Expert Systems With Applications 181 (2021) 115026

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new point-of-interest approach based on multi-itinerary recommendation engine

Joy Lal Sarkar *, Abhishek Majumder

Mobile Computing Laboratory, Department of Computer Science and Engineering, Tripura University, Tripura, India

ARTICLE INFO

Keywords:
Recommendation system
Tourist
Cost
Itineraries
POIs

ABSTRACT

The significance of tourism in the globe today is enormous since it is a major source of income and jobs for a nation. Tourists are facing a range of difficulties as they select suitable tours, consisting of several itineraries in terms of their interests and distinct constraints. An itinerary consists of many Points of Interest (POIs) and a POI can further be splitted into several attractions which are named as POI within POI. For selecting the itinerary, the existing techniques use the characteristics of POIs. However, a POI consists of many attractions. Out of these, one dominating attraction's type is considered as POI type. This ignores the other type of attraction's present in that POI. It may cause improper selection of itineraries. Therefore, selection of itineraries by considering POI within POI is of great benefit. But, it is very challenging. For this task, we suggest an algorithm called PWP. It recommends multiple itineraries that are based on the interest of visitors, popularity of itineraries and the cost of itineraries. If a tourist wants to visit unknown areas, the PWP algorithm can be expanded further. We have taken the similar user's features to advise multiple itineraries using the Flickr dataset. The findings show that the proposed PWP algorithm out-performs the baseline algorithms in terms of real-life matrices and heuristic based metrics.

1. Introduction

It is a difficult task to recommend a tour and a tourist itinerary (Li, Lee, & Yang, 2019). A tourist has his/ her limitations and favorite locations (Trachanatzi, Rigakis, Marinaki, & Marinakis, 2020). There are countless Internet resources that provide the tourists a lot of information, without taking into account their interests and limitations (Lim, 2015). The data provided on the Internet can easily confuse a tourist during selection of best itineraries (Lim, Chan, Karunasekera, & Leckie, 2017). If a visitor plans his/ her trip, he/ she may spend several days on several attractions. In such a situation, he/she would prefer to go for several itineraries instead of a single itinerary. An itinerary may consist of multiple Points of Interest (POIs) (Qiao, Luo, Li, Tian, & Ma, 2020; Expósito, Mancini, Brito, & Moreno, 2019). The POIs where tourists are more interested need to be chosen in multiple itineraries. Many attractions may be present in a POI. Among all the attractions, the most prominent one dominates other attractions. The POI's type will be same as the most prominent attraction's type. The other attractions' type present in the POI gets ignored. Therefore, during selection of POI, the attractions within it should play very significant role (see Fig. 1, 2). The key difference between the current POI technique and the proposed technique is that the existing techniques recommend POI based on the original form for which it is known, but the proposed technique considers POI-attractions which are also present inside this actual POI. In this work, these attractions are considered as standard POIs. For convenience, we write these POIs as POI-attractions. A tourist may be interested to visit a park or sea beach but has little interest in the temple. Let, the tourist visits a beach or park which is adjacent to a prominent temple. Because of the presence of the prominent temple, the whole POI was marked as temple. Considering the category of POI (temple) already visited by the tourist, the existing works try to recommend itineraries containing POIs of type temple during his/her visits to other places. But while recommending itineraries, it will not consider POIs marked as beach or park, where the tourist is actually interested. This is because, these techniques do not access the POI-attractions inside a POI. Therefore, an appropriate technique is necessary to recommend itineraries considering the POI-attractions present inside a POI. The aim of the paper is to propose a technique which constructs multiple itineraries considering POI-attractions. For example, a POI St. Pete Beach, Florida can be classified as a beach category (Fig. 1), but there are so many POIs

* Corresponding author at: Department of Computer Science & Engineering, Tripura University, Tripura, India.
E-mail address: joylalsarkar@gmail.com (J.L. Sarkar).

https://doi.org/10.1016/j.eswa.2021.115026
Received 25 November 2020; Received in revised form 27 February 2021; Accepted 8 April 2021
Available online 16 April 2021

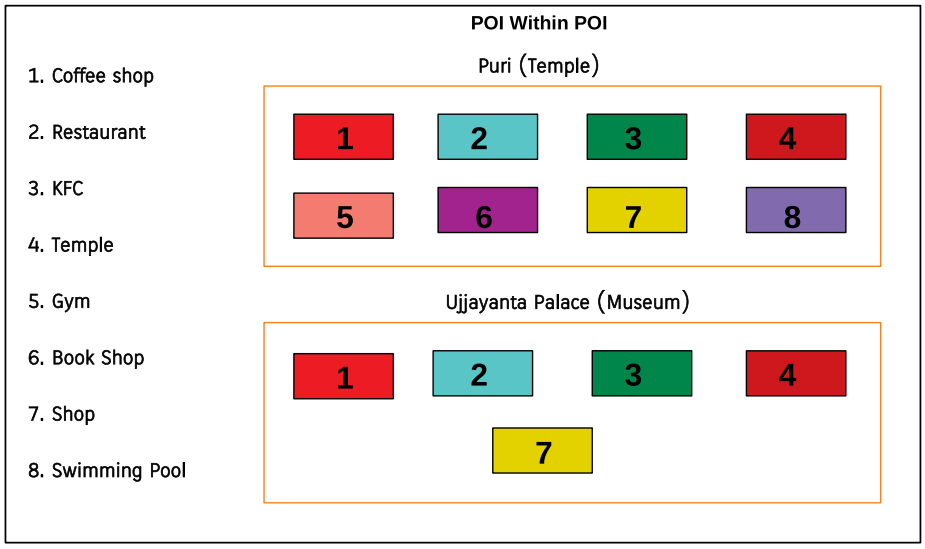

Fig. 1. Example of multiple itineraries recommendation problem with POI-attractions.

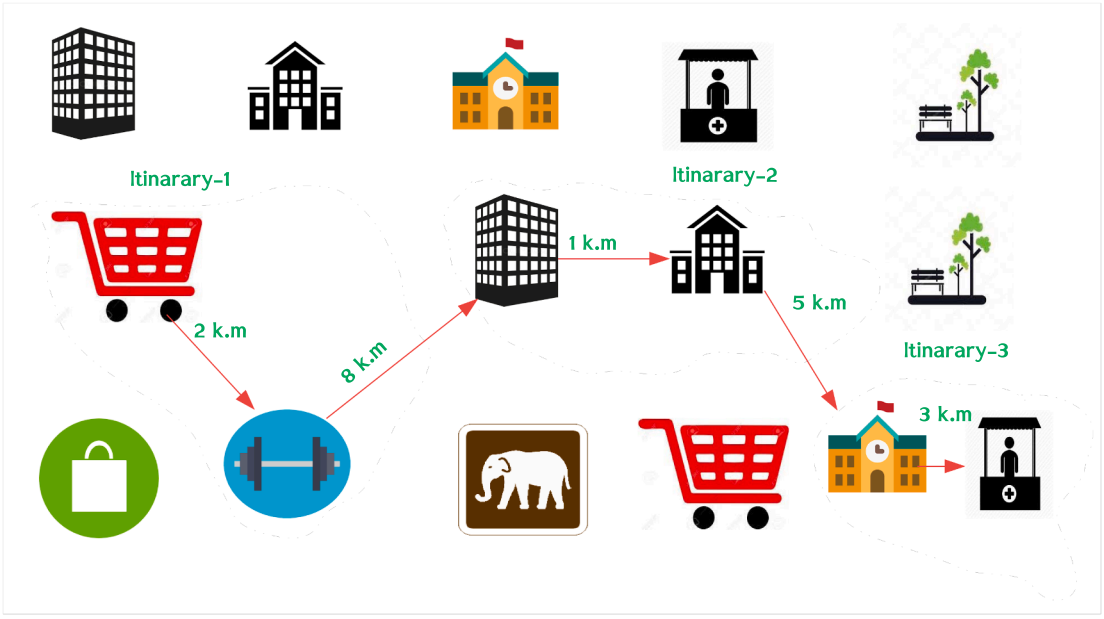

Fig. 2. Example of multiple itineraries recommendation problem with cost between POI-attractions. An itinerary consist of multiple POI-attractions. The number shows the distance between two POI-attractions.

available in St. Pete Beach, such as Cafe, Restaurants, etc. Let, some user who is not interested in beach, visits a restaurant in St. Pete Beach. When the user goes to some other place, the existing techniques will not recommend POIs of type restaurant while recommending itineraries to him/her. Instead, these techniques will recommend POIs of type beach. The proposed technique considers the POI-attractions from different POIs and recommends itineraries based on those POI-attractions.

In this paper, we propose an algorithm called PWP based on a multi-itinerary recommendation engine. It has been designed to recommend multiple itineraries in an efficient and user-friendly way for the total duration of the journey, depending on interest, tour popularity and the cost of travel. The system framework for the proposed approach is outlined in Fig. 3. The details such as POI visit durations, travel sequences, number of POI visits etc. about the Local User (*LU*) and Global Users (GUs) are taken and stored in the local and global user's repository. For each category, the interest of *LU* is taken into account in the calculation of the similarity between local and global users. The *Cosine- Similarity* test indicates how close two users are with respect to their preferences of interest. Hence, the *Cosine- Similarity* test is carried out for identifying a particular *LU*'s similar GUs. Then the itineraries where global users have visited is found out. The interest, popularity and cost of these itineraries are calculated. Ultimately, itineraries are suggested using the NSGA-II based PWP approach.

The key characteristics of PWP algorithm are as follows:

- In our previous work (Sarkar, Majumder, Panigrahi, & Roy, 2020), we proposed an approach where multiple itineraries were suggested. To the best of our knowledge, the PWP approach is the first of its kind to suggest multiple itineraries that take into account POI-attractions which are not directly accessible by existing POI techniques. The main difference between our previous work and the proposed PWP is that the MULTITOUR works well when multiple itineraries are

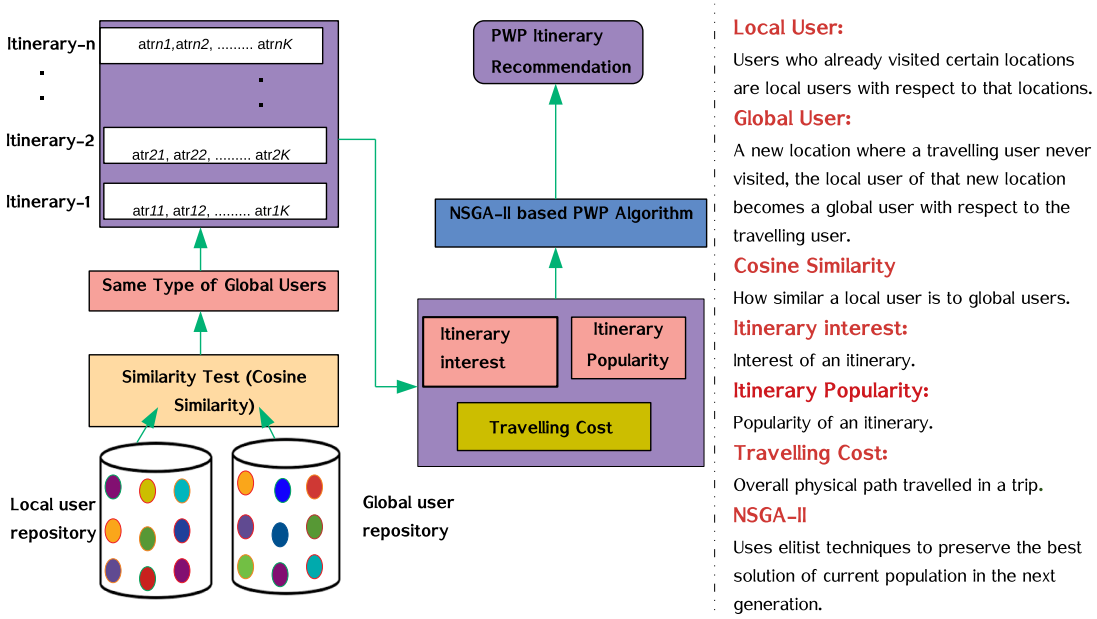

Fig. 3. System framework.

suggested considering POIs. On the other hand, PWP also offers various itineraries, but takes into account the attractions available within a POI.

- The PWP algorithm allows visitors to visit new POIs where visitors have not been visited before.
- The suggested framework extracts the amount of interest of visitors, POI-attraction popularity and distance associated with geo-tagged pictures.
- The PWP algorithm uses the NSGA-II algorithm to recommend several itineraries.
- The proposed scheme maximizes the popularity of tourist attractions, interest of the tourist and minimizes the costs through distance constraint.
- The time oriented concern of tourists tends to help tourists to obtain new itineraries if the previous record is unavailable.
- Flickr's dataset is utilized for evaluation of PWP method with different baseline approaches in various cities. The results show that when compared to the baselines, the PWP algorithm outperforms in terms of the *Precision*, *Recall* and *F1-Score*, tour popularity, accuracy, interest of the tourist and the amount of recommendable itineraries.

An itinerary consist of multiple POI-attractions. The number shows the distance between two POI-attractions.

2. Related work

Recently, the Tourism Recommendation System (TRS) is an important area of research (Lim et al., 2017). Numerous apps were created to offer beautiful, peaceful (Quercia, Schifanella, & Aiello, 2014) and happy tours (Lucchese, Perego, Silvestri, Vahabi, & Venturini, 2012).

2.1. Orienteering problem

In the case of Orienteering Problem (OP) (Lim, 2015; Lim et al., 2017), different checkpoints with corresponding scores have been located in different locations. All the participants aim to improve their total score within a short time frame by visiting the various checkpoints. The most scoring participant is the winner. The main point is to achieve the highest ranking by taking a small period of time.

2.2. TRS using the OP

De Choudhury et al. (2010) suggested a tour recommendation using the OPs to increase the rating of visits, which beginning with a unique POI and finishing with another POI. A plan was suggested by Vansteenwegen, Souffriau, Berghe, and Oudheusden (2011) for modifying the recommendation for tourists to best match the above-mentioned tourist interest rate with its limitations such as budget. Lim (2015) changed the orienteering issue in relation to the main POI category based on interest of the tourist. Lim et al. (2017) found the attractions to optimize interest of the tourist and tour popularity in form of the lowest queuing period. In order to recommend tour itineraries which reach different levels of tourism, the issue of orienteering has been expanded (Anagnostopoulos, Atassi, Becchetti, Fazzone, & Silvestri, 2017).

2.3. Top-k POI recommendation and next-location prediction

Itinerary recommendation are linked with next POI's prediction and top-k POIs recommendation (Chen, Zhang, Cao, Wu, & Cao, 2020). In LearNext, tourists' future POI visits are predicted using SVM algorithms and Gradient Boosted regression trees (Baraglia, Muntean, Nardini, & Silvestri, 2013). The *Markov method* is used for the estimation of next POI and seasonal POI data (Yamasaki, Gallagher, & Chen, 2013). The regularization of the class with the matrix factorizing principle recommends individual POIs (Shi, Serdyukov, Hanjalic, & Larson, 2011).

2.4. Different TRS related works

For the upcoming improvements, the authors shared the participation of the base travel recommendation model (Dietz & Weimert, 2018). Tourists can store, manage and communicate their trip information using *wondary* as a base. Wondary supports the planning and sharing of a global journey based on crowdsourcing. Individual and group itineraries are effectively framed and associated using *wondary* platform which inspires the other tourists to utilize and modify their own itineraries as per their interests (Dietz & Weimert, 2018). The delightful routes were grouped together with POI and recommended by Tourist Trip Recommender Systems (TTRSs) (Herzog, Promponas-Kefalas, & Wörndl, 2018). To enhance the tourists exposure of the TTRS and reduce the mobile device restrictions like limited display, the recommendation

system details were combined and exhibited into the publicly available manner but, this process is normally unusual. This work presents, a Distributed User Interface (DUI) concept which displays the RS information in public as well as individual end points. For a batch of tourists recommendation, public display is an impressive one and is discussed as an incorporation of group RS with the proposed model. Even, incorporation of public display with TTRS has numerous benefits over mobile display, but it has privacy issues still. DUI model permits the user to keep their personal data into a confidential source during the use of public display. Many preventative concepts were implemented to avoid shoulder-surfing (Brudy, Ledo, Greenberg, & Butz, 2014). The author planned to evaluate the batch recommendation with TTRS in his future work. The main objective of TTRS is to advise the suitable details to users to take necessary decisions based on the accessible data.

