Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems | School of Computing and Information Systems

12-2023

ExploreLah: Personalised and smart trip planner for mobile tourism

Aldy GUNAWAN
Singapore Management University, aldygunawan@smu.edu.sg

Siu Loon HOE
Singapore Management University, slhoe@smu.edu.sg

Xun Yi LIM
Singapore Management University, xunyi.lim.2021@scis.smu.edu.sg

Linh Chi TRAN
Singapore Management University, lctran.2019@scis.smu.edu.sg

Dang Viet Anh NGUYEN
Singapore Management University, dvanguyen@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Artificial Intelligence and Robotics Commons, and the Tourism and Travel Commons

Citation

GUNAWAN, Aldy; HOE, Siu Loon; LIM, Xun Yi; TRAN, Linh Chi; and NGUYEN, Dang Viet Anh. ExploreLah: Personalised and smart trip planner for mobile tourism. (2023). *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) : Singapore, December 18-21: Proceedings*. 1-5.

Available at: https://ink.library.smu.edu.sg/sis_research/8490

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

ExploreLah: Personalised and Smart Trip Planner for Mobile Tourism

A. Gunawan, S. L. Hoe, X. Y. Lim, L. C. Tran, D. V. A. Nguyen
School of Computing and Information Systems, Singapore Management University, Singapore
(aldygunawan, slhoe, xunyi.lim.2021, lctran.2019, dvanguyen@smu.edu.sg)

Abstract - **Various recommender systems for mobile tourism have been developed over the years. However, most of these recommender systems tend to overwhelm users with too much information and may not be personalised to user preferences. In this paper, we introduce ExploreLah, a personalised and smart trip planner for exploring Point of Interests (POIs) in Singapore. The user preferences are categorised into five groups: shopping, art & culture, outdoor activity, adventure, and nightlife. The problem is considered as the Team Orienteering Problem with Time Windows. The algorithm is developed to generate itineraries. Simulated experiments using test cases were performed to evaluate and validate the usability of the current version.**

Keywords - **Algorithm, point of interests, recommender system, trip planner**

I. INTRODUCTION

The process of urbanisation has been gaining momentum worldwide in the last century. More and more people are permanently re-located to cities from the rural areas. With the rapid evolution of technology over the last twenty years, governments have invested in a lot of resources in developing city infrastructure and improving the environment and social welfare [1]. According to Silva et al. [2], such initiatives, coupled with the increase in the demand for sustainable living from a growing population, eventually led to the phenomenon now known as "smart city". Lee et al. [3] defined the characteristics of a smart city as the use of technologically advanced applications to ensure sustainability, economic innovations, and quality residential livelihood within the cosmopolitan area.

Smart city development includes the application of Internet of Things (IoT) in city management, which requires optimizing urban digital information, from the collection to the usage of such data, to the facilitation of city operations to enhance the lives of city dwellers while minimising human interactions [2]. IoT applications could range from smart residential houses and car parking facilities to traffic monitoring structures and weather controlling solutions [4]. These applications can respond to real-time demand changes, optimising public experience and aiding urban governance. The fast progression of IoT applications has encouraged governments to leverage on such technologies to optimize tourism policymaking and enhance tourists' experience visiting the country. Such changes shift the focus towards smart tourism, which refers to the development and management of destinations and tourists through digital transformation [5].

Since December 2019, the "novel" coronavirus named COVID-19 has spread globally and affected numerous industries, including causing major disruptions to the tourism sector [6]. Many countries have enforced travel restrictions over tourist arrivals to slowdown the spread of the virus [7], causing a 50% revenue loss of 2.86 trillion US dollars [6]. With the restrictions imposed on physical events, tourism-dependent countries relied heavily on smart tourism, with products featuring augmented or virtual reality [3]. In today's post-COVID-19 world where most borders have reopened, countries are now focusing on tourism recovery. In Singapore, the tourism sector has also digitally transformed to cope with the spread of the pandemic: from robot cleaners in Changi Airport to virtual walks in Mandai Wildlife Reserve to hologram broadcasts in Marina Bay Sands [8]. A study from Ramos et al. [9] analysed and highlighted the role of smart tourism in crowd management strategies to improve users' experience and aid policy-making processes, attracting more return visitors. In a smart tourism ecosystem, travel web applications play a critical role in assisting visitors choose certain attractions or Point of Interests (POIs) for trip planning.

