Itinerary Recommendation Algorithm in the Age of MEC

Sand Luz Correa, Kleber Vieira Cardoso, Felipe F Fonseca, Lefteris Mamatas, Aline Carneiro Viana

Itinerary Recommendation Algorithm in the Age of MEC

Sand L. Correa, Kleber V. Cardoso, Felipe F. Fonseca, Lefteris Mamatas, and Aline C. Viana

Abstract—To provide fully immersive mobile experiences, next-generation touristic services will rely on the high bandwidth and low latency provided by the 5G networks and the Multi-access Edge Computing (MEC) paradigm. Recommendation algorithms, being integral part of travel planning systems, devise personalized tour itineraries for a user considering the popularity of the Points of Interest (POIs) of a city as well as the tourist preferences and constraints. However, in the context of next-generation touristic services, recommendation algorithms should also consider the applications (e.g., augmented reality) the tourist will consume in the POIs and the quality in which such applications will be delivered by the MEC infrastructure. In this paper, we address the joint problem of recommending personalized tour itineraries for tourists and efficiently allocating MEC resources for advanced touristic applications. We formulate an optimization problem that maximizes the itinerary score of individual tourists, while optimizing the resource allocation at the network edge. We then propose an exact algorithm that quickly solves the problem optimally considering instances of realistic size. Finally, we evaluate our algorithm using a real dataset extracted from Flickr. Results demonstrate gains up to 100% in the resource allocation and user experience in comparison with a state-of-the-art solution.

Index Terms—Travel itinerary recommendation, Next-generation touristic services, Multi-access edge computing, 5G networks.

1 Introduction

THe advent of 5G networks and relevant communication paradigms, such as the Multi-Access Edge Computing (MEC) [1] and Network Slicing [2], will drive new business models in nearly every vertical industry. Together, these technologies can provide extensive coverage, higher capacity, increased reliability, and low latency to a large number of connected devices on the top of a shared infrastructure. The new opportunities will impact the entire mobile network ecosystem. For example, end-users will be able to experience new services that could not be satisfied with current technologies; service providers will be able to tailor network services to their specific needs, while mobile network operators will be able to target new markets.

An example of such a market is Tourism, an important sector that accounts for nearly 10% of the world's Gross Domestic Product [3]. Indeed, 5G is foreseen as an essential enabler for next-generation touristic services, as demonstrated by existing initiatives from academic institutions, mobile network operators and public sectors [4], [5]. With the improved communication and computation capabilities provided by 5G, tourists will be able to benefit from immersive technologies (e.g., virtual, augmented, or mixed reality – VR/AR/MR) to augment and extend their sight-seeing activities, enjoy integrated transport, accommodation, and entertainment services, as well as to communicate with other tourists using new social-networking based ways. Consequently, tourists will be seeking an advanced, combined physical, and virtual visiting experience.

On the other hand, algorithms for recommending personalized tour itineraries have become integral components of travel planning systems. Given the tourist interests and preferences, the goal of such algorithms is to improve the touristic experience by identifying a set of Points of Interest (POIs) and schedule them as an itinerary with various time and space constraints [6]. To accomplish this task, algorithms for recommending tour itineraries are often proposed as a combinatorial problem called the *Orienteering Problem* (OP) [7]. Here, given a graph with nodes representing the POIs of the city, edges being the travel routes between nodes, with costs on the edges and profits on the vertices, the algorithms find an itinerary with maximum profit for the tourist considering factors such as the popularity of the POIs, the tourist preferences, the distance between the POIs, and the tourist time limit to complete the tour (i.e., the tourist time budget). Indeed, the problem of recommending tour itineraries is not new and has been well-studied. The authors in [8], [9] present comprehensive surveys of heuristic and algorithmic approaches to the problem, while the works in [6], [10], [11], [12] focus on data-driven approaches to model the problem by extracting visiting-related statistics (e.g., the popularity of the POIs, the duration of visits, and the user interest on the category of the POI) from location-based social media such as Facebook, Flickr, and Foursquare.

However, to fulfill the physical and virtual expectations of tourists in next-generation touristic services, recommendation algorithms will also have to consider (i) the set of service applications the tourists will consume in the POIs and (ii) the quality in which such services will be delivered, given the available resources in the Information and Communication Technology (ICT) infrastructure during the visits. Consider, for example, a Metropolitan Tourist Centre (MTC) that provides next-generation touristic services for its

Felipe Fonseca, Sand Correa and Kleber Cardoso are with the Instituto de Informática, Universidade Federal de Goiás, Goiânia - GO, Brazil (e-mails: fonsecafel@gmail.com, sand@inf.ufg.br, kleber@inf.ufg.br).
Lefteris Mamatas is with University of Macedonia, Thessaloniki, Greece (email:emamatas@uom.edu.gr).
Aline C. Viana is with Inria, Palaiseau, France (email: aline.viana@inria.fr).

tourists. Consider that the MTC provides a recommendation system to help the tourists to build their tour itineraries. The rough idea is that the recommendation system does not necessarily issue recommendations for the top-ranked itinerary according to the preferences of each user. Instead, it recommends itineraries to individual tourists that still match adequately with their preferences and, at the same time, optimize the ICT resource allocation at the network edge, when running the selected service applications. In this way, requirements involving both users' preferences and MEC resource allocation are satisfied.