Vu, Li, and Law (2019) proposed a method for analysis of the tourist's travel itinerary using the probabilistic topic modeling. The approach will indirectly expose the desires of tourist in complex travel pattern. Zheng, Liao, and Lin (2020) proposed a custom day itinerary model taking the transport mode into consideration. A non-dominated sorting heuristic method was developed with improved particle swarm optimization technique and differential evolution algorithm. Kotiloglu, Lappas, Pelechrinis, and Repoussis (2017) used Collaborative filtering technique to pick attractions on the basis of customer preferences, web rating and check-in number. The aim is to create tours from all necessary points and increase the cumulative collected score from the optional points visited each day. The user interests were analyzed from past browsing, buying and social media data (Chen, Yu, Tang, He, & Zeng, 2017; Seo & Cho, 2021). The choosing of flight itinerary along with TTRS has miner famous compared to online e-commerce strategy and which was partly described. Machine and Deep Learning approaches helped to track this problem with the help of Choice Modeling (CM) concepts (Mottini, Lheritier, Acuna-Agost, & Zuluaga, 2018). Mottini et al. (2018) have shown that, the TTRS and CM helped to solve the flight itinerary search issues. This approach acts as a primary benchmark among three kinds of CM models to find the utmost relevant solutions to this problem. Kapcak et al. (2018) proposed a *TourExplain* model to describe each user's choice of itineraries to choose the same POIs from all the users to visit. This model was also extended to various domains. But, each domain has specific limitations of real-time descriptions. In this work, for tourism domain, Group dynamics and Interaction Design were discussed to satisfy the individual and batch users' interests. In the early stage, for tourism domain, Location-based Social Network (LBSN) data helped to recommend the POIs. Many POIs based itinerary recommendation is a tedious task for batch of tourists.

The batch recommendation may deviate user interests, but it is necessary to fascinate each user in the batch. Palumbo, Rizzo, Troncy, and Baralis (2017) had used Foursquare data set to extract similar real-time itineraries and predict new itineraries by using Recurrent Neural Network (RNN) method. Also the data was scattered based on tourist demographics and identified a specific approach for every group of tourists. The efficiency of the proposed model was evaluated based on identifying the itineraries on behalf of test set complexity. The work proposed a model using RNN for naturalistic understanding of the personalized itineraries from the data and to identify the POI order for the future. The author examined with various criteria to understand the usefulness of learning rate of the multiple layers compared to invisible layers. To enhance the model efficiency, the author used *node2vec* encoding approach rather than one-hot approach based on the model complexity and execution time. Based on the model complexity, the bypass elements were recommended by the authors. The TRS helped the tourists to quickly choose the necessary POIs and its effective order to visit within the budget. Lee, Chung, and Nam (2019) discussed how the online intelligent tourism data can be helpful for the tourists as well as South Korea's tourism business. Based on big network data analytics, this study concluded that the offline aspirant community had more tourism information rather than online community.

The existing techniques suggest an itinerary based on the actual form of POI where as the proposed technique considers the POI-attractions which are present inside a POI. Generally, a visitor may not be interested in all the POI-attractions within a POI. Under these conditions, it is important to suggest POI-attractions that are of greater interest to the visitor. For example, tourists may only be interested in visiting the sea beach in the city of Puri in India. Nevertheless, as the famous Jagannath Temple is situated in Puri, Puri is listed under the category of temple. In addition, there are several types of POI-attractions such as restaurants, temples, parks, museums, etc. but a visitor may not be interested in all these POIs. In this work, a multiple itinerary tourist recommendation technique is developed by considering the POI-attractions which are present inside a POI.

Table 1
Notations and their descriptions.

Notations	*Descriptions*
B	$TotalBudget$
$inte_{u_x}$	$Interestoftouristu_x$
$inte_{u_y}$	$Interestoftouristu_y$
$Pop(i)$	$Popularityofitineraryi$
$matching\_user\_list$	$matchingbetweenthelocaland\ globalusers$
Θ	$weightvalue, \Theta \in [0,1]$
$\mathscr{T}^{cost}(x)$	$Travellingcost$
Π^{intr}	$Internaldistanceofanitineraryi_y$
Π^{extr}	$Externaldistancebetweentwoconsecutiveitineraries$
P^{size}	$Populationsize$
$\mathscr{G}^{no}$	$Generationnumber$
$\mathscr{I}_l$	$Itinerarylist$
R	$Ratioofinterestorpopularitywithrespecttocost$
F_1, F_2	$Listtostorefitness-1andfitness-2values$
$\mathscr{N}_d$	$Listtostorenon-determinantsortingvalues$
$\mathscr{C}_{crowd}$	$ListtostoreCrowdingdistancevalues$
$S^{sol}, \mathscr{I}_1^{sol}, \mathscr{I}_{new}^{sol}$	$Liststostoredifferentsolutions$

3. Problem definition

Let, $\mathscr{I} = \{i_1, i_2, i_3, \ldots i_n\}$ be the set of itineraries in a specific city. Each itinerary $i_\vartheta \in \mathscr{I}$ $(1 \leqslant \vartheta \leqslant n)$ consists of a sequence of POI-attractions $\{\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3, \ldots \mathbb{P}_k\}$. The respective *POI-attraction* are marked with a category $\mathbb{c}$ which may have distinct criteria such as sea beach, sport, temple, etc. In this work, each suggested itinerary comprises of several *POI-attractions* gathered from the distinct POIs accessible in this itinerary, depending on the tourist interest, *POI-attraction* popularity and the related traveling costs. Therefore, $i_\vartheta = \{\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3, \ldots \mathbb{P}_{\kappa'}\}$, here, $\kappa' =$ itinerary i_ϑ's size. The complete distance traveled by a user for a specific itinerary is calculated by adding the distance between $\mathbb{P}_\gamma$and $\mathbb{P}_{\gamma+1}$, here $1 \leqslant \gamma < \kappa'$. The gap between i_1 and i_2 is measured by sum of the distance between the itinerary i_1's last *POI-attraction* and itinerary i_2's first *POI-attraction*. We have taken 4 km per hour of travel velocity, as we have seen (Lim et al., 2016).

3.1. Local and global users characteristics

Users who already visited certain locations are LUs with respect to that locations. A new location in which a travelling user did not previously visit, the LU of that new location becomes a GU with relative to the travelling user. Let, traveller A and B has already visited Delhi and Edinburgh respectively. Therefore, A is LU for Delhi and B is local user for Edinburgh. When user A wants to visit Edinburgh, B becomes his GU. In other words, when user A wants to fly Edinburgh, user B from LU will become a GU of A (Sarkar et al., 2020) (see Table 1).

3.2. Average POI-attraction visit duration for LUs and GUs

The history of tour is known among all users u. The visit time can be determined with Eqn. 1 at a specific *POI-attraction*:

$$\mathscr{D}(\mathbb{P}) = \frac{\sum_{u=1}^{k}\sum_{j=1}^{\ell}(t_j^{dept} - t_j^{ari})\delta(\mathbb{P}_j = \mathbb{P})}{\sum_{u=1}^{\kappa'}\mathscr{V}_u\delta(\mathbb{P}_j = \mathbb{P})} \tag{1}$$

where, $j = \{1, 2, \ldots, \ell\}, u = \{1, 2, \ldots, k\}$ and $\mathscr{V}$ indicates the amount of visits to a specific *POI-attractions* by a visitor and $\delta(\mathbb{P}_j = \mathbb{P}) = \begin{cases} 1 & \text{if } (\mathbb{P}_j = \mathbb{P}) \\ 0 & \text{otherwise} \end{cases}$. $\mathscr{D}(\mathbb{P})$ is used to denote average visit duration to a specific *POI-attraction* $\mathbb{P}$ (Brilhante, de Macêdo, Nardini, Perego, & Renso, 2014; Chen et al., 2015).

3.3. Time-based user interest for LUs and GUs

As stated in the previous section, $\mathbb{c}_{\mathbb{P}}$ is the POI-attraction category $\mathbb{c}$. For a specific user, the interest for the category $\mathbb{c}$ can be stated using Eqn. 2 (Gu, Song, Jiang, Wang, & Liu, 2020; Wang, Leckie, Chan, Lim, & Vaithianathan, 2016).

$$inte_{u_i}\mathbb{c} = \sum_{j=1}^{\ell}\frac{(t_{\mathbb{P}_j}^{dept} - t_{\mathbb{P}_j}^{ari})}{\mathscr{D}(\mathbb{P}_j)}\delta(\mathbb{c}_{\mathbb{P}_j} = \mathbb{c})\forall\mathbb{c} \in \mathscr{C} \tag{2}$$

Here, $\delta(\mathbb{c}_{\mathbb{P}_j}) = \begin{cases} 1 & \text{if } \mathbb{c}_{\mathbb{P}_j} = \mathbb{c} \\ 0, & \text{otherwise} \end{cases}$. $\mathscr{C}$ is a set of different categories. Eqn. 2 is used as a means of determining the interest of users in a POI-attraction category with relative to the average visiting time of all visitors at that specific POI-attraction category $\mathbb{c}$. It is obvious that at the POI-attraction, a person would spend more time if he/she is more interested in it.

3.4. Travel history

For a specific user $u \in \mathscr{U}$, the traveling records can be described in the terms of a series of n travel itineraries $S_u = ((i_1, t_{i_1}^{ari}, t_{i_1}^{dept}), \ldots, (i_n, t_{i_n}^{ari}, t_{i_n}^{dept}))$ where, in a triplet $(i_\vartheta, t_{i_\vartheta}^{ari}, t_{i_\vartheta}^{dept}), i_\vartheta$ is the tourist's visited itinerary, $t_{i_\vartheta}^{ari}$ is the arrival time and $t_{i_\vartheta}^{dept}$ is the departure time. The gap between these two shows the length of the trip on the itinerary i_ϑ. In this work, $S_u = ((i_1, t_{i_1}^{ari}, t_{i_1}^{dept}), \ldots, (i_n, t_{i_n}^{ari}, t_{i_n}^{dept}))$ can be written as $S_u = (i_1, \ldots, i_n)$.

3.5. Interest of an itinerary

An itinerary of a visitor may be arranged with POI-attractions $\mathbb{P} = (\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3, .., \mathbb{P}_k)$. An itinerary interest i_ϑ which belongs to S_u and can subsequently be calculated using Eqn. 3 (Sarkar et al., 2020).

$$i_\vartheta(inte) = \sum_{j=1}^{k}\frac{(t_{\mathbb{P}_j}^{ari} - t_{\mathbb{P}_j}^{dept})}{\mathscr{D}(\mathbb{P}_j)} \tag{3}$$

3.6. Popularity of an itinerary

For a specific user, the popularity of an itinerary is calculated as the sum of the ratio of the number of visits done by that user and all the tourists to each of the POI-attraction category travelled by the user. The popularity has been calculated using Eqn. 4 and Eqn. 5 (Lim, Chan, Karunasekera, & Leckie, 2019) (Sarkar et al., 2020).

$$pop(\mathbb{c}) = \sum_{j=1}^{k}\frac{pop_{\mathbb{P}_j}}{\Phi(\mathbb{P}_j)}\ \delta(\mathbb{c}_{\mathbb{P}_j} = \mathbb{c}) \tag{4}$$

Here, $pop(\mathbb{c})$ denotes the popularity of $\mathbb{c}$ dependent upon the amount of $\mathbb{c}$'s popularity for each POI-attraction. The total count, a POI-attraction is toured by all tourists is denoted by $\Phi(\mathbb{P})$. At last Eqn. 5 calculates the popularity of itinerary i_ϑ.

$$i_\vartheta(pop) = \sum_{\varsigma=1}^{\omega}pop(\mathbb{c}_\varsigma) \tag{5}$$

Here, $\varsigma = \{1, 2, 3, \ldots, \omega\}$ represents the total count of categories available in i_ϑ.

3.7. Traveling cost

The cost of travel is determined by comparison to the overall physical area traveled in a trip. Although some of the previous works consider the whole duration of the tour. Time depends on transportation methods such as bus, trains, flights, walks etc. The distance is an important element when the tourists suggest that they use an extensive transportation mode to travel a long distance to visit various POIs. We minimize the traveling time with the help of faster modes of travel as we can cross huge distances while driving at a high speed. When there is a wide distance between two similar POIs, quick manner of transportation is needed and results in higher cost of transportation. We therefore, plan to keep the overall physical range of the whole tour to a minimum. The cost of the journey is measured with Eqn. 6 (Sarkar et al., 2020).

$$\mathscr{T}^{cost}(x) = \sum_{\vartheta=1}^{n}\sum_{j=2}^{k}\Pi^{intr}(i_\vartheta^{\mathbb{P}_{j-1,j}}) + \sum_{\vartheta=1}^{n}\Pi^{extr}(i_\vartheta^{\mathbb{P}_n}, i_{\vartheta+1}^{\mathbb{P}_1}) \tag{6}$$

where, $(\vartheta + 1) < n$. The initial element of Eqn. 6 is the total of the inner area of the tourist itineraries in the tour package. The intrinsic distance i_ϑ is determined based on the total geographical distance between all POI attractions in a path. The second component of the Eqn. 6 is the extraneous range from i_ϑ and $i_\vartheta + 1$ for two successive itineraries, and it is calculated by the distance between the last POI-attraction of i_ϑ and first POI-attraction of $i_\vartheta + 1$.

3.8. Similarity between LU and GUs

The similarity between LU and GUs is calculated on the basis of the Cosine Similarity test in terms of the interest of a particular location for both the LU and GUs and is calculated using Eqn. 7.

$$Cos\_sim(u_x, u_y) = \frac{\vec{i}\,nte_{u_x} \cdot \vec{i}\,nte_{u_y}}{||\vec{i}\,nte_{u_x}|| \cdot ||\vec{i}\,nte_{u_y}||} \tag{7}$$

where, the two distinct tourist names are u_x and u_y.

3.9. Problem definition

The tour recommendation issue for various itineraries is discussed in this section by taking into account *POI-attractions* for a single user. The primary goal is to maximize the interest of a tourist and *POI-attraction's* popularity and minimize travel costs.

The problem of optimization is referred to as the TRIPTOURREC issue labeled with the (Lim et al., 2017) version of the Orienteering Problem.

$$A_i = \Theta(\alpha_1|inte_i(\mathbb{c}_a) - inte_i^{'}(\mathbb{c}_a)| + \alpha_2|inte_i(\mathbb{c}_b) - inte_i^{'}(\mathbb{c}_b)| + \alpha_3|inte_i(\mathbb{c}_c) - inte_i^{'}(\mathbb{c}_c)| + \ldots + \alpha_n|inte_i(\mathbb{c}_n) - inte_i^{'}(\mathbb{c}_n)|) + (1-\Theta)(\beta_1|pop_i(\mathbb{c}_a) - pop_i^{'}(\mathbb{c}_a)| + \beta_2|pop_i(\mathbb{c}_b) - pop_i^{'}(\mathbb{c}_b)| + \beta_3|pop_i(\mathbb{c}_c) - pop_i^{'}(\mathbb{c}_c)| + \ldots + \beta_n|pop_i(\mathbb{c}_n) - pop_i^{'}(\mathbb{c}_n)|) \quad (8)$$

$$\alpha_1 + \alpha_2 + \ldots + \alpha_n = 1 \quad (9)$$

$$\beta_1 + \beta_2 + \ldots + \beta_n = 1 \quad (10)$$

$$\Gamma_{1\varrho} = \sum_{k=1}^{\varrho} A_k \quad (11)$$

$$\Gamma_{1(\varrho+1)} = \Gamma_{\varrho} - A_{\varrho} + A_{\varrho+1} \quad (12)$$

$$\Gamma_{1(\varrho+2)} = \Gamma_{(\varrho+1)} - A_{(\varrho+1)} + A_{(\varrho+2)} \quad (13)$$

$$\vdots$$
$$\Gamma_{1n} = \Gamma_{(n-1)} - A_{(n-1)} + A_n \quad (14)$$

$$\Gamma_{2(\varrho+1)} = \sum_{k=2}^{\varrho+1} A_k \quad (15)$$

$$\Gamma_{2(\varrho+2)} = \Gamma_{2(\varrho+1)} - A_{(\varrho+1)} + A_{(\varrho+2)} \quad (16)$$

$$\Gamma_{2(\varrho+3)} = \Gamma_{2(\varrho+2)} - A_{(\varrho+2)} + A_{(\varrho+3)} \quad (17)$$

$$\vdots$$
$$\Gamma_{2n} = \Gamma_{2(n-1)} - A_{(n-1)} + A_n \quad (18)$$

$$\vdots \quad (19)$$

$$\Gamma_{(n-\varrho+1)n} = \sum_{k=(n-\varrho+1)}^{n} A_k \quad (20)$$

In Eqn. 8, A_i represents the function of tourist's interest and popularity of the tour. $inte(\mathbb{c}_a), inte(\mathbb{c}_b), .., inte(\mathbb{c}_n)$ be the LU's interest for the POI-attraction categories $\mathbb{c}_a$, $\mathbb{c}_b$, .., $\mathbb{c}_n$respectively and $inte^{'}(\mathbb{c}_a), inte^{'}(\mathbb{c}_b)$,..,$inte^{'}(\mathbb{c}_n)$ be the GU's interest for the same POI-attraction categories $\mathbb{c}_a$, $\mathbb{c}_b$, .., $\mathbb{c}_n$respectively from each itinerary. The value of $|inte(\mathbb{c}_a) - inte^{'}(\mathbb{c}_a)|$ is close to zero means that there's a big similarity between the LU and GUs. $pop(\mathbb{c}_a), pop(\mathbb{c}_b), .., pop(\mathbb{c}_n)$ be the LU's tour popularity for the POI-attraction categories $\mathbb{c}_a, \mathbb{c}_b, .., \mathbb{c}_n$ respectively and $pop^{'}(\mathbb{c}_a), popt^{'}(\mathbb{c}_b)$,..,$pop^{'}(\mathbb{c}_n)$ be the GUs tour popularity for the same POI-attraction categories $\mathbb{c}_a, \mathbb{c}_b, .., \mathbb{c}_n$respectively from each itinerary. $\alpha_1, \alpha_2 \ldots, \alpha_n$ and $\beta_1, \beta_2, \ldots, \beta_n$ are the weight parameters respectively and are calculated using Eqn. 21 and 22.

