = 4\), \(\rho_5 = 6\), \(\rho_6 = 6\), \(\rho_7 = 6\). That means that 4 consecutive rides are required to make the ride experience for theme 1 significant, 5 consecutive rides are required to make the ride experience for theme 2 significant, and so on.

Table 8
The initial settings for system parameters and visitor preferences.

<table>
<thead>
<tr>
<th>System parameters</th>
<th>Value</th>
<th>Personal preferences</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The average number of rides consecutively been taken in theme (t_j)</td>
<td>BL</td>
<td>(R_1)</td>
<td>90</td>
</tr>
<tr>
<td>The average time length spent for theme (t_j)</td>
<td>LT</td>
<td>(720) (</td>
<td>(T_{45}),(T_{60}), (T_{100}),(T_{140}), (T_{150})</td>
</tr>
<tr>
<td>(K)</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(IW)</td>
<td>30 (min.)</td>
<td>FR</td>
<td>{R_{28}, R_{24}, R_{13}, R_5, R_{28}}</td>
</tr>
</tbody>
</table>

Fig. 12. The route recommendations when the visitor spends 3, 5, 7, and 10.5 h.

Table 9
The corresponding \(\rho_j\) values.

<table>
<thead>
<tr>
<th>(\mu_{CR_j})</th>
<th>Theme</th>
</tr>
</thead>
<tbody>
<tr>
<td>40%</td>
<td>1111111</td>
</tr>
<tr>
<td>60%</td>
<td>1221222</td>
</tr>
<tr>
<td>80%</td>
<td>2222323</td>
</tr>
<tr>
<td>100%</td>
<td>2332334</td>
</tr>
<tr>
<td>120%</td>
<td>3332444</td>
</tr>
<tr>
<td>140%</td>
<td>3443445</td>
</tr>
<tr>
<td>160%</td>
<td>3443556</td>
</tr>
<tr>
<td>180%</td>
<td>4554666</td>
</tr>
<tr>
<td>200%</td>
<td>4554667</td>
</tr>
<tr>
<td>220%</td>
<td>5664778</td>
</tr>
</tbody>
</table>

Table 8
The initial settings for system parameters and visitor preferences.

<table>
<thead>
<tr>
<th>System parameters</th>
<th>Value</th>
<th>Personal preferences</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The average number of rides consecutively been taken in theme (t_j)</td>
<td>BL</td>
<td>(R_1)</td>
<td>90</td>
</tr>
<tr>
<td>The average time length spent for theme (t_j)</td>
<td>LT</td>
<td>(720) (</td>
<td>(T_{45}),(T_{60}), (T_{100}),(T_{140}), (T_{150})</td>
</tr>
<tr>
<td>(K)</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(IW)</td>
<td>30 (min.)</td>
<td>FR</td>
<td>{R_{28}, R_{24}, R_{13}, R_5, R_{28}}</td>
</tr>
</tbody>
</table>

Table 9
The corresponding \(\rho_j\) values.
does not have the screening capability for all themes, since all sequences survived. On the other hand, if $\phi_j$ is set at a higher value, the number of remaining sequences is fewer because many sequences are deleted. Fig. 14 shows the average number of themes among those sequences passing the screening test when different $\phi_j$ values are adopted. If $\phi_j$ is set as the value from 60% to 160% of $\mu_{MT,j}$, the average number of themes declines significantly from 2.27 to 1.22. Therefore, based on the observation from the two figures, this implementation suggests that $\phi_j$ is as 120% of $\mu_{MT,j}$.

4.4.2. The minimum time threshold

Similar to the consecutive rides threshold $\rho_j$ in step two of transformation process, the minimum time threshold, $\psi_j$, in step four is another critical threshold that should be decided. If the time length a visitor stays in theme $t_j$ is less than a user-defined $\psi_j$, the theme experience should be ignored and eliminated from the sequence. To observe how $\psi_j$ impacts the system, $\psi_j$ is set at a value ranging from 40% to 220% of $\mu_{MT,j}$ where $\mu_{MT,j}$ is the average time length spent by all customers in theme $t_j$ while other settings remain the same as the one in Table 8. Table 10 shows the relationship between $\psi_j$ and $\mu_{MT,j}$, where the unit of value in the table is minute.