The main motivation for the development of the personalised and smart trip planner arose from experiences faced when attempting to plan for full-day itineraries of places to visit. With the numerous attractions, there is a need to create personalised travel plans due to the limited time available and personal preferences for certain attractions. A popular approach in itinerary planning is to search from Google. The challenge faced by many is that the initial results always yield many different lists of attractions or travel websites with varying types of recommendations. Another common problem faced is that while one may usually end up with a list of seemingly interesting places to visit by browsing the multiple recommended websites, it is a tedious process when one tries to arrange for the order of visits. In this paper, we introduce ExploreLah, a smart tourism planner, that can help visitors create personalised visits in Singapore with less effort. Users can select their preferences in real-time based on specific categories: shopping, art & culture, outdoor activity, adventure and nightlife.

II. LITERATURE REVIEW

Recommender systems have become effective tools for filtering information on content or products due to major shifts in consumer habits and accessibility of Internet users [10]. The term was broadly defined by

SMU Classification: Restricted

Resnick and Varians in 1997 [11] as follows: *"In a typical recommender system, people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases, the primary transformation is in the aggregation; in others the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations".* Jalloulia, Lajmia and Amous [12] presented a conceptual framework for constructing and evaluating a recommender system using the example of LibSCars. Such frameworks for evaluation could be very useful in evaluating recommender systems across multiple domains, including e-learning [13], books [14], movie [15] and online shopping [16].

Specifically in the smart tourism domain, there have been many innovative applications of recommender systems in various countries such as Japan, and Belgium by Hidaka et al. [17], and Vansteenwegen et al. [18] respectively. The on-site recommender developed by Hidaka et al. [17] curated tourist preferences and current information available on the tourist site to cater to tourists who do not prefer detailed plans. However, currently, the system only recommended the next tourist spots, not a comprehensive sightseeing schedule. City Trip Planner is a web application that suggests routes within five Belgium cities, accounting for opening hours as well as user preference and limits to predict user interest [18]. However, changes to the recommended plans due to unforeseeable events are currently not supported by the platform. Migrating to a mobile version could be a possible solution, but mobile platforms generally have issues related to insufficient computational resources for the algorithm.

III. EXPLORELAH FRAMEWORK

In this section, we first depict the overall architecture of the ExploreLah system. We then describe the complete system flow, followed by the three layers involved.

A. System Architecture

As shown in Fig. 1, the application consists of three main layers: Data Layer (database), Application Layer (back-end) and Presentation Layer (front-end).

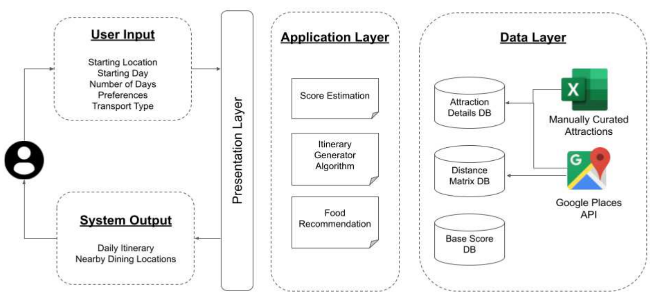

Fig. 1. System Architecture

B. System Flows

The whole process begins with the user providing the following inputs for his or her trip:

1) the starting Location
2) the starting day
3) the number of days
4) preferences for categories
5) transport mode

The starting location consists of any of the locations selected by the user as the starting point of the trip on a particular day. The starting day refers to the first day of the entire trip which is used in conjunction with the number of days. The number of days is capped at 7 days.

Preferences are based on five categories: shopping, arts & culture, outdoor activity, adventure and nightlife. The user is asked to rank from one to five, labelled as "No thank, I'll pass", "Rather do something else", "I wouldn't mind", "Would be nice", "Definitely a must!" respectively. Users are allowed to rank multiple categories at the same rank and by default they are all ranked 3 – "I wouldn't mind". The transport type has two options: public transport and private transport. This differentiates the time travelled between POIs. The recommender system will then generate the recommended POIs based on the user's inputs.