$$\alpha(\mathbb{i}) = \frac{\mathbb{inte}(\mathbb{c}_{\mathbb{i}})}{\sum_{\mathbb{j}=1}^{\vartheta} \mathbb{inte}(\mathbb{c}_{\mathbb{j}})} \quad (21)$$

$$\beta(\mathbb{i}) = \frac{\mathbb{pop}(\mathbb{c}_{\mathbb{i}})}{\sum_{\mathbb{j}=1}^{\vartheta} \mathbb{pop}(\mathbb{c}_{\mathbb{j}})} \quad (22)$$

$$\mathbb{L}(x) = (\Gamma_{1\varrho}, \Gamma_{1(\varrho+1)}, .., \Gamma_{1n}, \Gamma_{2(\varrho+1)}, \Gamma_{2(\varrho+2)}, \ldots\ldots, \Gamma_{(n-\varrho+1)n}) \quad (23)$$

Where, $inte(\mathbb{c}_i)$ and $pop(\mathbb{c}_i)$ are the interest and popularity of *POI-attraction* category $\mathbb{c}_i$. Eqns. 11–19 represent the various itinerary formulation. $\Gamma_{i,\varrho}$ denotes the sum of A for all the itineraries available in the set of i^{th} itinerary of size ϱ. The main aim of this work is to suggest multiple itineraries $i_1, i_2, .., i_n$ that maximize the interest of the tourist, tour popularity and minimize the cost of travel. That is the goal is to recommend several itineraries $(i_1, i_2 \ldots, i_n)$ which consist of *POI-attractions*, such that,

$$Minimize\ (\mathbb{L}(x)) \quad (24)$$

$$Minimize(\mathscr{T}^{cost}(x)) \quad (25)$$

Let, $u_{\mathbb{i},\mathbb{i}'} = 1$, if both the itineraries $\mathbb{i}$ and $\mathbb{i}'$ have been visited in sequence by the visitor. A tourist will fly directly from $\mathbb{i}$ to $\mathbb{i}'$. $u_{\mathbb{i},\mathbb{i}'} = 0$, otherwise (Lim, 2015). The Eqn. 24 and 25 can be solved using the constraints given in 25–29.

$$\sum_{\mathbb{i}'=2}^{\mathbb{N}} u_{1,\mathbb{i}'} = \sum_{\mathbb{i}=1}^{\mathbb{N}-1} u_{\mathbb{i},\mathbb{N}} = 1 \quad (26)$$

$$\sum_{\mathbb{i}=1}^{\mathbb{N}-1} u_{\mathbb{i},\mathbb{k}} = \sum_{\mathbb{i}'=2}^{\mathbb{N}} u_{\mathbb{k},\mathbb{i}'} \leqslant 1; \forall \mathbb{k} = 2, \ldots, \mathbb{N}-1 \quad (27)$$

$$2 \leqslant \mathscr{L}_{\mathbb{i}'} \leqslant \mathbb{N}; \forall \mathbb{i}^{'} = 2, \ldots, \mathbb{N} \quad (28)$$

$$\mathscr{L}_{\mathbb{i}} - \mathscr{L}_{\mathbb{i}'} + 1 \leqslant (\mathbb{N}-1)(1-u_{\mathbb{i},\mathbb{i}'}) \forall \mathbb{i}, \mathbb{i}^{'} = 2, \ldots, \mathbb{N} \quad (29)$$

$$|cost(x)| \leqslant B \quad (30)$$

Eqns. 24 and 25 have shown that the above problem is multi-objective problem and is referred to as TRIPTOURREC problem. The TRIPTOURREC problem involves a range of limitations. The constraints set out in Eqn. 25 suggests that it is possible to launch the suggested package from $\mathbb{i}_1$ and to finish at the $\mathbb{i}_{\mathbb{N}}$. From Eqn. 26, it is observed that the suggested itineraries are linked together and that is not possible to visit an particular itinerary multiple time. Since $\mathscr{L}_{\mathbb{i}}$ is the position of itinerary $\mathbb{i}$ in a package, it is noted from constraints of Eqn. 27 and 28 that the above solution does not provide a sub-tour and is depending on the Traveling Salesman Problem (TSP) (Miller et al., 1960). Constraint 29 ensure that the total distance considered for the package is within the budget B using the $\mathscr{T}^{cost}(x)$ function, which includes the internal and external cost of the different itineraries. The TRIPTOURREC issue is an NP-hard, since this problem depends on the cost function. The different itineraries chosen from a vast number of variations often bring extra complexity to the TRIPTOURREC problem. In order to solve these challenges, a PWP algorithm has been proposed using NSGA-II which is discussed in the subsequent section.

4. NSGA-II based PWP Algorithm

aaa

Algorithm 1: PWP-Calculate-*Remove_duplicate_itinerary_list* (solution)

Algorithm 1: PWP-Calculate-*Remove_duplicate_itinerary_list* (solution)

$\mathcal{I}_l$ =[ ]
for q *in solution* ▷ *q is the itinerary list* **do**
  if q *not in* $\mathcal{I}_l$ **then**
    $\mathcal{I}_l$.append(i) ▷ *Append itinerary if it is not available in the itinerary list*
return $\mathcal{I}_l$

Algorithm 2: PWP-Calculate-*Remove_duplicate_itinerary*(*solution*)

Algorithm 2: PWP-Calculate-*Remove_duplicate_itinerary*(*solution*)

for q *in solution* **do**
  ▷ *q is the set of itinerary list*
  for q' *in range(0, len(solution) - 1)* ▷ *itinerary list from solution* **do**
    for *k in range(0, len*(q') *- 1)* ▷ *itinerary from the itinerary list* **do**
      for *g in range(1, len*(q')*)* ▷ *itinerary from the itinerary list* **do**
        if *k!=g* **then**
          if q' *in solution* **then**
            if $q'[k] == q'[g]$ ▷ *Check whether two itineraries are equal* **then**
              solution.remove(q') ▷ *Remove itinerary from itinerary list if two itineraries are equal*
return *solution*

Algorithm 3: PWP-*CalculateBudget*(*solution*)

Algorithm 3: PWP-*CalculateBudget*(*solution*)

$\mathcal{I}_l$ =[ ]
for q *in solution* **do**
  if *Cost <=B* ▷ *If cost is within the budget* **then**
    $\mathcal{I}_l$.append(q)
return $\mathcal{I}_l$

Algorithm 4: PWP- Calculate-Crossover-Mutation(solution)

$\mathcal{S}^{sol}$ ← *solution* ▷*Created solution based on itineraries of similar global users*
$\mathcal{S}^{cross}$ ←[Crossover($\mathcal{S}^{sol}$, $\mathcal{S}^{sol}$ + 1)]▷*Perform crossover*
$\mathcal{S}^{mut}$ ←[Mutation($\mathcal{S}^{cross}$, $\mathcal{S}^{cross}$ + 1)▷*Perform mutation*
$\mathcal{S}^{B}$ ← *CalculateBudget*($\mathcal{S}^{mut}$)
$\mathcal{S}^{R}$ ← *Remove_duplicate_itinary*($\mathcal{S}^{B}$)
$\mathcal{S}^{L}$ ←*Remove_duplicate_itinary_list*($\mathcal{S}^{B}$)
Return $\mathcal{S}^{L}$

Algorithm 5: Algorithm for NSGA-II based PWP approach

Algorithm 5: Algorithm for NSGA-II based PWP approach

$\mathcal{P}^{size} \leftarrow len(\mathcal{S}^{sol})$ ▷ *Population size*
$\mathcal{G}^{no} \leftarrow 0$
$\mathcal{I}_l \leftarrow 0$
for $\mathcal{U}$ *in matching_user_list* **do**
  $\mathcal{I}_l \leftarrow (\mathcal{U}$, *itinerary)* ▷ *itineraries of similar global users*
$\mathcal{S}^{sol} \leftarrow \mathcal{I}_l$
while $(\mathcal{G}^{no} < max\_gen)$ **do**
  $\mathcal{F}_1 \leftarrow Calculate\ fitness\_1,\ \mathcal{S}^{sol}$
  $\mathcal{F}_2 \leftarrow Calculate\ fitness\_2,\ \mathcal{S}^{sol}$
  $\mathcal{N}_d \leftarrow non\_dominated\_sort(\mathcal{F}_1, \mathcal{F}_2)$ ▷ *NSGA-II's fast non dominated sort*
  $\mathcal{C}_{crowd}$ = [ ]
  for *each* $\mathcal{N}_d$ **do**
    $\mathcal{C}_{crowd}.append(crowding\_distance(\mathcal{F}_1, \mathcal{F}_2, \mathcal{N}_d))$ ▷ *Calculate crowding distance*
  $\mathcal{S}_1^{sol} = \mathcal{S}^{sol}$
  for *each* $\mathcal{S}_1^{sol}$ **do**
    $\mathcal{S}_2^{sol} = Calculate - Crossover - Mutation(\mathcal{S}_1^{sol})$
  $\mathcal{F}_1' \leftarrow Calculate\ fitness\_1,\ \mathcal{S}_2^{sol}$ ▷ *based on Eqn. 30*
  $\mathcal{F}_2' \leftarrow Calculate\ fitness\_2,\ \mathcal{S}_2^{sol}$ ▷ *based on Eqn. 31*
  $\mathcal{N}_d' \leftarrow non\_dominated\_sort(\mathcal{F}_1', \mathcal{F}_2')$
  $\mathcal{C}_{crowd}'$ = [ ]
  for *each* $\mathcal{N}_d'$ **do**
    $\mathcal{C}_{crowd}'.append(crowding\_distance(\mathcal{F}_1', \mathcal{F}_2', \mathcal{N}_d'))$ ▷ *Calculate crowding distance*
  $\mathcal{S}_{new}^{sol}$ = [ ]
  for *each* $\mathcal{N}_d'$ **do**
    $front = Sort(\mathcal{N}_d', \mathcal{C}_{crowd}')$ ▷ *Function to sort by values*
    for *value in front* **do**
      $\mathcal{S}_{new}^{sol}.append(value)$
      if $(len(\mathcal{S}_{new}^{sol}) == \mathcal{P}^{size})$ **then**
        break
  $\mathcal{S}^{sol} = \mathcal{S}_{new}^{sol}$ ▷ *Update the solution*
  $\mathcal{G}^{no} = \mathcal{G}^{no} + 1$

We propose a new multiple itinerary recommendation system named as PWP which is based on multi-objective optimization approach. The multi-objective optimization approach used here is NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002). The genetic operators used in NSGA-II help to establish a better solution than manipulating the motion of the particles in the non-dominated Sorting Particle Swarm Optimization Algorithm (NSPSO). Although PSO is easier to implement, NSGA-II does work well when multi-objective problems are involved (Subashini & Bhuvaneswari, 2013). This method describes each target as vectors and

Fig. 4. Rows indicate the number of tourists and the itineraries visited by each tourist is shown in columns. Process of crossover and mutation between itineraries.

optimizes all vectors simultaneously (Deb et al., 2002). The proposed PWP algorithm addresses the problem of TRIPTOURREC. The TRIPTOURREC problem is based on an orienteering problem variant that is NP-hard because the orienteering problem is a specialist instance of the Travelling Salesman Problem (Lim et al., 2017). The basic difference between PWP over baselines is that, PWP effectively optimizes interest, popularity and cost during the selection of each itinerary using the NSGA-II approach. An itinerary list is generated by comparing local and global tourists. The algorithm for *Remove_*

$$duplicate\_itinerary\_list, Remove\_duplicate\_$$

$$itinerary, CalculateBudget, Calculate - Crossover - Mutation$$

and NSGA-II based PWP approach is presented in Algorithm 1–5. As defined in Eqns. 31 and 32, NSGA-II uses two fitness functions.

$$fitness_1 = A_i \tag{31}$$

$$fitness_2 = \mathscr{T}^{cost}(x) \tag{32}$$

The collective solution list is given as an input to the NSGA-II based PWP approach presented in Algorithm 5. Using the cosine similarity test a solution is created based on 10 best global user(Steps 4–6). The fitness functions $fitness_1$ and $fitness_2$ is applied on every solution (Steps 9–10 and 17–18) using Eqn. 31 and 32. NSGA-II uses quick non-dominated-sort (Steps 10 and 19) as a method to compare every solution of the population list with another solution from the list to sort the single solution list into a distinct dominant form. Prepare the new population based on the crossover process (Step 16) (see Fig. 4). With respect to bi-objective non-dominated sorting optimization algorithm, one individual dominates the other if only if the dominant concept is used to split the N individual population list into multiple fronts. For proper selection, the NSGA-II uses crowding distance by suspending a front and selecting a subgroup of individual solutions with equal number of dominance level (Steps 13 and 22). This steps continue up to the final repetition or end criteria (step 7). If the higher number of generations are needed to be obtained, the NSGA-II repeats the process. Else the top-N optimal recommendations of individual solutions will be considered. It may be possible for the solution to have a duplicate itinerary or duplicate itinerary list. To solve these issues, PWP utilizes the *Remove_duplicate_itinerary_list*() and *Remove_duplicate_itinerary*(). In other words, the solution does not have a redundant itinerary. However, the complete travel costs of recommended itinerary should be within the budget. The CalculateBudget () is thus used by the PWP algorithm.

4.1. Complexity analysis

The complexity of the non-dominated sorting of NSGA-II algorithm is $O(M(2N)^2)$ and complexity of the crowding-distance assignment is $O(M(2N)log(2N))$ [11]. Here M and N are the amount of goals and volume of the population. In Algorithm 5, steps 1,2 and 3 take $O(1)$ each. Let us consider that the size of the matching user list is $|U|$. So, steps 4 and 5 take $O(|U|)$. If maximum number of generation is G then steps 7–9 take $O(G)$. Step 10 takes $O(MN^2 \times G)$. Step 12 takes $O(|\mathscr{N}_d| \times G)$ and step 13 takes $O(M(2N)log(2N) \times |\mathscr{N}_d| \times G)$. Where, $|\mathscr{N}_d|$ is the size of $\mathscr{N}_d$ (list to store non-determinant sorting values). Steps 1–13 take $O(MN^2 \times G)$ as it dominates all other operations from steps 1–13. Step 14 takes $O(G)$ and step 15 is taken $O(N \times G)$. During the crossover selection PWP utilizes *Remove_duplicate_itinerary_list*(), *Remove_duplicate_itinerary*() and *CalculateBudget*() function. In Algorithm 4, steps 4 and 6 take $O(N)$ each, step 5 takes $O(N(\mathbb{Q}-1)^2 \times (\mathbb{Q}^{'}-1))$. Where $\mathbb{Q}$ and $\mathbb{Q}^{'}$ are the length of q and $q^{'}$ respectively (see Algorithm 2). So, the overall complexity of step 16 of Algorithm 5 is $O(N(\mathbb{Q}-1)^2 \times (\mathbb{Q}^{'}-1) \times N \times G)$ as it dominates all other operations from steps 4–6 of Algorithm 4. Therefore, steps 1–16 of Algorithm 5 take, $O(MN^2 \times G) + O(N(\mathbb{Q}-1)^2 \times (\mathbb{Q}^{'}-1) \times N \times G)$. Steps 17 and 18 take $O(G)$ each. Step 19 takes $O(MN^2 \times G)$. Steps 20 take $O(G)$. Step 21 takes $O(|\mathscr{N}^{'}_d| \times G)$ and step 22 takes $O(M(2N)log(2N) \times |\mathscr{N}^{'}_d| \times G)$. Where, $|\mathscr{N}^{'}_d|$ is the size of $\mathscr{N}^{'}_d$ (list to store non-determinant sorting values). Steps 23 and 24 take $O(G)$ and $O(|\mathscr{N}^{'}_d| \times G)$ respectively. Step 25 takes $O(2Nlog(2N)) \times |\mathscr{N}^{'}_d| \times G$. Hence, steps 19–25 take $O(MN^2 \times G)$ as it dominates all other operations. Steps 26–28 take $O(F \times |\mathscr{N}^{'}_d| \times G)$, where F is the front size. Steps 30 and 31 take $O(G)$ each. Hence, the total complexity is $O(MN^2 \times G) + O(N^2(\mathbb{Q}-1)^2 \times (\mathbb{Q}^{'}-1) \times G) + O(F \times |\mathscr{N}^{'}_d| \times G)$.