Fig. 15 illustrates the number of sequences passing the screening test after conducting the transformation process. It is clear that the screening ability is very limited when $\psi_j$ ranges from 40% to 80% of $\mu_{MT,j}$. Conversely, no sequence passes the test if $\psi_j$ is set at a value greater than 180% of $\mu_{MT,j}$. Fig. 16 indicates the average number of rides in the RecRt generated by the route recommendation generation module after 30 experimental runs. The largest average number of rides is 20.37 when $\psi_j$ is set at 100% of $\mu_{MT,j}$. Based on the observation from the two figures, this implementation suggests that $\psi_j$ be set at 100% of $\mu_{MT,j}$.

4.4.3. The number of sub-groups

As mentioned in Section 3.2.3, the K-Medoids clustering algorithm was applied to cluster visiting sequences in RD. To observe how the number of sub-groups $K$ affects the system, this experiment changes $K$ from 2 to 5 while the other settings remain the same as the one in Table 8. The clustering results are summarized in Table 11. It is clear that the clustering result might be either too rough or trivial if $K$ is set at 2 or 5, respectively.

Fig. 17 illustrates that the average number of rides in the RecRt generated from the route recommendation generation module after 30 experiment runs. The largest average number of rides in the

Table 10

The corresponding $\psi_j$ values.

<table>
<thead>
<tr>
<th>$\psi_{MT,j}$</th>
<th>Theme 1</th>
<th>Theme 2</th>
<th>Theme 3</th>
<th>Theme 4</th>
<th>Theme 5</th>
<th>Theme 6</th>
<th>Theme 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>40%</td>
<td>20.0</td>
<td>20.3</td>
<td>27.3</td>
<td>20.1</td>
<td>32.6</td>
<td>27.6</td>
<td>44.9</td>
</tr>
<tr>
<td>60%</td>
<td>39.9</td>
<td>40.6</td>
<td>54.7</td>
<td>40.3</td>
<td>65.3</td>
<td>55.1</td>
<td>89.9</td>
</tr>
<tr>
<td>80%</td>
<td>59.9</td>
<td>60.9</td>
<td>82.0</td>
<td>60.4</td>
<td>97.9</td>
<td>82.7</td>
<td>134.8</td>
</tr>
<tr>
<td>100%</td>
<td>79.8</td>
<td>81.1</td>
<td>109.4</td>
<td>80.5</td>
<td>130.5</td>
<td>110.2</td>
<td>179.7</td>
</tr>
<tr>
<td>120%</td>
<td>99.8</td>
<td>101.4</td>
<td>136.7</td>
<td>100.7</td>
<td>163.2</td>
<td>137.8</td>
<td>224.7</td>
</tr>
<tr>
<td>140%</td>
<td>119.7</td>
<td>121.7</td>
<td>164.0</td>
<td>120.8</td>
<td>195.8</td>
<td>165.3</td>
<td>269.6</td>
</tr>
<tr>
<td>160%</td>
<td>139.7</td>
<td>142.0</td>
<td>191.4</td>
<td>140.9</td>
<td>228.4</td>
<td>192.9</td>
<td>314.6</td>
</tr>
<tr>
<td>180%</td>
<td>159.6</td>
<td>162.3</td>
<td>218.7</td>
<td>161.1</td>
<td>261.1</td>
<td>220.4</td>
<td>359.5</td>
</tr>
<tr>
<td>200%</td>
<td>159.6</td>
<td>162.3</td>
<td>218.7</td>
<td>161.1</td>
<td>261.1</td>
<td>220.4</td>
<td>359.5</td>
</tr>
<tr>
<td>220%</td>
<td>159.6</td>
<td>162.3</td>
<td>218.7</td>
<td>161.1</td>
<td>261.1</td>
<td>220.4</td>
<td>359.5</td>
</tr>
</tbody>
</table>

RecRt is obtained when $K$ is set as 4, while the smallest one is the case that no clustering is performed ($K = 0$). A one way ANOVA test was conducted to know whether the above five cases are significantly different in terms of the average number of recommended rides. Based on the result in Table 12, the difference between them is significant since p-value (0.000) is less than the significance level (0.05). A two-sample t-test was further conducted to test whether the difference between the clustering result with $K = 4$ and non-clustering was significant. The result indicates the two approaches are significantly different in term of the average number of recommended rides. Based on the above observation, this implementation suggests that the number of sub-groups $K$ is 4.