C. Data Layer

The database comprises of several POIs in Singapore. Their categories were tagged one of the following categories: shopping, arts & culture, outdoor, adventure and nightlife. Additional categories are added, hotels and food, or the starting locations and additional recommendations for nearby places for food. Google Places API is used to collect additional information for each place, such as opening hours, longitude and latitude, PlaceID, and unique ID from Google for future queries. Opening hours is used to determine if the attraction would be opened during the requested or selected days of the proposed trip. Longitude and latitude are used to determine the straight-line distance between POIs. Unique PlaceID is used as keys in both the distance-matrix and in a PlaceID: Attraction Name mapping. At each POI, we estimate a preset time spent at the attraction based on the assigned categories (Table I). The time is arbitrary and sourced via searches on specific locations and their Average Time Spent provided by Google and then aggregating them.

TABLE I
TIME SPENT FOR EACH CATEGORY

Category	Estimated Time (hrs)
Shopping	4
Nightlife	4
Arts & Culture	2
Outdoors	3
Adventure	2

Our database also stores the base score of each POI to facilitate calculations in the algorithm. Base score refers to the category matrix assigned to each attraction. This is a one-hot encoded matrix of where the position corresponds to categories. Additional information can also be encoded inside this base score. For instance, in this implementation, an additional entry is included in the base score to signify the presence of food at the location. This is later used to help recommend possible food places.

D. Presentation Layer

This is a client-facing web application that allows a user to input the required details as listed in Section III-B and a result page to display the generated itinerary. The form that captures the user input is modelled as a multi-step form with the following stages:

1) Starting Location, Auto-completed Text Input
2) Trip Dates, date picker
3) Categories, Slider
4) Transportation, two options

Fig. 2 illustrates the different stages of the multi-step form. The result page's main feature is the timeline illustrated in Figure 3. The timeline provides a clear itinerary for users to follow, with the visit order, start time and end times clearly listed. The vertical timeline can be easily viewed and saved with a mobile device, providing an easy reference on the go. There is also a map provided for each day where the pins are labelled in the order of which they are suggested in the itinerary. The algorithm also provides suggestions for meals near and around lunch and dinner time. This will be explained below. Several other details are present in the overall result page as illustrated in Figure 4. Additional features included in the result page pull data from external APIs. Some examples are weather forecast where applicable for the suggested time and date and pictures sourced from Google APIs. There is sufficient space to include other APIs to provide functions to improve the overall user experience as well.

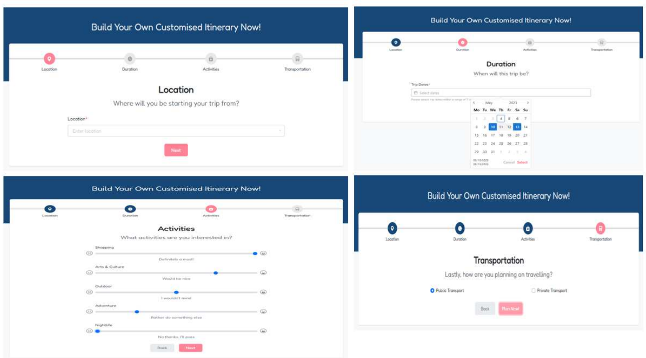

Fig. 2. Multi-Step Form

E. Application Layer

The tourist trip planning problem can be modeled as the well-known Team Orienteering Problem with Time Windows (TOPTW) [19]. TOPTW is defined as a directed graph $G = (V, E)$, where $V = \{0, ..., n + 1\}$ represents the set of Points of Interest (POIs), and $E = \{(i, j)|i, j \in V, i \neq j, i \neq n + 1, j \neq 0\}$ is the set of edges. Nodes 0 and (n + 1) denote the start and end points of all tours, respectively. Both can be the same or different nodes. Each POI $i \in V\backslash\{0, n + 1\}$ has a positive integer score p_i associated with it, and $p_0 = p_{n+1} = 0$. Each POI $i \in V\backslash\{0, n + 1\}$ has an opening hour l_i and a closing hour e_i, creating a time window $[l_i, e_i]$, while the time window for nodes 0 and (n + 1) is denoted as $[l_0, e_0]$. Let T be the maximum total travel time allowed to complete a tour in a day. The time to travel on an edge $(i, j) \in E$ is represented as t_{ij}, while the time to serve tourists at POI $i \in V\backslash\{0, n+1\}$ is denoted as s_i. The objective of TOPTW is to choose a subset of POIs that maximizes the total score while satisfying the conventional vehicle flow, time window, and service time constraints. The backend (application layer) consists of three modules: (1) Score Estimation, (2) Itinerary Generator Algorithm and (3) Food Recommendation. We will describe each module in more details below.