Table 2
Definition of data sets about total count of visitors, visits with POI attractions and series of journeys.

City	# of visitors	POI-attraction Visits	Traveling series
Osaka	390	6531	124
Edinburgh	1320	13200	1001
Budapest	835	13610	2001
Delhi	261	3990	410
Vienna	950	21100	1132
Glasgow	500	10111	2008
Toronto	1070	29501	4021

5. Experimental methodology

5.1. Dataset

We used the dataset outlined in (Lim et al., 2017) in this study. The dataset includes Yahoo! Flickr Creative Commons 100 M (YFCC100M) dataset (Thomee et al., 2016), 100 M Flickr images and videos. In addition, we have utilized the YFCC100M dataset, as shown in Table 2 and collected geo-tagged imagery in various parts of the world. The scene of a tourist (A) where he/ she is indeed a Delhi person wishing to journey Edinburgh is depicted Fig. 6. In this scenario, Delhi is the local dataset (A) with the background of tourists and Edinburgh that is the global data set of specific (B) tourists. A Delhi dataset sample is shown in Fig. 5. The various parameters are: User ID, POI-attractions with different IDs, visit date etc. A user encounters numerous POIs, including Chandni Chowk and Jama Masjid etc. A specific ID, and a category including religious, entertainment etc., are named in every POI. The dataset includes the meta information of the pictures. It consists date/ time of visits.The dataset also includes geo-coordinate data that can be used to detect the distance between itineraries. The datasets used in this work can be downloaded from https://sites.google.com/site/limkwan-hui/datacode?authuser = 0. For our experiment, we can easily extract the required data on the *POI-attractions* from photoID. Fig. 5 shows a scenario where a tourist belongs to Delhi and intends to travel to Edinburgh without any past travel history. Here, Delhi is a local dataset with the history of the tourist and Edinburgh is a global dataset with histories of similar tourists.

5.2. Baseline algorithms

Based on the work (Lim et al., 2017), in our experiment we considered all the baseline algorithms start from one POI and then pick the next successive POIs until the budget is reached. To recommend the multiple itineraries, we use the sequence of travels by the visitor. To arrange multiple itineraries, we utilized the remaining POIs after establishing a single itinerary. We have considered 4 h of travel time with a single

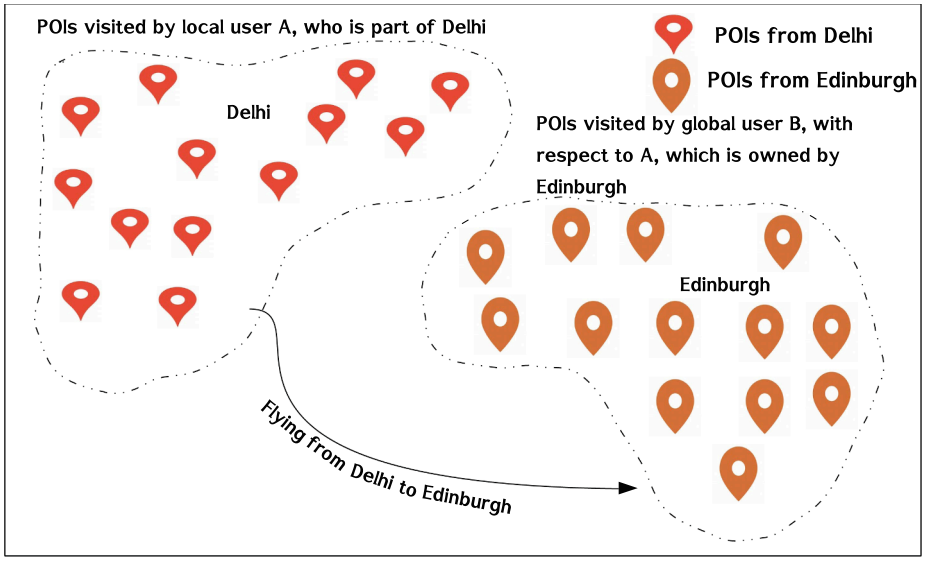

Fig. 5. Delhi-Edinburgh dataset situation, if a tourist wishes to visit Edinburgh from Delhi.

USER-ID	POI	POI-Attraction	POI-Attraction ID	CATEGORY	DATE TAKEN
@#01	Chandni Chowk	Restaurant	1	Entertaiment	Friday, August 6, 2011 6:02:01 PM
@#01	Chandni Chowk	Restaurant	1	Entertaiment	Friday, August 6, 2011 6:02:01 PM
@#01	Chandni Chowk	Museum	2	Entertaiment	Friday, August 6, 2011 6:02:01 PM
@#01	Chandni Chowk	Temple	3	Religious	Friday, August 6, 2011 6:02:01 PM
@#01	Jama Masjid	Restaurant	4	Entertaiment	Tuesday, June 6, 2012 8:02:01 AM
@#01	Jama Masjid	Temple	5	Religious	Tuesday, June 6, 2012 8:02:01 AM
@#01	Jama Masjid	Temple	5	Religious	Tuesday, June 6, 2012 8:02:01 AM
@#02	Lotus temple	Park	6	Entertaiment	Monday, June 6, 2012 12:02:01 PM
@#02	Lotus temple	Temple	7	Religious	Monday, June 6, 2012 1:12:05 PM
@#02	Lotus temple	Temple	7	Religious	Monday, June 6, 2012 3:22:04 PM
@#02	Jama Masjid	Restaurant	4	Entertaiment	Friday, June 15, 2012 8:02:01 AM
@#2	Jama Masjid	Masjid	8	Religious	Friday, June 15, 2012 9:02:03 AM

Fig. 6. Definition of the dataset with various POIs, POI-attractions, dates of category and travel of 2 Delhi tourists.

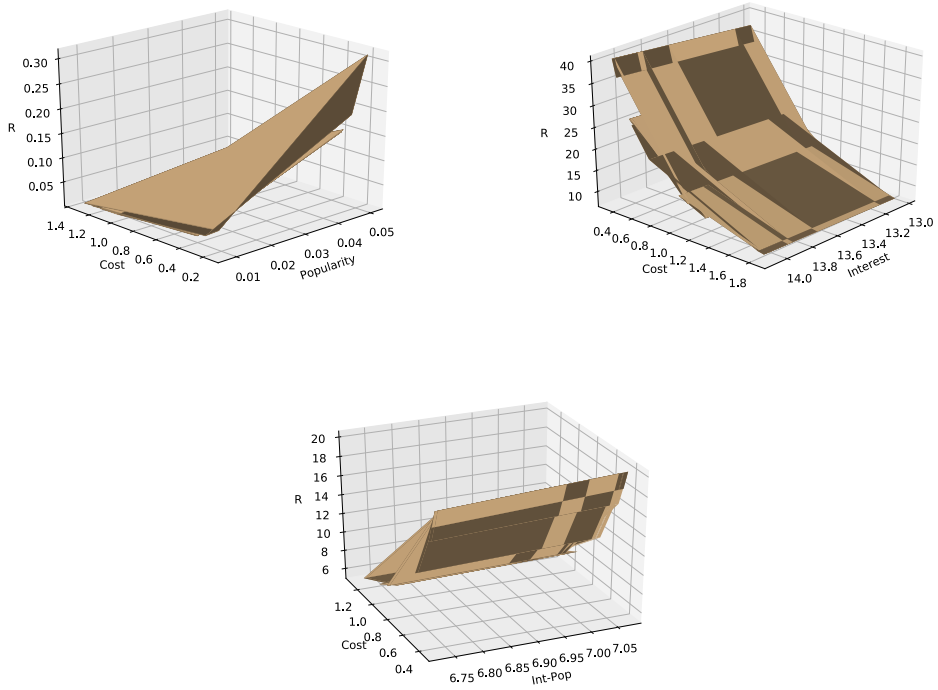

Fig. 7. Representation of A_i when Θ=0,1 and 0.5.

itinerary, which can quickly help to construct a number of itineraries.

- **Greedy Nearest (G-NEAR)**: By choosing the three nearest attractions, we use this algorithm for the next unvisited POI (Lim et al., 2016).
- **Greedy Most Popular (G-POP)**: By choosing the three most popular destinations, we select resulting unexplored POI (Lim et al., 2016).
- **Multiple Itinerary Tourist Recommendation (MULTITOUR)**: It offers multiple itineraries considering the itinerary interest and popularity of the destination and travel cost (Sarkar et al., 2020).
- **Personalized Tour Recommendation (PERSTOUR-P)**: This offers exclusive itineraries in view of the popularity of the destination (Lim et al., 2016).
- **PERSTOUR with Adaptive Weighting (PERSTOUR-W)**: PersTour focuses on both the POI popularity and the interest of the tourist with weighted updates. It gives importance on an efficient scaling of the number of visits to POIs (Lim, Chan, Leckie, & Karunasekera, 2018).
- **Iterative Heuristic Approximation (IHA)**: To maximize the gain of a visitor as per the travel cost, a fresh POI is chosen depend on the highest heuristic value of destinations utilizing $\frac{inte(cp_j)+Pop(p_j)}{Trav_{p_j\ p_j}+Dur_{p_j}}$ and is put between the starting POI and the end POI p_n of the itinerary. This procedure is conducted in compliance with the budget (Zhang, Liang, & Wang, 2016).
- **Tour Recommendation With Interest Category (TOURINT)**: This illustrates the problem of recommended tour with a mandatory category that the tourist will visit on the recommended itinerary at least once. It defines this mandatory category as the most commonly visited category of other tourist visit sequence (Lim, 2015).
- **Trip Builder (TRIPBUILD)**: This creates the tourist's customized itinerary depending on interest and popularity of the destination. A POI's interest is computed as the sum of a particular category's POI visitations as compared to his/ her total visit (Brilhante, Macedo, Nardini, Perego, & Renso, 2015).

5.3. Real-life Evaluation

The evaluation will be carried out only for visitors with minimum two visiting series and more than two categories. The method is implemented on local and global datasets and the matching users have been identified. We will determine the matching users in this research by recognizing the top 10 related users of the GU's list. The different features of related users are extracted from the local dataset. We choose the following matrices to equate various baselines with our proposed approach. For our tests, the real-life travel sequences are selected based on the previous records of related users at a location, a visitor wishes to visit.

- **Tour Recall (TourRec(I))**: It defines the portion of visitor's real-life traveling sequence that is still portion of the recommended Itinerary *I*. It is presumed that the itinerary *I* consists C_{rec} which represents a list of suggested categories and C_{real} depicts a collection of categories encountered by a visitor's real-life travel sequence. The $TourRec(I)$ can be represented using Eqn. 33.

$$TourRec(I) = \frac{|C_{rec} \cap C_{real}|}{|C_{real}|} \tag{33}$$

- **Tour Precision (TourPre(I))**: The *Tour Precision* is specified as the ratio of suggested categories in the *I* itinerary that were still portion of the tourist's real-life travel series. It is presumed that the *I* itinerary comprises C_{rec} which specifies a list of recommended categories and C_{real} represents a collection of categories viewed by a tourist in his/ her real-life traveling series. *Tour Precision* can be expressed as shown in Eqn. 34.

$$TourPre(I) = \frac{|C_{rec} \cap C_{real}|}{|C_{rec}|} \tag{34}$$

Table 3
Precision analysis of PWP with multiple baseline approaches utilizing various data sets. "bold" colour represents the best value.

Algorithms	PWP (0.5)	PWP (1)	PWP (0)	MULTITOUR	PERSTOUR-W	IHA	PERSTOUR-P	TOURINT	TRIPBUILDER	G-POP	G-NEAR
Delhi-Edinburgh	**0.690** ± **0.037** 1	0.548 ± 0.038 4	0.5 ± 0.029 6	0.618 ± 0.024 2	0.528 ± 0.035 5	0.444 ± 0.044 8	0.474 ± 0.022 7	0.581 ± 0.017 3	0.368 ± 0.041 10	0.381 ± 0.009 9	0.345 ± 0.016 11
Osaka-Edinburgh	**0.571** ± **0.014** 1	0.396 ± 0.038 7	0.370 ± 0.03 8	0.478 ± 0.023 2	0.453 ± 0.013 4	0.422 ± 0.05 6	0.442 ± 0.027 5	0.467 ± 0.019 3	0.308 ± 0.028 9	0.270 ± 0.026 10	0.259 ± 0.007 11
Vienna-Edinburgh	0.515 ± 0.038 3	0.449 ± 0.022 5	0.357 ± 0.017 8	0.535 ± 0.048 2	0.409 ± 0.047 6	0.391 ± 0.046 7	**0.611** ± **0.032** 1	0.462 ± 0.014 4	0.258 ± 0.028 11	0.318 ± 0.023 9	0.295 ± 0.019 10
Delhi-Osaka	0.611 ± 0.019 2	0.519 ± 0.025 3	0.467 ± 0.038 4	0.459 ± 0.014 5	**0.65** ± **0.026** 1	0.440 ± 0.041 6	0.429 ± 0.016 7	0.414 ± 0.036 8	0.391 ± 0.039 9	0.364 ± 0.017 10	0.326 ± 0.042 11
Glasgow-Edinburgh	**0.439** ± **0.037** 1	0.333 ± 0.019 4	0.316 ± 0.029 6	0.341 ± 0.044 3	0.325 ± 0.027 5	0.280 ± 0.035 8	0.292 ± 0.022 7	0.348 ± 0.009 2	0.233 ± 0.032 10	0.250 ± 0.015 9	0.212 ± 0.026 11
Delhi-Buda	**0.698** ± **0.015** 1	0.544 ± 0.039 6	0.440 ± 0.036 9	0.634 ± 0.030 2	0.568 ± 0.038 5	0.606 ± 0.010 3	0.524 ± 0.042 7	0.583 ± 0.018 4	0.411 ± 0.043 11	0.421 ± 0.016 10	0.492 ± 0.007 8
Buda-Edinburgh	0.357 ± 0.034 5	0.419 ± 0.043 2	0.391 ± 0.042 3	0.340 ± 0.023 6	0.310 ± 0.048 7	0.294 ± 0.017 8	0.283 ± 0.028 9	**0.480** ± **0.021** 1	0.233 ± 0.027 10	0.182 ± 0.037 11	0.375 ± 0.006 4
Delhi-Vienna	0.565 ± 0.036 3	0.647 ± 0.026 2	0.432 ± 0.025 7	**0.690** ± **0.023** 1	0.396 ± 0.034 9	0.375 ± 0.050 10	0.408 ± 0.016 8	0.342 ± 0.039 11	0.5 ± 0.020 5	0.536 ± 0.028 4	0.478 ± 0.006 6
Delhi-Glasgow	**0.521** ± **0.048** 1	0.354 ± 0.043 3	0.313 ± 0.02 6	0.479 ± 0.009 2	0.333 ± 0.018 8	0.375 ± 0.042 7	0.458 ± 0.019 4	0.417 ± 0.025 11	0.229 ± 0.038 9	0.271 ± 0.019 10	0.208 ± 0.007 5
Buda-Toronto	**0.595** ± **0.044** 1	0.513 ± 0.023 3	0.386 ± 0.025 8	0.525 ± 0.013 2	0.487 ± 0.043 4	0.366 ± 0.040 9	0.333 ± 0.029 10	0.295 ± 0.015 11	0.407 ± 0.031 7	0.433 ± 0.021 6	0.471 ± 0.028 5
Buda-Vienna	**0.559** ± **0.048** 1	0.410 ± 0.020 6	0.379 ± 0.017 7	0.5 ± 0.039 2	0.471 ± 0.022 4	0.455 ± 0.016 5	0.481 ± 0.042 3	0.266 ± 0.011 11	0.304 ± 0.031 9	0.276 ± 0.041 10	0.345 ± 0.008 8
Buda-Glasgow	**0.531** ± **0.028** 1	0.471 ± 0.039 4	0.455 ± 0.035 5	0.423 ± 0.014 9.5	0.381 ± 0.019 2	0.350 ± 0.049 3	0.316 ± 0.011 11	0.293 ± 0.044 9.5	0.288 ± 0.022 8	0.288 ± 0.042 7	0.263 ± 0.006 6

- **Tour F1-Score (TourF1-score(I))**: The suggested Itinerary I to the tourist which has the harmonic average of Precision and Recall is called the *Tour F*1 *−Score* which is in Eqn. 35.

$$TourF1-score(I) = \frac{2 \times TourPre(I) \times TourRec(I)}{TourPre(I) + TourRec(I)} \tag{35}$$

- **Mean Absolute Error (MAE)**: The MAE metric is mainly used to determine the accuracy and reliability of Recommended Itineraries. The MAE differentiates the recommended itineraries from the real itineraries visited. The *MAE* can be represented using Eqn. 36.

$$MAE = \frac{|\mathscr{A}_{real} - \mathscr{P}_{rec}|}{|N|} \tag{36}$$

where, $\mathscr{P}_{rec}$and $\mathscr{A}_{real}$are the collections of properly recommended categories and the categories of real-life travel sequence that a specific tourist has encountered. N is the cumulative amount of real-life categories by a particular visitor.