Although such recommendations can slightly affect the recommended POIs, they aim at higher MEC resource allocation efficiency and, thus, a better global physical and virtual experience. To the best of our knowledge, only our earlier conference paper [13] focuses on the problem of recommending personalized itineraries for tourists taking into consideration the availability of the edge resources to run advanced touristic service applications. We here build on this prior effort by presenting a much more comprehensive investigation and offering the following contributions:

- We discuss related works and introduce our contributions to the literature (cf. Section 2).
- We define a system model that captures the joint problem of (i) recommending personalized tour itineraries for multiple tourists and (ii) the efficient allocation of MEC resources that enables advanced touristic service applications (cf. Section 3). Our system model considers the popularity of the POIs, the preferences and time constraint of each tourist, the demands of the chosen service applications, and the availability of network/computing resources when deciding which itineraries to recommend.
- We define a novel problem, called MEC-aware Personalized Tour Recommendation (MEC-PTR), that formalizes the problem of recommending personalized tour itineraries for multiple tourists in the context of next-generation touristic services at the network edge (cf. Section 4). Given the set of POIs in a city and the interests of the tourists, the goal of the MEC-PTR problem is to find a set of itineraries (containing exactly one itinerary for each tourist) that maximizes the sum of the profits perceived by the tourists and prioritizes the ICT resource allocation at the network edge, while satisfying the tourists and infrastructure (service/ICT resources) constraints.
- Since the MEC-PTR problem is NP-hard, we devise an algorithm, named MEC-aware Personalized Itinerary Recommendation (MEC-PIR), that efficiently solves the MEC-PTR problem using a Dynamic Programming approach and Mixed Integer Linear Programming (MILP) (cf. Section 5).
- We evaluate the effectiveness of MEC-PIR using a real dataset extracted from Flickr geotagged photos and Wikipedia. To better understand this dataset, we provide an analysis of the visiting-related statistics extracted from it (cf. Section 6). Although the methodology to extract visiting information from Flickr has been proposed and used in other works [6], [10], [11], to the best of our knowledge, none of them provide an exploratory analysis of the obtained dataset.
- We compare MEC-PIR with a modified version of a state-of-the-art solution for recommending personalized tour itinerary, the PersTour algorithm [6], which we named Resource-Aware PersTour (RA-PersTour) (cf. Section 6). To this end, to take into account the tourist experience, we propose a new metric that combines traditional recommendation scoring with attendance of service application demands. Results show that MEC-PIR achieves gains from near 20% up to 100% in both aspects, i.e., resource allocation efficiency and user experience while presenting a similar performance to RA-PersTour in traditional metrics for the evaluation of personalized itinerary recommendation. Results also show that MEC-PIR can solve the MEC-PTR problem optimally for instances of realistic size, in a reasonable amount of time.

Conclusions and future work directions are presented in Section 7.

2 Related work

Broadly, works focused on generating tour itineraries can be classified into two sub-areas [14]: *Operational Research* and *Location-based Recommendation*. Next, we discuss relevant works in both sub-areas.

Operational Research. Indeed, the problem of recommending personalized tour itineraries has originated in the Operational Research community as the Tourist Trip Design Problem (TTDP). The authors in [15] have defined the generic TTDP as an extension of the Orienteering Problem (OP), where the objective is to schedule an optimal path (i.e., the path that maximizes the collected profit) for the tourist, given the set of candidate POIs and the travel time among them; the profit of each POI; the visiting duration at a POI; starting/ending POIs for the tour; and the tourist time budget. A comprehensive survey on the use of OP and Travel Salesman Problem (TSP) to model multiple variants of the TTDP was presented in [8] and [9]. An important drawback of these works, however, is the lack of individual preferences associated with individual tourists. As a consequence, the same tour itinerary is recommended to all tourists that provide as input the same starting/ending POIs and time budget.