4.4.4. The time interval width

According to Eq. (9), different time interval widths $IW$ might affect the discretization result in the route recommendation generation module. To observe how $IW$ affects the recommendation result, $IW$ was set as a value ranging from 10 to 50 min while other settings remained the same as those shown in Table 8. Fig. 18 shows the relationship between $IW$ and the number of time intervals. When $IW$ is 1, the number of time intervals is 74 because the longest staying time at a ride is 73.7 min in RD. Fig. 19 shows the average number of recommended rides in the RecRt generated by the route recommendation generation module after 30 experimental runs. The figure indicates that if $IW$ is too large, the staying time at a ride tends to fall into a few time intervals. This makes the system hard to discriminate the time preference among sequences. Conversely, if $IW$ is too small, the discrimination for the time preference is too strong so that few
gested to collect tourist visiting behavior in the studied theme
is collected correctly and instantly. Therefore, a RFID system is sug-
congested route. Should lead visitors to escape the crowded area by providing a less
the same time other adjacent areas are vacant. Route suggestions
often in theme parks visitors congregate in certain areas while at
the same time other adjacent areas are vacant. Route suggestions
should lead visitors to escape the crowded area by providing a less

general behaviors are found. Therefore, this implementation suggests
that IW be set to 30 min because the setting generates the largest av-
average number of recommended rides.

5. Discussion

The major goal of this research is to provide tourists a visiting itin-
ery service that guides them completely through their trip. This
service is not found in previous tourism recommendation systems,
which focuses primarily on designing user friendly interfaces that
can smoothly interact with the environment, provide convenient
information query tools or suggest a set of associated products (or
services). However, helpful visiting sequence information such as
"you can follow the route C→B→A→D to complete your visit" is
critical to assisting tourists complete their trips instead of "you are
suggested to visit A, B, C, and D." Without a route suggestion, tourists
tend to have an inefficient trip or even get lost in the complex theme
park environment. Based on the above goal, the contribution of this
research is to generate a customized itinerary service based on tour-
ists’ personal needs and the visiting experiences of previous tourists.
First, most visitors do not have plenty of time to visit the whole theme
park and wish to finish their trip close to their given intended-visiting
time. Second, tourists usually have a set of "must-play" rides in mind
before starting their trips in the theme park. Third, instead of consid-
ering the route recommendation problem as an optimization problem
that minimizes the total visiting time, route suggestions should be
based on the visiting behaviors of previous tourists. Finally, quite
often in theme parks visitors congregate in certain areas while at
the same time other adjacent areas are vacant. Route suggestions
should lead visitors to escape the crowded area by providing a less
congested route.

This research is based on the assumption that the tourist location
is collected correctly and instantly. Therefore, a RFID system is sug-
gested to collect tourist visiting behavior in the studied theme
parks. When tourists enter a theme park, they are provided a

wristband embedded with a RFID tag. RFID readers installed in the en-
trance and exit of each recreation facility (ride) will record the tag
code and event time of each tourist into databases. With this recording
process, the visiting sequence of all tourists and the queue length of
each ride can be obtained in real time. However, when more people
have intelligent mobile phones, the proposed route recommendation
system can be packaged as an App and installed in tourists’ personal
mobile devices before they visit the parks. In the App, the routing algo-
rithm is executed on the server side while the route suggestion result
is shown on the client side. Moreover, if the mobile phones have NFC
(Near field communication) capability which is an extension of RFID
technology, the data collection process can be completed by tourists’
own mobile phones instead of using RFID wristbands.

6. Conclusions

Like any other industry, theme parks are now facing severe chal-
enges from other entertainment competitors. To survive in this rapidly
changing environment, creating high quality products/services in
terms of consumer preferences has become a critical issue for theme
park managers. Knowing which rides have been taken, which shows
have been attended (and how much they’ve been enjoyed) and
which shops and squares have attracted the attention of tourists,
could lead to a radical improvement in satisfaction performance. This
study proposed a route recommendation system that provides theme
park tourists with which facilities (rides) they should visit and in
what order.