5.4. Heuristic based evaluation

We have found the following heuristic metrics in order to determine the efficiency of our proposed PWP algorithm over various baseline algorithms:

- **Total Itineraries Recommended**: A visitor is advised the number of itineraries for the plan he/she wants.

Table 4
Recall analysis of PWP with multiple baseline approaches utilizing various data sets. "bold" colour represents the best value.

Algorithms	PWP (0.5)	PWP (1)	PWP (0)	MULTITOUR	PERSTOUR-W	IHA	PERSTOUR-P	TOURINT	TRIPBUILDER	G-POP	G-NEAR
Delhi-Edinburgh	0.339 ± 0.023 3	0.288 ± 0.050 6	0.220 ± 0.043 7	0.356 ± 0.014 2	0.322 ± 0.019 4	0.203 ± 0.028 8	0.305 ± 0.039 5	**0.424** ± **0.036** 1	0.119 ± 0.031 11	0.136 ± 0.028 10	0.169 ± 0.190 9
Osaka-Edinburgh	**0.483** ± **0.039** 1	0.362 ± 0.022 5	0.293 ± 0.032 7	0.379 ± 0.036 4	0.414 ± 0.017 2	0.328 ± 0.011 6	0.397 ± 0.024 3	0.121 ± 0.048 11	0.138 ± 0.020 10	0.172 ± 0.035 9	0.241 ± 0.013 8
Vienna-Edinburgh	0.304 ± 0.033 4	0.393 ± 0.024 2	0.268 ± 0.015 5	**0.411** ± **0.031** 1	0.321 ± 0.043 3	0.161 ± 0.042 8	0.196 ± 0.011 7	0.107 ± 0.040 11	0.143 ± 0.029 9	0.125 ± 0.034 10	0.232 ± 0.013 6
Delhi-Osaka	0.415 ± 0.009 2	**0.509** ± **0.018** 1	0.264 ± 0.042 7	0.321 ± 0.027 4	0.245 ± 0.035 8	0.208 ± 0.029 10	0.170 ± 0.017 11	0.226 ± 0.028 9	0.340 ± 0.013 3	0.302 ± 0.049 5	0.283 ± 0.021 6
Glasgow-Edinburgh	**0.316** ± **0.023** 1	0.228 ± 0.034 4.5	0.211 ± 0.050 6	0.263 ± 0.016 2	0.228 ± 0.039 4.5	0.246 ± 0.020 3	0.123 ± 0.014 11	0.140 ± 0.024 10	0.175 ± 0.033 8	0.158 ± 0.004 9	0.193 ± 0.033 7
Delhi-Buda	0.588 ± 0.024 2	**0.608** ± **0.016** 1	0.431 ± 0.013 7	0.510 ± 0.014 4	0.412 ± 0.041 8	0.392 ± 0.012 9	0.216 ± 0.011 11	0.275 ± 0.044 10	0.451 ± 0.040 6	0.471 ± 0.034 5	0.569 ± 0.013 3
Buda-Edinburgh	**0.364** ± **0.038** 1	0.236 ± 0.026 6	0.164 ± 0.050 8	0.291 ± 0.014 4	0.327 ± 0.032 2	0.273 ± 0.044 5	0.309 ± 0.049 3	0.218 ± 0.033 7	0.127 ± 0.041 10	0.145 ± 0.029 9	0.109 ± 0.014 11
Delhi-Vienna	0.553 ± 0.037 2	0.468 ± 0.019 3	0.340 ± 0.029 7	**0.617** ± **0.017** 1	0.404 ± 0.018 5	0.383 ± 0.038 6	0.426 ± 0.028 4	0.277 ± 0.011 9	0.213 ± 0.021 11	0.319 ± 0.037 8	0.234 ± 0.016 10
Delhi-Glasgow	**0.521** ± **0.014** 1	0.354 ± 0.026 6	0.313 ± 0.041 8	0.479 ± 0.021 2	0.333 ± 0.044 7	0.375 ± 0.047 5	0.458 ± 0.049 3	0.417 ± 0.034 4	0.229 ± 0.045 10	0.271 ± 0.029 9	0.208 ± 0.041 11
Buda-Toronto	**0.489** ± **0.014** 1	0.444 ± 0.028 3	0.378 ± 0.046 6	0.467 ± 0.032 2	0.422 ± 0.034 4	0.333 ± 0.020 7	0.156 ± 0.038 11	0.400 ± 0.022 5	0.244 ± 0.017 9	0.289 ± 0.042 8	0.178 ± 0.006 10
Buda-Vienna	**0.452** ± **0.021** 1	0.381 ± 0.015 4.5	0.262 ± 0.043 8	0.429 ± 0.026 2	0.381 ± 0.030 4.5	0.357 ± 0.014 6	0.31 ± 0.038 7	0.405 ± 0.018 3	0.167 ± 0.045 11	0.19 ± 0.034 10	0.238 ± 0.016 9
Buda-Glasgow	0.436 ± 0.036 2	0.282 ± 0.031 7	0.205 ± 0.025 9	0.41 ± 0.023 4.5	0.462 ± 0.050 3	0.256 ± 0.017 8	0.385 ± 0.041 4.5	**0.487** ± **0.029** 1	0.308 ± 0.024 6	0.154 ± 0.029 11	0.179 ± 0.013 10

Table 5
*F*1 −*Score* analysis of PWP with multiple baseline approaches utilizing various data sets. "bold" colour represents the best value.

Algorithms	PWP (0.5)	PWP (1)	PWP (0)	MULTITOUR	PERSTOUR-W	IHA	PERSTOUR-P	TOURINT	TRIPBUILDER	G-POP	G-NEAR
Delhi-Edinburgh	0.455 ± 0.040 2	0.378 ± 0.049 5	0.306 ± 0.011 7	0.452 ± 0.025 3	0.400 ± 0.016 4	0.279 ± 0.026 8	0.371 ± 0.013 6	**0.490** ± **0.032** 1	0.179 ± 0.021 11	0.200 ± 0.26 10	0.227 ± 0.17 9
Osaka-Edinburgh	**0.523** ± **0.018** 1	0.378 ± 0.045 5	0.327 ± 0.021 7	0.423 ± 0.015 3	0.432 ± 0.016 2	0.369 ± 0.042 6	0.418 ± 0.031 4	0.192 ± 0.043 10	0.190 ± 0.026 11	0.211 ± 0.036 9	0.250 ± 0.019 8
Vienna-Edinburgh	0.382 ± 0.016 3	0.419 ± 0.041 2	0.306 ± 0.026 5	**0.465** ± **0.034** 1	0.36 ± 0.02 4	0.228 ± 0.013 8	0.297 ± 0.044 6	0.174 ± 0.023 11	0.184 ± 0.025 9	0.179 ± 0.042 10	0.260 ± 0.015 7
Delhi-Osaka	0.494 ± 0.049 2	**0.514** ± **0.034** 1	0.337 ± 0.045 6	0.378 ± 0.048 3	0.356 ± 0.043 5	0.282 ± 0.02 10	0.243 ± 0.021 11	0.293 ± 0.044 9	0.364 ± 0.047 4	0.330 ± 0.014 7	0.303 ± 0.036 8
Glasgow-Edinburgh	**0.367** ± **0.028** 1	0.271 ± 0.011 3	0.253 ± 0.021 6	0.297 ± 0.017 2	0.268 ± 0.018 4	0.262 ± 0.038 5	0.173 ± 0.036 11	0.200 ± 0.026 8.5	0.200 ± 0.025 8.5	0.194 ± 0.009 10	0.202 ± 0.036 7
Delhi-Buda	**0.638** ± **0.026** 1	0.574 ± 0.046 2	0.436 ± 0.047 8	0.565 ± 0.017 3	0.477 ± 0.032 5	0.476 ± 0.019 6	0.306 ± 0.033 11	0.373 ± 0.021 10	0.430 ± 0.045 9	0.444 ± 0.028 7	0.527 ± 0.037 4
Buda-Edinburgh	**0.36** ± **0.011** 1	0.302 ± 0.040 4	0.231 ± 0.016 8	0.314 ± 0.022 3	0.319 ± 0.015 2	0.283 ± 0.009 7	0.296 ± 0.018 6	0.3 ± 0.039 5	0.165 ± 0.019 10	0.162 ± 0.029 11	0.169 ± 0.015 9
Delhi-Vienna	0.559 ± 0.014 2	0.543 ± 0.024 3	0.381 ± 0.033 7	**0.652** ± **0.044** 1	0.4 ± 0.027 5.5	0.379 ± 0.035 8	0.417 ± 0.022 4	0.306 ± 0.009 10	0.299 ± 0.032 11	0.400 ± 0.022 5.5	0.314 ± 0.017 9
Delhi-Glasgow	0.526 ± 0.016 2	0.447 ± 0.036 4	0.366 ± 0.039 8	**0.568** ± **0.027** 1	0.368 ± 0.035 7	0.396 ± 0.029 5	0.484 ± 0.017 3	0.377 ± 0.028 6	0.289 ± 0.013 11	0.310 ± 0.027 9	0.290 ± 0.036 10
Buda-Toronto	**0.537** ± **0.042** 1	0.476 ± 0.011 3	0.382 ± 0.016 5	0.494 ± 0.041 2	0.452 ± 0.026 4	0.349 ± 0.048 6	0.212 ± 0.047 11	0.340 ± 0.033 8	0.306 ± 0.024 9	0.347 ± 0.011 7	0.258 ± 0.035 10
Buda-Vienna	**0.5** ± **0.019** 1	0.395 ± 0.028 5	0.310 ± 0.024 8	0.462 ± 0.023 2	0.421 ± 0.013 3	0.400 ± 0.032 4	0.377 ± 0.036 6	0.321 ± 0.050 7	0.215 ± 0.027 11	0.225 ± 0.015 10	0.282 ± 0.036 9
Buda-Glasgow	**0.479** ± **0.037** 1	0.338 ± 0.038 4	0.267 ± 0.040 9	0.330 ± 0.032 5	0.438 ± 0.021 2	0.328 ± 0.016 6	0.313 ± 0.026 7	0.362 ± 0.013 3	0.300 ± 0.043 8	0.207 ± 0.035 11	0.237 ± 0.017 10

- **Itinerary Popularity**: The summing up of popularity rates for a specific tourist on all suggested itineraries and is determined by Eqn. 37.

$$Tour\_Pop(I) = \sum i \in pPop(i) \tag{37}$$

- **Tourist Interest**: It is determined by summarizing the different values of interest for all suggested itineraries using Eqn. 38.

$$Tour\_Int(I) = \sum i \in pinte(i) \tag{38}$$

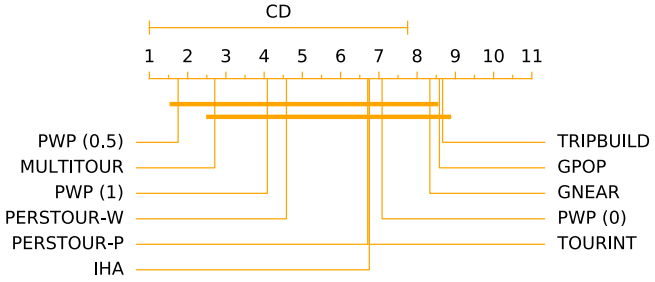

Fig. 8. CD diagrams (Recall) of the comparing algorithms.

5.5. Variants of PWP algorithm

We have used the following variants in our experiment.

- **PWP using** $\Theta = 0$ **(PWP (0))** PWP with complete focus on *POI-attraction* popularity, ignoring the interest of users.
- **PWP using** $\Theta = 1$ **(PWP (1))** PWP with complete focus on interest of the tourist and ignoring the popularity of *POI-attraction*.
- **PWP using** $\Theta = 0.5$ **(PWP (0.5))** equal focus on balancing both the *POI-attraction* popularity and the tourist interest.

5.6. Comparison of Recall, Precision, & F1-Score values

The performance of PWP approach is better than various baseline algorithms. The scores of *Recall*, *Precision* & *F*1-*Score* between the different variants of PWP approach and the various baseline algorithms are shown in Tables 3–5. In case of PWP (1), the *Recall*, *Precision* & *F*1-*Score* metrics are high for all the dataset as compared with PWP (0). The PWP (1) algorithm considers the time-based interest for the tourist and for PWP (0), it considers the popularity based interest. PWP (1) more accurately recommends the set of itineraries due to the use of POI-attraction visit duration instead of POI-attraction popularity. Let us consider a scenario where a tourist visited three Museum but spent more than 5 h and another tourist B who have visited six Museum but spent less than five hours. Under this circumstance, PWP (0) incorrectly categorize tourist B as having more interest on Museum due to his/her six visits. However, PWP (1) can more precisely categorize the interest of visitors based on how long a tourist spent in a specific POI-attraction. In comparison to two other variants, the PWP (.5) scores of *Recall*, *Precision* & *F*1-*Score* are higher in for all datasets, since PWP (.5) gives a balanced emphasis on the *POI-attraction* as well as the interests of visitors. The improvement in *Recall* values for the proposed PWP algorithm varies from 3.4%-22.6% relative to other baseline algorithms. The values of *Recall* are depend on $|C_v|$ and $|C_{rec} \cap C_{real}|$ based on the Eqn. 33. In this case, the value of $|C_{rec} \cap C_{real}|$ obtained using the PWP algorithm, is higher relative to different baselines. Generally, the proposed PWP algorithm is run on two datasets, namely local and global datasets, and finally recommends multiple itineraries which result in higher *Recall* values relative to different baselines.

The efficiency of *MULTITOUR* algorithm is better than the different baseline algorithms because it considers the itinerary popularity, interest of the visitor and travel costs on the basis of distance constraint

Table 6
Overview of the Friedman statistics $\mathscr{F}_F(\mathscr{K}' = 11, \mathscr{N}' = 12)$ and the critical value for each assessment criterion($\mathscr{K}'$: The count of comparison approaches; $\mathscr{N}'$: # of Data Sets).

Metrics	$\mathscr{F}_F$	**Critical score**($\hat{\Gamma} = 0.05$)
Precision	9.77	1.918
Recall	11.66	
F1-Score	22.26	

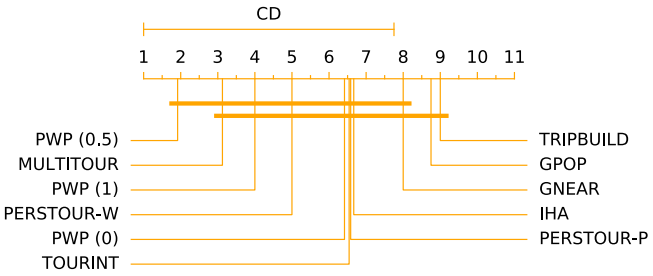

Fig. 9. CD diagrams (Precision) of the comparing algorithms.