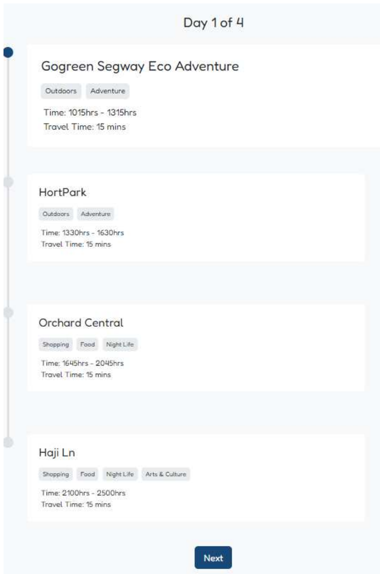

Fig. 3. Itinerary Timeline

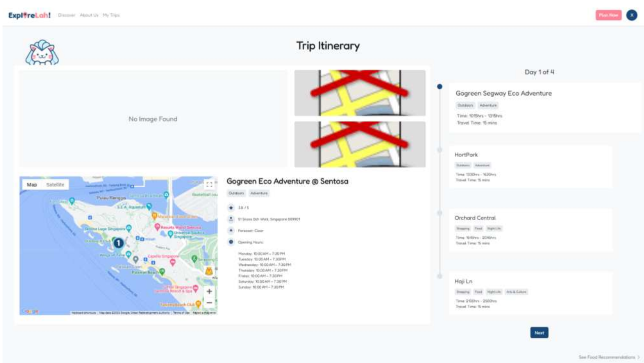

Fig. 4. Result Page

Score Estimation Module To personalise itinerary suggestions in the application and capture the unique

SMU Classification: Restricted

preferences of each user, we use a method to determine the score, p_j, of each attraction $j \in V$ as follows:

$$p_i = \frac{u_i}{d_{ij}} \quad (1)$$

The unique attraction score u_j is determined based on the customer's preferences and the attraction categories, as shown in the following equation:

$$u_j = \sum_{k \in K} x_j^k r_k \quad (2)$$

The set of five attraction categories including shopping, arts & culture, outdoors, adventure, and nightlife is denoted by K. The binary variable x^k_j equals 1 if the attraction j is classified into group $k \in K$. The user preference rating r_k is randomly selected based on the category and a rating scale ranging from 1 (least preferred) to 5 (most preferred). The range of r_k is specified in Table II, and we normalize r_k by scaling it between 0 and 1. It is worth noting that one attraction can be classified into different groups in K.

TABLE II
RANKING AND MULTIPLER

Ranking	Multiplier
5 (Highest)	80-100
4	70-90
3	50-75
2	30-55
1 (Lowest)	20-35

d_{ij} is the distance between the considered attraction j and the previous visited attraction i, which can be identified based on their latitude and longitude coordinates obtained from Google API using a Haversine formula. This method of estimating the score p_j for each attraction has the advantage of avoiding homogeneous suggestions for users, which can result in a monotonous and unappealing itinerary. By using this score estimation, we can maintain the user's preferences while ensuring diverse attraction suggestions.