When tourists enter a theme park, they are provided a wristband
embedded with a RFID tag with a unique electronic product code
(EPC). RFID readers are installed in the entrance and exit of each
recreation facility (ride). Whenever a tourist enters the navigating
area of a RFID reader, the reader will record the corresponding EPC
code and time and transfer that information to the route database.
In this way, tourist visiting behaviors (sequences) are continuously and
persistently collected through a RFID system. The Route Recommenda-
tion System consists of three major modules. The first module, the
route clustering module, segments all tourist visiting sequences in
the route database into sub-groups based on the dissimilarity among
tourist visiting sequences and time lengths. The second module, the
sub-group retrieval module, finds the sub-group which is the most
similar to the preference of a visitor’s input. The personal preference
includes the intended departure time, favorite thematic regions with

<table>
<thead>
<tr>
<th>K</th>
<th>Number of sequences and its medoid in sub-groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11 (cid 16), 23 (cid 50)</td>
</tr>
<tr>
<td>3</td>
<td>5 (cid 5), 8 (cid 54), 21 (cid 16)</td>
</tr>
<tr>
<td>4</td>
<td>5 (cid 7), 7 (cid 5), 8 (cid 16), 14 (cid 50)</td>
</tr>
<tr>
<td>5</td>
<td>5 (cid 10), 5 (cid 27), 6 (cid 6), 8 (cid 17), 10 (cid 11)</td>
</tr>
</tbody>
</table>

Fig. 17. The average number of rides in the RecRt under different K values.

<table>
<thead>
<tr>
<th>K</th>
<th>Number of sequences and its medoid in sub-groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11 (cid 16), 23 (cid 50)</td>
</tr>
<tr>
<td>3</td>
<td>5 (cid 5), 8 (cid 54), 21 (cid 16)</td>
</tr>
<tr>
<td>4</td>
<td>5 (cid 7), 7 (cid 5), 8 (cid 16), 14 (cid 50)</td>
</tr>
<tr>
<td>5</td>
<td>5 (cid 10), 5 (cid 27), 6 (cid 6), 8 (cid 17), 10 (cid 11)</td>
</tr>
</tbody>
</table>

Fig. 18. The relationship between IW and the number of time intervals.

Table 12

ANOVA analysis for the five cases.

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>4</td>
<td>33.893</td>
<td>8.473</td>
<td>16.43</td>
</tr>
<tr>
<td>Error</td>
<td>145</td>
<td>108.693</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>149</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F(4,145) = 2.3808, p < .05

Fig. 19. The average number of rides in the RecRt under different IW settings.
preferred order and wished staying time and favorite rides. The last module, the route generation module, takes the visiting behavior data identified in the second module and the queuing information from each ride identified by the RFID system to generate an appropriate visiting recommendation for the visitor.

There are some directions to improve the proposed system in the future. First, a set of system parameters including the consecutive ride threshold, the minimum time threshold, the number of subgroups, and the time interval width will affect the performance of the proposed system. Although their suitable ranges can be found through a set of experimental designs, it is a time-consuming task. It is suggested that researchers adopt optimization approaches to find the best values in these parameters. Second, to enrich the proposed route recommendation, factors such as personal spending habits and diet favorites can be taken into consideration when generating the visiting sequence recommendation. Finally, some theme parks might have multi-entrances and multi-exits. It would be interesting if more complicated layouts can be explored and studied.

Acknowledgement

This work was partially supported by the National Science Council of Taiwan, ROC, No. NSC 99-2221-E-155-055.

References


Chieh-Yuan Tsai is a professor in the Department of Industrial Engineering and Management at Yuan-Ze University, Taiwan. He received his M.S. and Ph.D. degrees in Department of Industrial and Manufacturing Systems Engineering from the University of Missouri-Columbia, U.S.A. His research activities include data mining, customer relationship management, e-commerce, and RFID applications.

Shang-Hsuan Chung is an Industrial Engineer at System and Engineering Management Department of ChipMOS Technologies Corp., Taiwan. She graduated with a master degree in Department of Industrial Engineering and Management, Yuan Ze University, Taiwan. Her research interests include data mining, RFID applications, and theme park management.