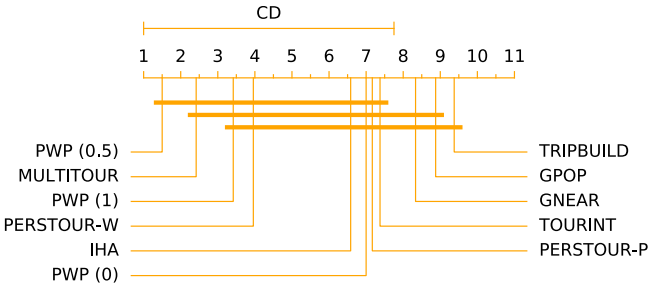

Fig. 10. CD diagrams (F1-Score) against various baseline approaches.

and supports multiple itineraries. On the other hand, the other baselines consider the attraction's popularity or interest of tourists and doesn't endorse a variety of itineraries. The improvement in *Precision* values for the proposed PWP algorithm varies from 6.3%-39.3% relative to considered baseline approaches. Also, while considering Delhi-Edinburgh, Osaka-Edinburgh, Delhi-Osaka, Glasgow-Edinburgh, and Vienna-Edinburgh, the *Precision* scores are more since it depends on $|C_{rec}|$ and $|C_{rec} \cap C_{real}|$ using Eqn. 34. During experimentation, we observed that the values of C_{rec} are different for different baselines. In the case of the proposed PWP algorithm, the values of $|C_{rec} \cap C_{real}|$ are greater.

The improvement in *F1-Score* values for the proposed PWP algorithm varies from 4.0%-23.5% relative to other baseline algorithms as *F1-Score* values depends on *Precision* and *Recall*. Therefore, it can be seen from Tables 3–5 that the proposed PWP approach performs well against comparison algorithm with respect to *Recall*, *Precision* & *F1-Score* as PWP support POI within POI and also suggest more number of itineraries.

We perform statistical performance tests on all versions of PWP algorithms as well as on the entire baseline. Table 6 provides the Friedman test result for precision, recall, and F1-Score statistics. It rejects a null hypothesis of the same tests at a significant point $\dot{\Gamma} = 0.05$. It means the pairs are contrasted with post hoc trials. The Nemenyi test is used to determine whether the proposed PWP achieves significant achievements compared to different baseline algorithms (Sarkar et al., 2020). $\dot{\Gamma} = 0.05, Q_{\dot{\Gamma}} = 3.219$. In this work $\mathscr{K}' = 11$, thus $CD = 4.354$. The results of any two approaches are different if they varies by at least the critical difference, $CD = Q_{\dot{\Gamma}}\sqrt{\frac{\mathscr{K}'(\mathscr{K}'+1)}{6\mathscr{N}'}}$. Fig. 8–10, are shown CD diagram for all the real-life metrics. On the left is the lower rank. It is evident from Fig.8–10, we can see clearly that, in terms of all real-life metrics, our proposed PWP outperforms all the baseline algorithms.

5.7. Comparative analysis of Interest, Popularity and Itinerary recommendations

Fig. 7 shows the graphic representation of A_i if the emphasis is given on interest, popularity and both with different fronts. Fig. 11 (a) depicts the comparative analysis of PWP approach against various baselines for interest of the visitor. On the other hand, considering the same dataset, Fig. 11 (b) depicts the comparative analysis of PWP algorithm against various baselines for tour popularity of the itineraries recommended to the tourist. All these figures have considered Buda-Edin dataset. In case of PWP algorithm, the total number of suggested itineraries are more. Hence, the value of interest and popularity are more in case of PWP algorithm. It is observed that PWP is the best performer and MULTITOUR is the second highest performer based on the values of popularity and interest. PWP algorithm prepares the itinerary by considering the POI inside the POI. As a result, the interest and popularity values of PWP are higher than *MULTITOUR*. The performance of the *IHA* and *PERSTOUR* algorithms is also good because the two algorithms are based on the popularity of POI and tourists interests. In the case of a PWP algorithm, it gives maximum importance to interest when $\Theta = 1$ and popularity when $\Theta = 0$. Due to the same reason, Fig. 12 (a), 13 (a) 14 (a), (a)(a)15–17 (a) and 12 (b), 13 (b), 14 (b), 15 (b), 16 (b), and 17 (b) are also shows a superior performance of the proposed PWP algorithm. It is evident from the above discussion that the PWP algorithm offers more interest and popularity with respect to the different baselines Considering the Buda-Edin dataset, Fig. 11 (c), depicts the performance of proposed PWP algorithm with all baselines using the total number of itineraries suggested to the tourist. In our experiment, 4 h is the travel duration by the tourist on an itinerary. For number of recommended itineraries, 5.0 is the best indicator, while 0.0 is the worst indicator. Based on this Recommended Itineraries, PWP out-performs the baseline algorithms. Due to a large number of GUs, the PWP algorithm recommends more itineraries than other baselines. On the other hand, the MULTITOUR algorithm also provides better performance as it supports multiple itinerary recommendations. The PWP algorithm prepares the

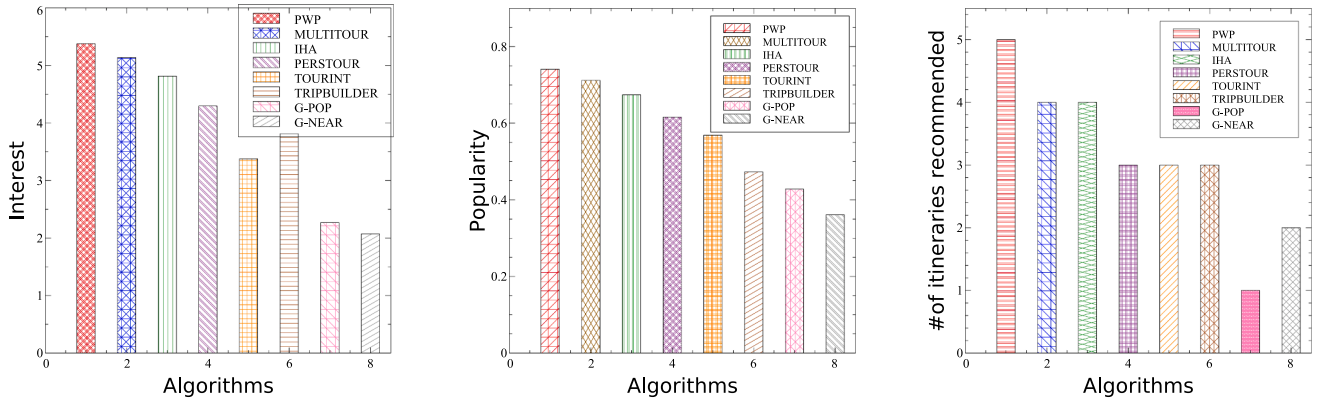

Fig. 11. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Buda-Edin datasets, with reference to various baseline approaches.

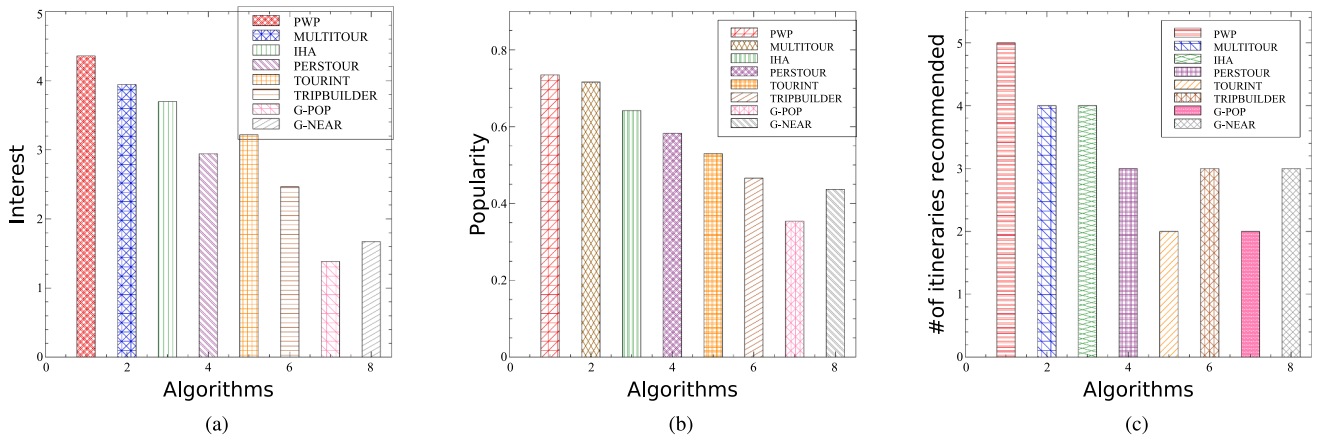

Fig. 12. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Delhi-Buda datasets, with reference to various baseline approaches.

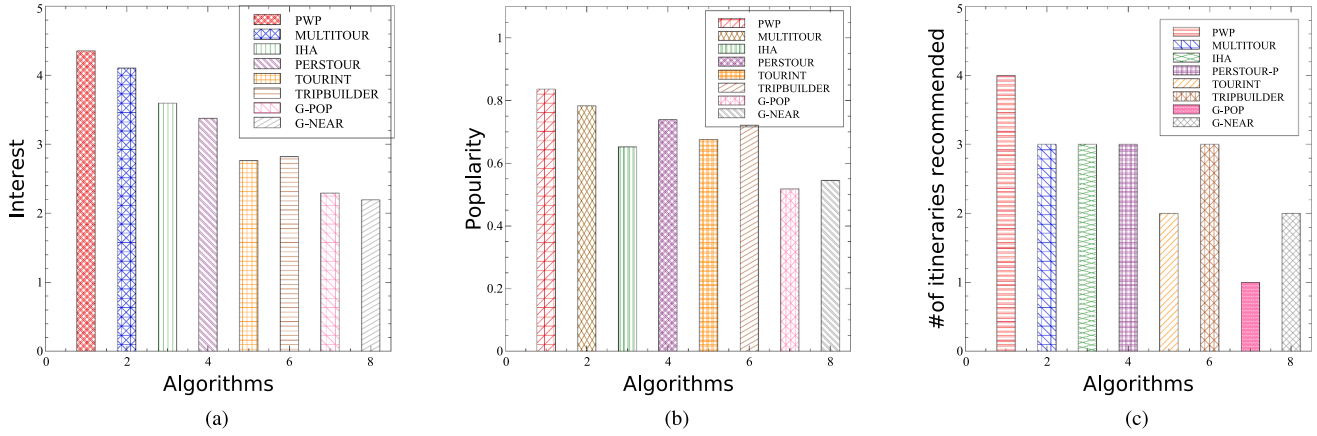

Fig. 13. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Delhi-Edin datasets, with reference to various baseline approaches.

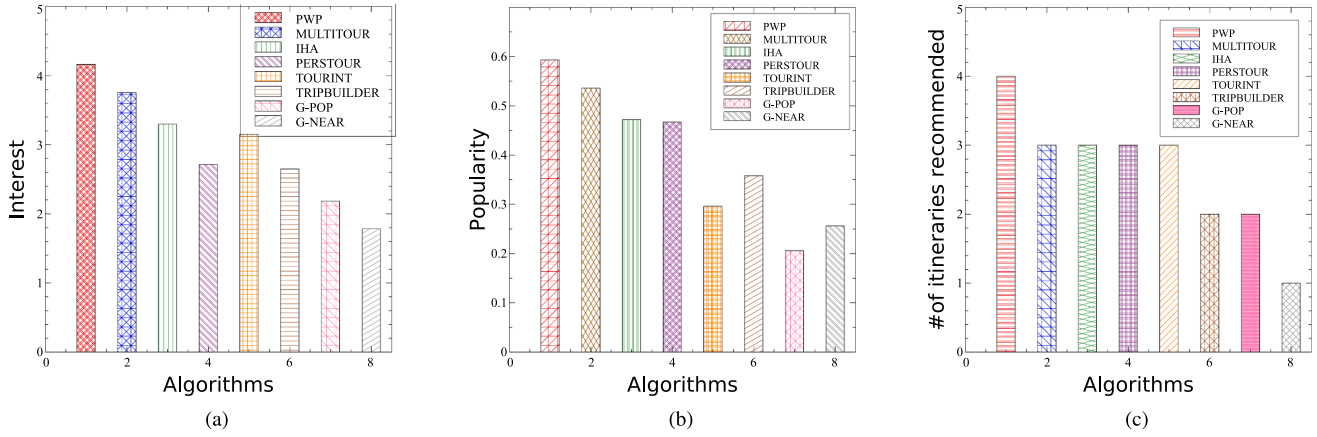

Fig. 14. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Delhi-Osaka datasets, with reference to various baseline approaches.

itinerary by considering the POI inside the POI. Hence, it recommends more number of itineraries with respect to MULTITOUR. Due to the same reason, Fig. 12 (c), 13 (c) 14 (c), 15 (c), 16(c) and 17 (c) also show a huge improvement in the performance of the proposed PWP scheme compared to other baseline algorithms.

5.8. *Prediction efficiency and correlation of throughput*

Considering the Budapest- Edinburgh Dataset, Fig. 18 (a) depicts a comparative analysis of proposed PWP algorithm with various baseline algorithms using MAE when there are changes in the number of itineraries. In this case, for MAE scores, 0.0 shows the highest performance, while 1.0 shows the lowest. The accuracy of the prediction is inversely

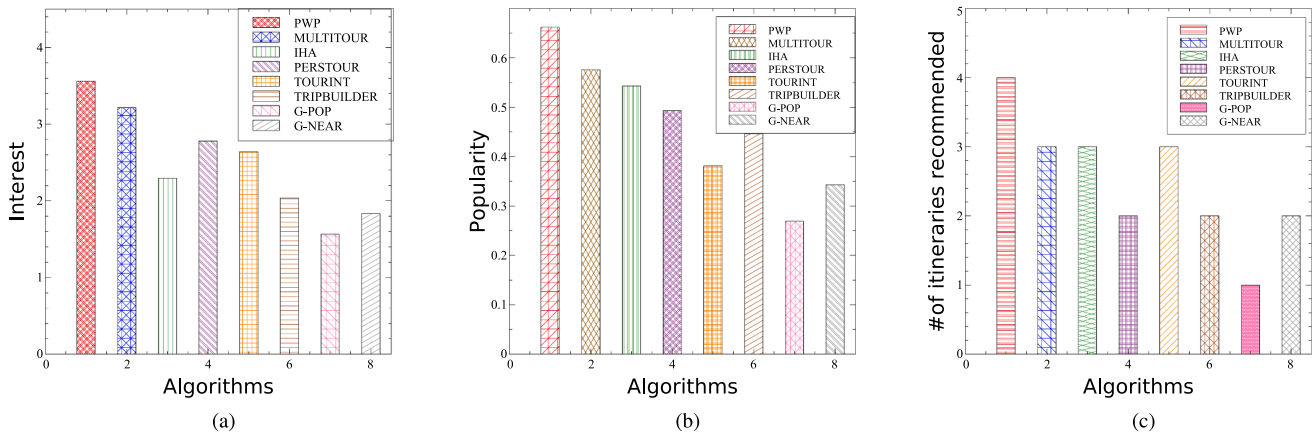

Fig. 15. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Glas-Edin datasets, with reference to various baseline approaches.

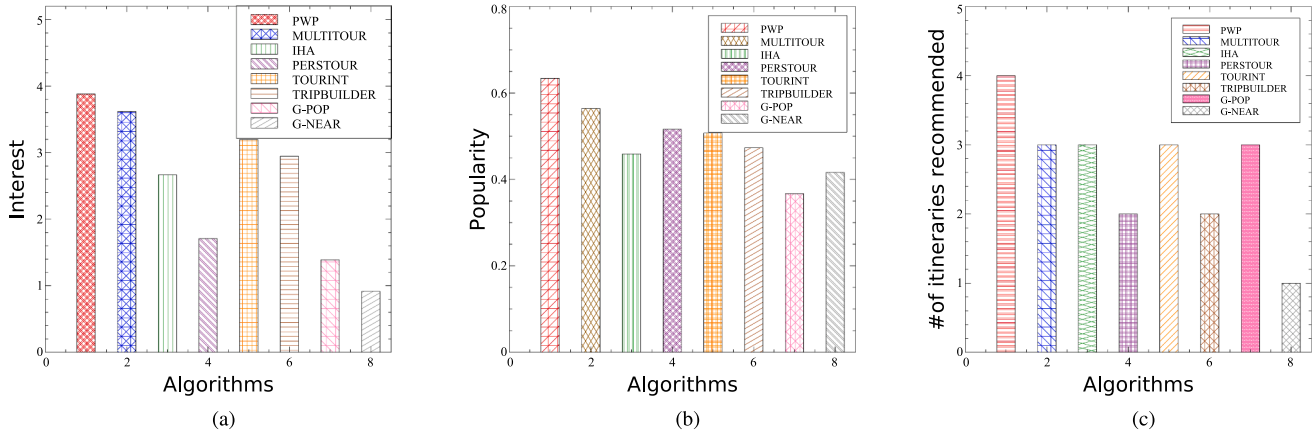

Fig. 16. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Osaka-Edin datasets, with reference to various baseline approaches.