Itinerary Generator Algorithm Module

Algorithm 1 outlines the itinerary generation process for each day n within travel period N. The algorithm calculates the unique attraction score u_i of each attraction in attraction set V (Lines 4 - 7). Next, the attractions in V are sorted in descending order of their unique attraction score u (Line 8). Then, top 20 attractions with the highest unique attraction scores are selected and stored in set Q (Line 9). The score p_i of each attraction in set Q is then calculated and the algorithm sorts the attractions in Q in descending order of score (Lines 12 - 16). The itinerary is created by randomly selecting attractions from top 5 unvisited attractions with the highest scores in Q. The algorithm checks if the selected attraction j is open upon arrival, then adding them to the itinerary until the maximum total travel time T is reached (Lines 11 - 29). The algorithm repeats this process for all days in n and terminates by returning the itinerary set I (Line 32).

Algorithm 1 Itinerary generator algorithm

1: **Input:** The attractions set V
2: **for** day n in period N **do**
3: Accumulated travel time, c
4: **for** i in V **do**
5: Calculate the unique attraction score, u_i
6: $U \leftarrow u_i$
7: **end for**
8: Sort set V in descending order of u
9: Select top 20 attractions in V then store in Q
10: Starting point, $i = 0$
11: **while** $c \leq T$ **do**
12: **for** j in Q **do**
13: Calculate the score p_j of attraction j
14: Score set $P \leftarrow p_j$
15: **end for**
16: Sort set Q in descending order of p
17: Q^m, top 5 attractions in Q
18: Randomly select an attraction $j \in Q^m$
19: **if** $c + t_{ij} < e_i$ **then**
20: **if** $c + t_{ij} \leq T$ **then**
21: Add attraction j to itinerary of day n, I^n
22: Remove attraction j from Q
23: $c = c + t_{ij}$
24: $i = j$
25: **else**
26: Break
27: **end if**
28: **end if**
29: **end while**
30: **end for**
31: Itinerary set $I \leftarrow I^n$
32: **Return:** Itinerary set, I

Food Recommendation Module

When following an itinerary, a point of concern for the user would be the availability and option for food nearby. We include this as a suggestion during lunch or dinner times. These results are then presented to the user in the form of an additional tab in the result page as shown in Figure 5. This approach allowed the focus of the generated itinerary to be on suggesting attractions as well as giving assurance and flexibility to the user.

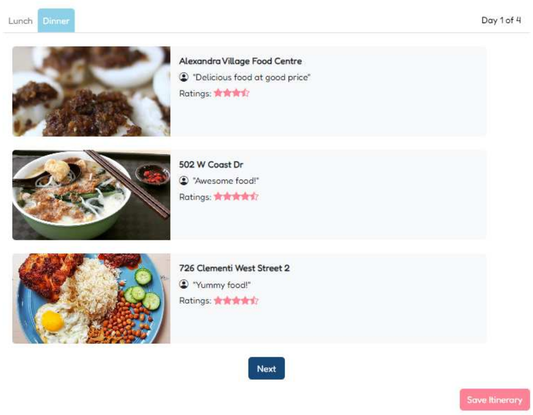

Fig. 5. Food Recommendation

IV. EXPERIMENTS

At the current stage, we are still working on the actual implementation of ExploreLah. Once we obtain the complete user study results, we will be reporting on the findings. Various cases have been conducted. However, due to the space limit, we only present three different scenarios with user preferences, as listed below: Case 1: shopping and nightlife for 4-days itinerary with the public transport; Case 2: outdoor, adventure and shopping for 4-days itinerary with the public transport; Case 3: art & culture for 2-days itinerary with the public transport. Table III summarizes the generated itineraries. The results look promising where the POIs are matched with the user's preferences based on the selected categories.

TABLE III
THE GENERATED ITINERARIES

Cases	Days	Itineraries
Case 1	Day 1	Hotel A→ Mustafa Center → Haji Ln → Bugis Street
	Day 2	Hotel A → Paragon Shopping Centre → Lucky Plaza → Far East Plaza
	Day 3	Hotel A → Raffles City → Orchard Central → SCAPE
	Day 4	Hotel A → Little India → City Plaza → Robertson Quay
Case 2	Day 1	Hotel B → Haji Ln → Mustafa Centre → Sentosa
	Day 2	Hotel B → Gogreen Segway Eco Adventure → Mega Adventure → Night Safari
	Day 3	Hotel B → ION Orchard → Hort Park → Mac Ritchie Reservoir
	Day 4	Hotel B → Raffles City → Super Park → SCAPE → Universal Studio
Case 3	Day 1	Hotel C → Mustafa Centre → Little India → Raffles City
	Day 2	Hotel C → Bugis Street → Paragon Shopping Centre → Far East Plaza

V. CONCLUSION

We introduced ExploreLah, a smart tourism planner framework designed specifically for the Singapore context. It leverages information about points of interest and user preferences to help create personalised tourist itineraries.