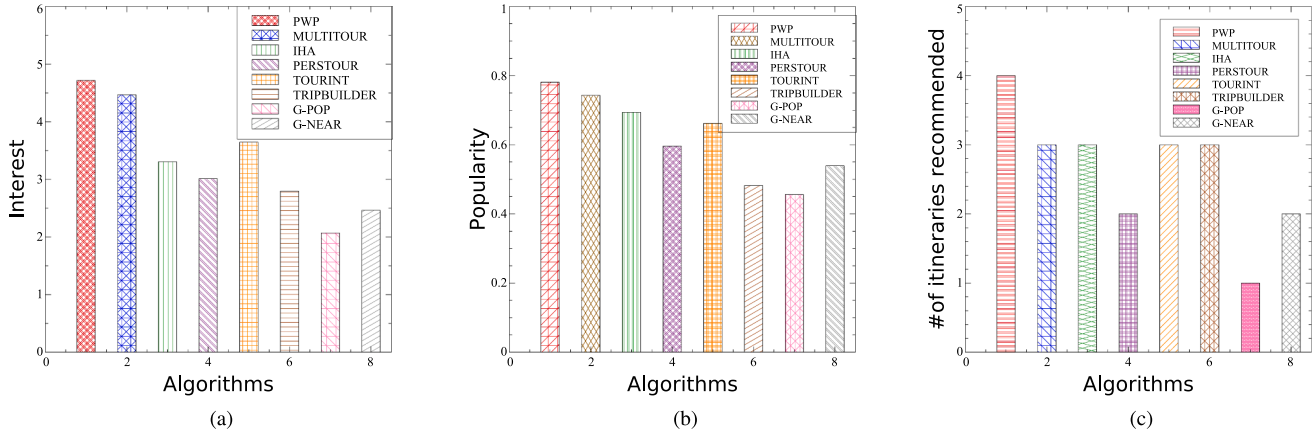

Fig. 17. Comparison of PWP for (a) Interest; (b) Popularity; (c) Number of recommended itineraries for Vien-Edin datasets, with reference to various baseline approaches.

proportional with MAE. PWP(0.5) is the best performer, as it considers interest and popularity in a balanced way using the concept of POI within POI. The MULTITOUR algorithm's efficiency is also better compared to baselines algorithms as this supports multiple itineraries. On the contrary, popularity and interest are given full attention by PWP (0) and PWP (1) respectively. Therefore, their efficiency is somewhere diminished. Fig. (b)18–23 are also shown the performance of PWP algorithm against different baselines, utilizing MAE with various datasets. The results show that the PWP algorithm achieves a significant consistency with respect various baselines.

After a certain itineraries, the output for different baselines are fixed at some stages, as only one itinerary is suggested by baseline algorithms, unlike the PWP and MULTITOUR algorithms. As comparison with various baselines, the accuracy of *IHA* and *PERSTOUR-W* algorithms is

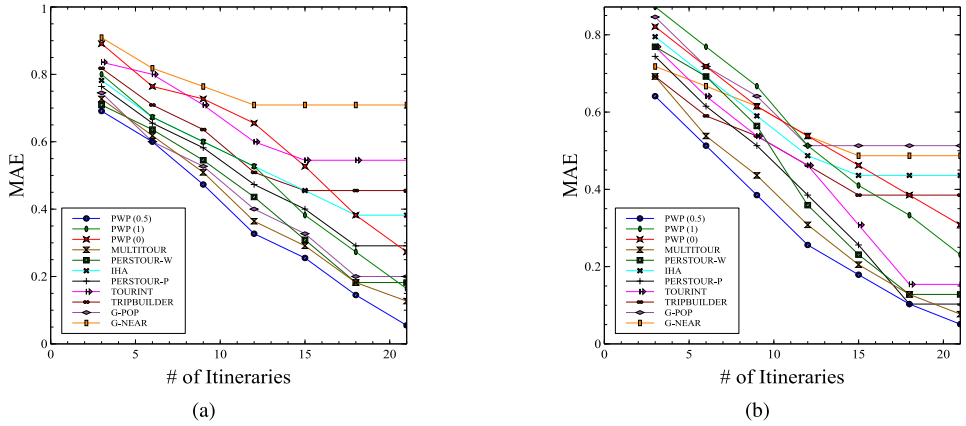

Fig. 18. PWP relative to multiple baseline approaches for *MAE*, (a) Budapest-Edinburgh data set, (b) Budapest-Glasgow data set.

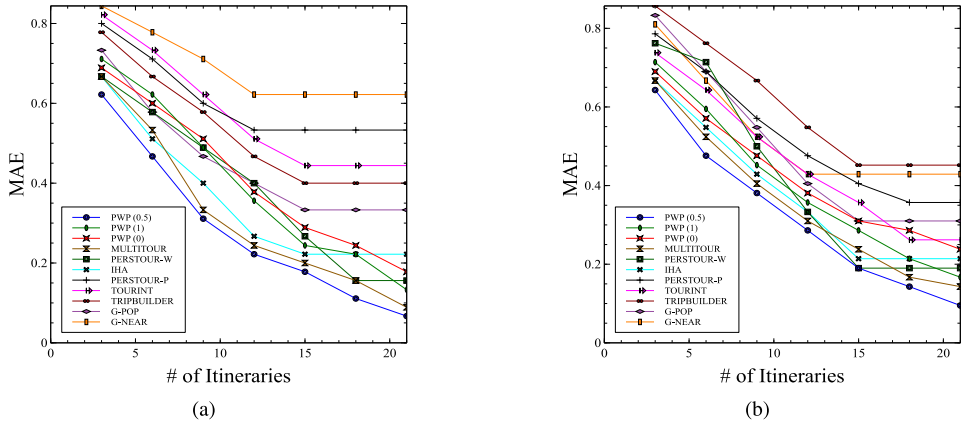

Fig. 19. PWP relative to multiple baseline approaches for *MAE*, (a) Budapest-Toronto data set, (b) Budapest-Vienna data set.

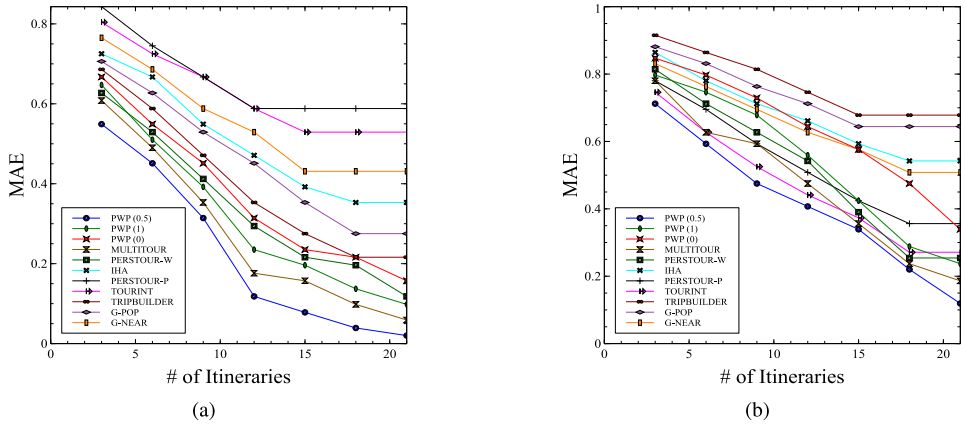

Fig. 20. PWP relative to multiple baseline approaches for *MAE*, (a) Delhi-Budapest data set, (b) Delhi-Edinburgh data set.

increased as both of the algorithms considered popularity of the POI and tourist's interests. Figs. 24 (a) depicts the throughput for proposed PWP algorithm. Here, 400 shows the highest throughput scores, while 0 indicates the worst output. PWP(0.5) is the most effective performer because of its balanced consideration of interest and popularity. The efficiency of *MULTITOUR* algorithms is also increased with respect to different baseline algorithms as this supports multiple itineraries.

Like MAE, *PERSTOUR-W* and *IHA* are also good in terms of throughput. The GU and POI within POI are taken into account in the PWP algorithm. So, in comparison to all baseline algorithms, the PWP algorithm has the highest throughput. Fig. (b)24–29 also depict the throughput of the suggested itineraries for different Dataset. From the above discussion it is clear that PWP has a greater throughput value than the other baseline algorithms.

6. Conclusion

The TRIPTOURREC issue was developed as an Orienteering problem. The PWP algorithm is designed to solve the Orienteering problem. Interest level of the tourist, tour popularity is maximized and total travel

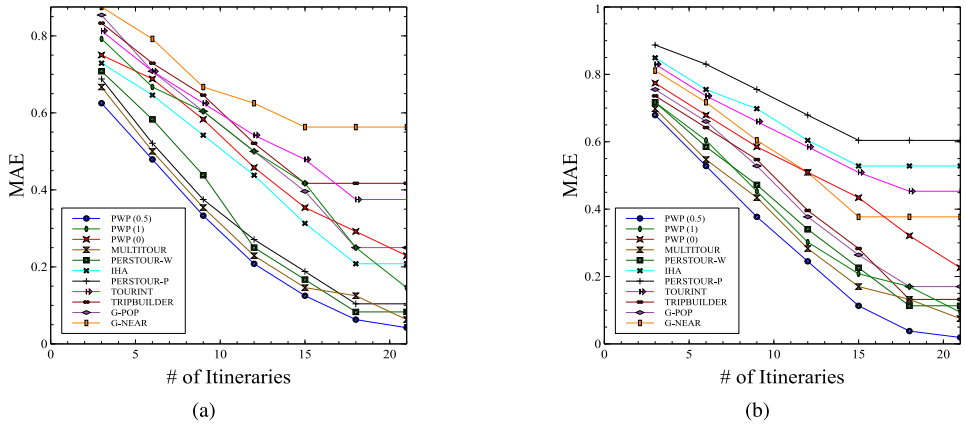

Fig. 21. PWP relative to multiple baseline approaches for *MAE*, (a) Delhi-Glasgow data set, (b) Delhi-Osaka data set.

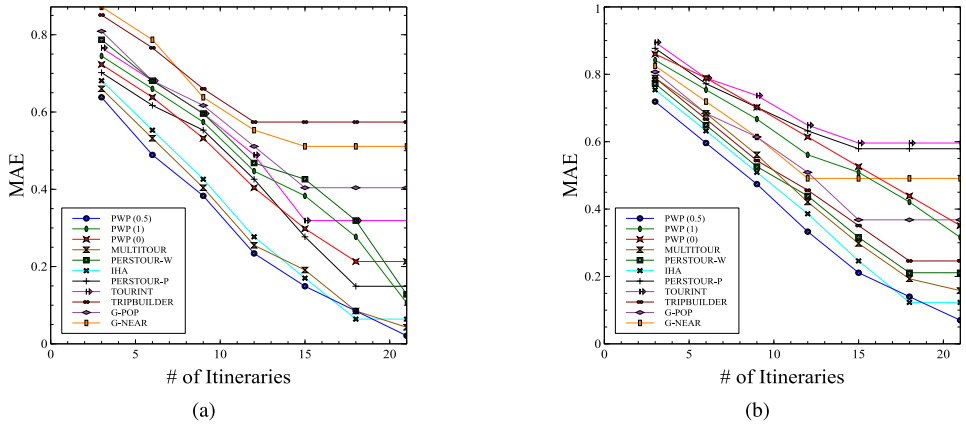

Fig. 22. PWP relative to multiple baseline approaches for *MAE*, (a) Delhi-Vienna data set, (b) Glasgow-Edinburgh data set.

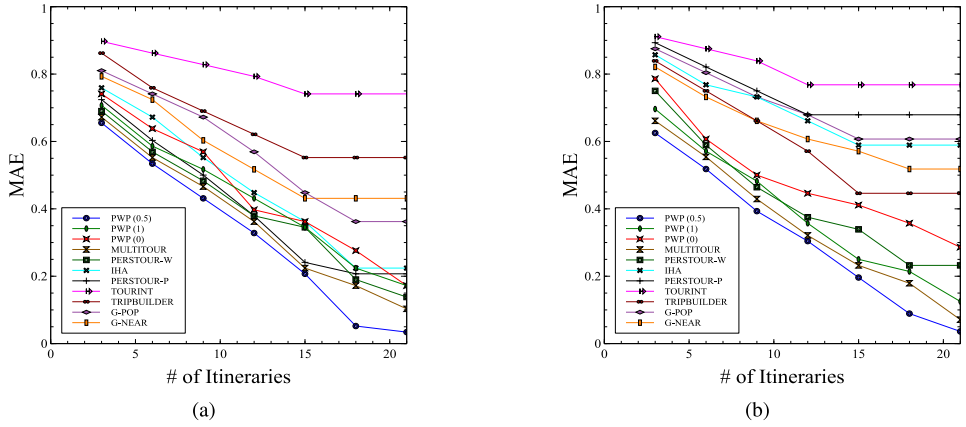

Fig. 23. PWP relative to multiple baseline approaches for *MAE*, (a) Osaka-Edinburgh, (b) Vienna-Edinburgh data set.

expenses are minimised by using the PWPalgorithm. Proposed algorithms use Geo-tagged images that provide real-life travel sequences for visitor. The PWP algorithm improvises the previous work on one major aspect, such as the collection of multiple itineraries, by considering the POI inside the POI. The PWP algorithm also takes into account the situation in which (i) visitors wishes to travel new sites. The proposed algorithm does not rely on a person's travel experience of new locations, (ii) the tourist has more than one itineraries, (iii) cost of travel is calculated for the tour. We compare our algorithm across various towns, with different baseline algorithms. The algorithm can be used by powerful smart systems to suggest multiple itineraries by considering POI-attraction as a bridge between calculation time and reliabilities for recommendation systems. The proposed technique has not considered other factors such as, traffic condition, weather condition etc. while recommending set of itineraries to the users. The proposed algorithm has been designed considering that POI-attractions belong to any one of the categories such as park, beach, temple etc. But it does not consider the scenario where a POI-attractions belong to multiple categories such as park and beach. Therefore, the algorithm can be extended to work under such scenarios in future. The algorithm can also be expanded to

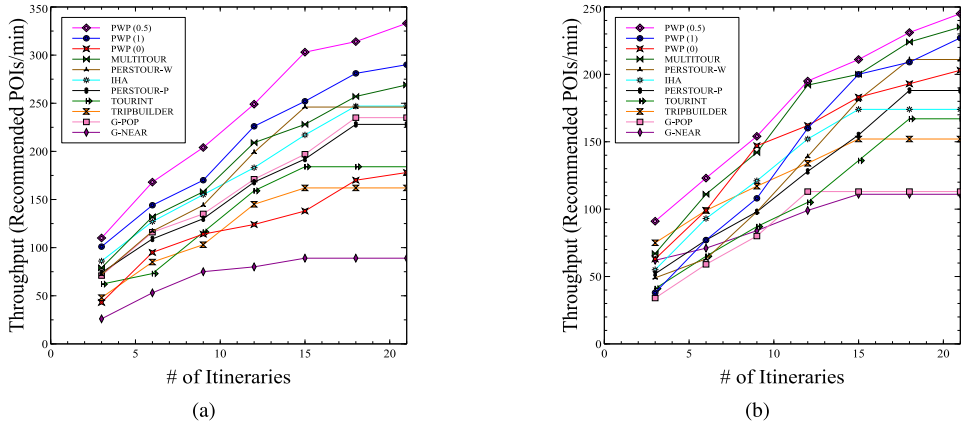

Fig. 24. PWP relative to multiple baseline approaches for *Throughput*, (a) Budapest-Edinburgh data set, (b) Budapest-Glasgow data set.

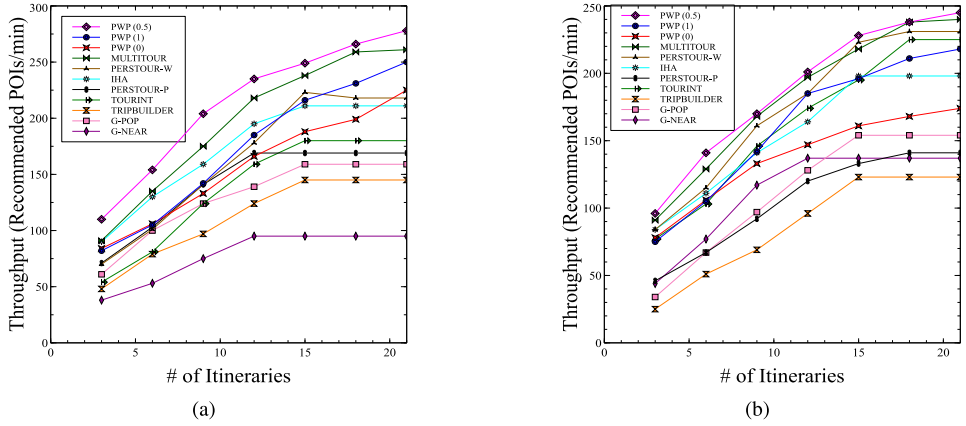

Fig. 25. PWP relative to multiple baseline approaches for *Throughput*, (a) Budapest-Toronto data set, (b) Budapest-Vienna data set.

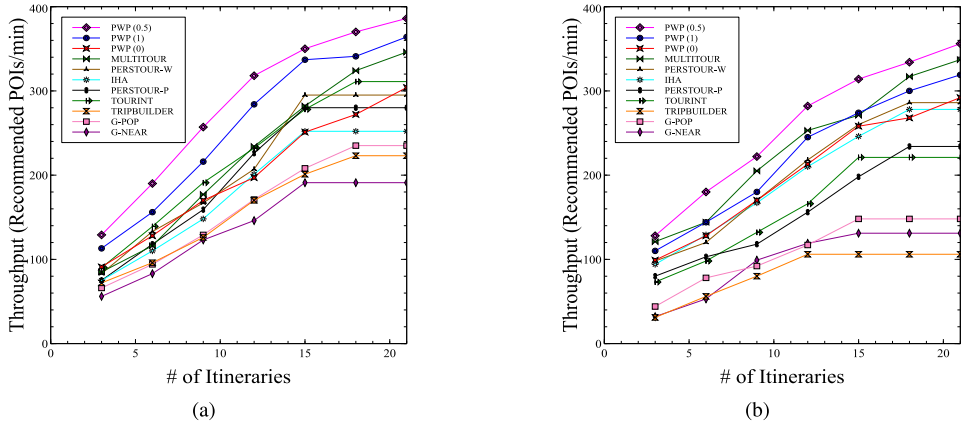

Fig. 26. PWP relative to multiple baseline approaches for *Throughput*, (a) Delhi-Osaka data set, (b) Delhi-Vienna data set.

provide suggestions for products and films. Nonetheless, in these cases user feedback needs to be taken into consideration instead of geo tagged photos. In our future work, we intend to improve the proposed work for a group of tourists who are planning to spend a few days together.

CRediT authorship contribution statement

Joy Lal Sarkar: Conceptualization, Writing - original draft, Writing - review & editing, Visualization. **Abhishek Majumder:** Conceptualization, Methodology, Supervision, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

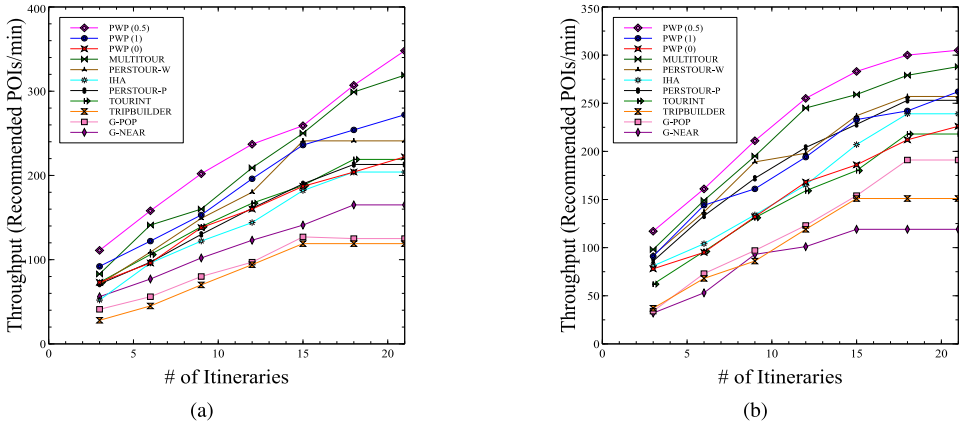

Fig. 27. PWP relative to multiple baseline approaches for *Throughput*, (a) Delhi- Edinburgh data set, (b) Delhi-Glasgow data set.

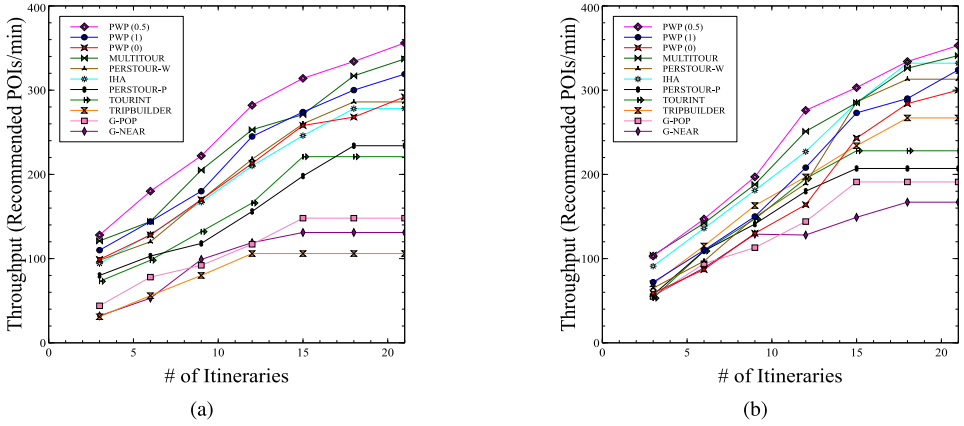

Fig. 28. PWP relative to multiple baseline approaches for *Throughput*, (a) Delhi-Vienna data set, (b) Glasgow-Edinburgh data set.

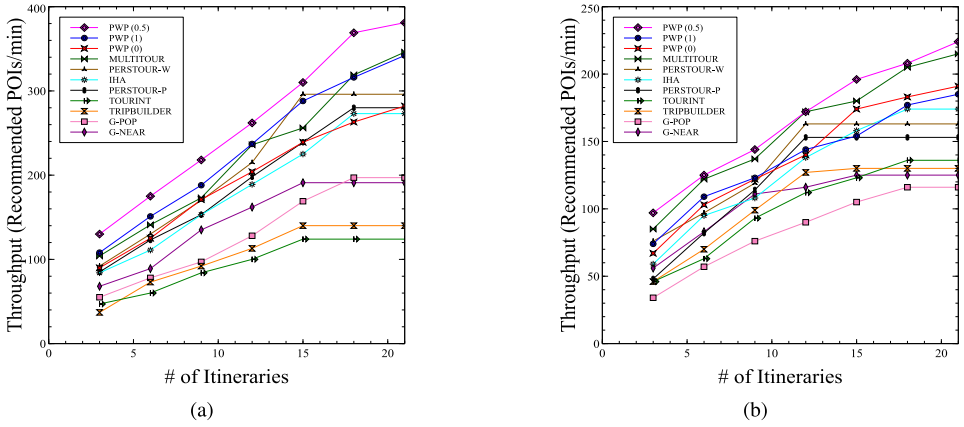

Fig. 29. PWP relative to multiple baseline approaches for *Throughput*, (a) Osaka-Edinburgh data set, (b) Vienna-Edinburgh data set.

Acknowledgements

We would like to acknowledge the Mobile Computing Laboratory, Department of Computer Science and Engineering, Tripura University, Tripura, India for providing the necessary infrastructure.

References

Anagnostopoulos, A., Atassi, R., Becchetti, L., Fazzone, A., & Silvestri, F. (2017). Tour recommendation for groups. *Data Mining and Knowledge Discovery, 31*(5), 1157–1188.

Baraglia, R., Muntean, C. I., Nardini, F. M., & Silvestri, F. (2013). Learnext: Learning to predict tourists movements. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management CIKM '13 (pp. 751–756). New York, NY, USA: ACM.

Brilhante, I. R., Macedo, J. A., Nardini, F. M., Perego, R., & Renso, C. (2015). On planning sightseeing tours with tripbuilder. *Information Processing & Management, 51*(2), 1–15.

Brilhante, I. R., de Macêdo, J. A. F., Nardini, F. M., Perego, R., & Renso, C. (2014). Tripbuilder: A tool for recommending sightseeing tours. In Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13–16, 2014. Proceedings (pp. 771–774).

Brudy, F., Ledo, D., Greenberg, S., & Butz, A. (2014). Is anyone looking? mitigating shoulder surfing on public displays through awareness and protection. In

Proceedings of The International Symposium on Pervasive Displays PerDis '14 (pp. 1:1–1:6). New York, NY, USA: ACM.
Chen, B., Yu, S., Tang, J., He, M., & Zeng, Y. (2017). Using function approximation for personalized point-of-interest recommendation. *Expert Systems with Applications, 79*, 225–235.
Chen, C., Zhang, D., Guo, B., Ma, X., Pan, G., & Wu, Z. (2015). Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Transactions on Intelligent Transportation Systems, 16*(3), 1259–1273.
Chen, L., Zhang, L., Cao, S., Wu, Z., & Cao, J. (2020). Personalized itinerary recommendation: Deep and collaborative learning with textual information. *Expert Systems with Applications, 144*, Article 113070.
Clair E. Miller, A. W. T., & Zemlin, R. A. (1960). Journal of the ACM, 7(4), 326–329.
De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., & Yu, C. (2010). Automatic construction of travel itineraries using social breadcrumbs. In Proceedings of the 21st ACM Conference on Hypertext and Hypermedia HT '10 (pp. 35–44). New York, NY, USA: ACM.
Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.
Dietz, L. W., & Weimert, A. (2018). Recommending crowdsourced trips on wondary. In RecSys 2018.
Expósito, A., Mancini, S., Brito, J., & Moreno, J. A. (2019). A fuzzy grasp for the tourist trip design with clustered pois. *Expert Systems with Applications, 127*, 210–227.
Gu, J., Song, C., Jiang, W., Wang, X., & Liu, M. (2020). Enhancing personalized trip recommendation with attractive routes. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(01), 662–669.
Herzog, D., Promponas-Kefalas, N., & Wörndl, W. (2018). Integrating public displays into tourist trip recommender systems. In Proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018), Vancouver, Canada, October 7, 2018. (pp. 18–22).
Kapcak, Ö., Spagnoli, S., Robbemond, V., Vadali, S., Najafian, S., & Tintarev, N. (2018). Tourexplain: A crowdsourcing pipeline for generating explanations for groups of tourists. In Proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018), Vancouver, Canada, October 7, 2018. (pp. 33–36).
Kotiloglu, S., Lappas, T., Pelechrinis, K., & Repoussis, P. (2017). Personalized multi-period tour recommendations. *Tourism Management, 62*, 76–88.
Lee, H., Chung, N., & Nam, Y. (2019). Do online information sources really make tourists visit more diverse places?: Based on the social networking analysis. *Information Processing & Management, 56*(4), 1376–1390.
Li, L., Lee, K. Y., & Yang, S.-B. (2019). Exploring the effect of heuristic factors on the popularity of user-curated 'best places to visit' recommendations in an online travel community. *Information Processing & Management, 56*(4), 1391–1408.
Lim, K. H. (2015). Recommending tours and places-of-interest based on user interests from geo-tagged photos. In Proceedings of the 2015 ACM SIGMOD on PhD Symposium (pp. 33–38). New York, NY, USA: ACM.
Lim, K. H., Chan, J., Karunasekera, S., & Leckie, C. (2017). Personalized itinerary recommendation with queuing time awareness. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '17 (pp. 325–334). New York, NY, USA: ACM.
Lim, K. H., Chan, J., Karunasekera, S., & Leckie, C. (2019). Tour recommendation and trip planning using location-based social media: a survey. *Knowledge and Information Systems, 60*.
Lim, K. H., Chan, J., Leckie, C., & Karunasekera, S. (2018). Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems, 54*, 375–406.
Lim, K. H., Wang, X., Chan, J., Karunasekera, S., Leckie, C., Chen, Y., Loong Tan, C., Quan Gao, F., & Ken Wee, T. (2016). Perstour: A personalized tour recommendation and planning system.
Lucchese, C., Perego, R., Silvestri, F., Vahabi, H., & Venturini, R. (2012). How random walks can help tourism. In Proceedings of the 34th European Conference on Advances in Information Retrieval ECIR'12 (pp. 195–206). Berlin, Heidelberg: Springer-Verlag.
Mottini, A., Lheritier, A., Acuna-Agost, R., & Zuluaga, M. A. (2018). Understanding customer choices to improve recommendations in the air travel industry. In: Workshop on Recommenders in Tourism, pp, (pp. 28–32).
Palumbo, E., Rizzo, G., Troncy, R., & Baralis, E. (2017). Predicting your next stop-over from location-based social network data with recurrent neural networks. In *In Proceedings of the ACM RecSys Workshop on Recommenders in Tourism ser. RecSys*.
Qiao, Y., Luo, X., Li, C., Tian, H., & Ma, J. (2020). Heterogeneous graph-based joint representation learning for users and pois in location-based social network. *Information Processing & Management, 57*(2), Article 102151.
Quercia, D., Schifanella, R., & Aiello, L. M. (2014). The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In Proceedings of the 25th ACM Conference on Hypertext and Social Media HT '14 (pp. 116–125). New York, NY, USA: ACM.
Sarkar, J. L., Majumder, A., Panigrahi, C. R., & Roy, S. (2020). Multitour: A multiple itinerary tourists recommendation engine. *Electronic Commerce Research and Applications, 40*, Article 100943.
Seo, Y.-D., & Cho, Y.-S. (2021). Point of interest recommendations based on the anchoring effect in location-based social network services. *Expert Systems with Applications, 164*, Article 114018.
Shi, Y., Serdyukov, P., Hanjalic, A., & Larson, M. (2011). Personalized landmark recommendation based on geotags from photo sharing sites. In *In Fifth International AAAI Conference on Weblogs and Social Media* (pp. 622–625).
Subashini, G., & Bhuvaneswari, M. (2013). Comparison of multi-objective evolutionary approaches for task scheduling in distributed computing systems. *Sadhana, 37*.
Thomee, B., Shamma, D., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., & Li, L.-J. (2016). The new data and new challenges in multimedia research. *Communications of the ACM, 59*(2), 64–73.
Trachanatzi, D., Rigakis, M., Marinaki, M., & Marinakis, Y. (2020). An interactive preference-guided firefly algorithm for personalized tourist itineraries. *Expert Systems with Applications, 159*, Article 113563.
Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Oudheusden, D. V. (2011). The city trip planner. *Expert Systems with Applications, 38*(6), 6540–6546.
Vu, H. Q., Li, G., & Law, R. (2019). Discovering implicit activity preferences in travel itineraries by topic modeling. *Tourism Management, 75*, 435–446.
Wang, X., Leckie, C., Chan, J., Lim, K. H., & Vaithianathan, T. (2016). Improving personalized trip recommendation by avoiding crowds. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management CIKM '16 (p. 25–34). New York, NY, USA: Association for Computing Machinery.
Yamasaki, T., Gallagher, A., & Chen, T. (2013). Personalized intra- and inter-city travel recommendation using large-scale geotags. In Proceedings of the 2nd ACM International Workshop on Geotagging and Its Applications in Multimedia GeoMM '13 (pp. 25–30). New York, NY, USA: ACM.
Zhang, C., Liang, H., & Wang, K. (2016). Trip recommendation meets real-world constraints: Poi availability, diversity, and traveling time uncertainty. ACM Trans. Inf. Syst., 35(1), 5:1–5:28.
Zheng, W., Liao, Z., & Lin, Z. (2020). Navigating through the complex transport system: A heuristic approach for city tourism recommendation. *Tourism Management, 81*, Article 104162.