Through various test cases, we have validated the functionality of ExploreLah. However, we acknowledge some limitations of the algorithm. These include not accounting for serving time at POIs and having multiple trips on the same day. Future work will focus on addressing these limitations to improve the overall effectiveness of ExploreLah. It would also be interesting to compare the proposed algorithm to existing algorithms from the literature.

ACKNOWLEDGMENT

We would like to thank Prof. Mok H.N. for his feedback and the group members who worked on the web app: Benji, Desiree, Chulin & Ciara.

REFERENCES

[1] E. Trindade, M. Hinnig, E. Costa, J. Marques, R. Bastos, and T. Yigitcanlar, "Sustainable development of smart cities: A systematic review of the literature," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 3, no. 3, pp. 1-14, 2017.

[2] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697–713, 2018.

[3] P. Lee, W. C. Hunter, and N. Chung, "Smart tourism city: Developments and transformations," *Sustainability*, vol.12, no. 10, 2020.

[4] Y. Kaluarachchi, "Implementing data-driven smart city applications for future cities," *Smart Cities*, vol. 5, no. 2, pp. 455–474, 2022.

[5] B. H. Ye, H. Ye, and R. Law, "Systematic review of smart tourism research," *Sustainability*, vol. 12, no. 8, 2020.

[6] J. Abbas, R. Mubeen, P. T. Iorember, S. Raza, and G. Mamirkulova, "Exploring the impact of covid-19 on tourism: transformational potential and implications for a sustainable recovery of the travel and leisure industry," *Current Research in Behavioral Sciences*, vol. 2, pp. 100033, 2021.

[7] M. Rahman, Md. A. Gazi, M. Bhuiyan, and Md A. Rahaman, "Effect of covid-19 pandemic on tourist travel risk and management perceptions," *PLoS ONE*, 2021.

[8] Singapore Tourism Board. "Smart tech for seamless & contactless travel," n.d.

[9] V. Ramos, W. Yamaka, B. Alorda, and S. Sriboonchitta, "High-frequency forecasting from mobile devices' bigdata: an application to tourism destinations' crowdedness," *International Journal of Contemporary Hospitality Management*, vol. 33, pp. 1977–2000, 2021.

[10] D. Roy, and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, 2022.

[11] P. Resnick, and H. R. Varian, "Recommender systems," *Communication ACM*, vol. 40, pp. 56–58, 1997.

[12] M. Jallouli, S. Lajmi, and I. Amous. "Designing recommender system: Conceptual framework and practical implementation," *Procedia Computer Science*, vol. 112, pp. 1701–1710, 2017.

[13] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowledge-Based Systems*, vol. 22, pp. 261–265, 2009.

[14] R. J. Mooney, and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. 5th Conf. ACM on Digital Libraries*, *DL'00*, New York, NY, USA, pp. 195–204.

[15] P. Winoto, and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6086–6092, 2010.

[16] M. Dabrowski, and T. Acton, "The performance of recommender systems in online shopping: A user-centric study," *Expert Systems with Applications*, vol. 40, pp. 5551–5562, Oct. 2013.

[17] M. Hidaka, Y. Kanaya, S. Kawanaka, Y. Matsuda, Y. Nakamura, H. Suwa, M. Fujimoto, Y. Arakawa, and K. Yasumoto, "On-site trip planning support system based on dynamic information on tourism spots," *Smart Cities*, vol. 3, no. 2, pp. 212–231, 2020.

[18] P. Vansteenwegen, W. Souffriau, and G. V. Berghe, "The city trip planner: An expert system for tourists," *Expert Syst. Appl.*, vol. 38, pp. 6540–6546, Jun. 2011.

[19] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "Iterated local search for the team orienteering problem with time windows," *Computers & Operations Research*, vol. 36, no. 12, pp. 3281–3290, 2009.