Location-based Recommendation. With the popularity of smartphones and the prevalence of location-based social media, several works have focused on data-driven approaches to better model the tourist interest preferences when recommending POIs/itineraries. The main idea of these works is to exploit the tourism content, available in such services, to enable a deeper understanding of the user preferences and to issue personalized recommendations that satisfy these preferences. The works on location-based recommendation can be divided into three groups [14]: *next POI recommendation*, *top-k POI recommendation*, and *tour itinerary recommendation*. Next POI recommendation [16], [17] aims to identify the next location that a tourist is likely to visit based on her previous trajectories. Works on top-k location recommendation [18], [19] recommend multiple POIs as part of a ranked list, but they do not structure these POIs as a connecting itinerary. Finally, works on tour itinerary recommendation [6], [10], [11], [12], [20], [21] recommend a sequence of POIs as a connecting path taking into account the interest preferences of the tourists, while adhering to the several temporal and spatial constraints. Works on tour itinerary recommendation can be further divided into those that focus on a *group of tourists* [20], [21] and those aiming at *individual travelers* [6], [10], [11], [12]. A survey on recommending tour itineraries using location-based social media can be found in [14]. Since our work falls into the category of location-based tour itinerary recommendation, in the following, we pinpoint some relevant works related to this topic.

Location-based Tour Itinerary Recommendation. The authors in [10] are one of the first to combine an optimization-based approach and the mining of tourist past trajectories based on geotagged photos, extracted from Flickr, to recommend personalized tour itineraries for an individual tourist. This combined approach was further refined in [6], [11] by assigning categories to POIs and using these POI categories to determine the interests of the tourist. Particularly, the authors in [11] model the personalized tour recommendation for an individual tourist as a Generalized Maximum Coverage problem that optimizes for the POI popularity and the tourist interest. The latter is computed as the number of times the tourist has visited a POI of a certain category, relative to her total POI visits (frequency-based). PersTour [6] follows a similar approach, but it models the personalized tour recommendation as an OP and the tourist interest levels are computed based on the amount of time the tourist stays in a POI relative to the average tourist (time-based). In [6], it is shown that the time-based tourist interest approach reflects more accurately the real-life tours of users when compared to frequency-based tourist interest. The problem of recommending tour itineraries for an individual tourist where the trip duration takes more than a day is tackled in [12], where the authors use an anonymized Google historical visit dataset to mine tourist interests. Recommending tour itineraries for a group of tourists engaged in temporary mobile social networks (e.g., created for a conference/meeting) is the main focus of the work in [21].

Our contribution. Since our work focuses on recommending personalized tour itineraries for a one-day trip, it is conceptually more close to the investigations performed in [6], [10], [11]. Similar to these works, we also obtain POI visit information and tourist interests from past histories extracted from Flickr. However, we are distinctly different from all the previous works, since we consider the joint problem of recommending personalized tour itineraries for multiple tourists and the efficient allocation of MEC resources that enables advanced touristic service applications to be used by them to enrich their touristic experience. Thus, we introduce a novel multi-tourist perspective and recommend tour itineraries based on users' preferences, the network/computing demand of the consumed service applications, and the availability of the MEC infrastructure.

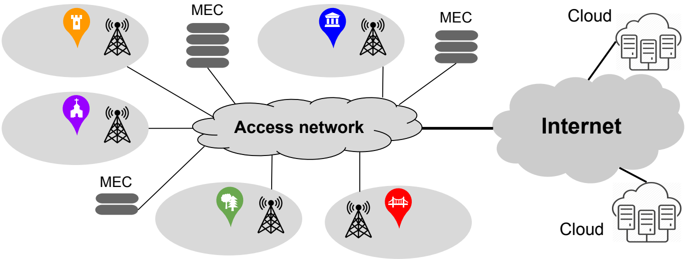

Fig. 1. Tourist spots with ICT infrastructure.

3 System model

We consider a Metropolitan Tourist Centre (MTC) that aims to provide next-generation touristic services to enhance the experiences of tourists visiting a specific city. The MTC leases ICT resources from a mobile network operator in the form of a network slice. The network slice is composed of wireless coverage in each POI, MEC host capabilities near to the POIs, and remote cloud computing resources accessed through the Internet. Figure 1 illustrates the described scenario, in which the MTC manages only the virtualized infrastructure built for its demand, i.e., the MTC controls only its own network slice.

We assume that MTC provides a recommendation system helping tourists to build their tour itineraries. Tourists access the system before their trips, providing information such as the day they wish the tour to take place (we consider that the tourist is interested in a one-day tour), preferred starting and ending location, time limit to complete the tour, and their preferences concerning: (i) the categories of the POIs (e.g., Shopping, Entertainment, Historical, Museum, Sport, Park), and (ii) the set of service applications offered by the MTC (e.g., Augmented reality, Video streaming). Daily, the system retrieves all planned tours targeting the next day and generates a personalized tour itinerary for each tourist, detailing information about the POI visit sequence and the appropriate time to arrive at and depart from each POI. The objective of the system is to find a set of itineraries that satisfies the following requirements:

- **R1**: the generated set contains exactly one itinerary for each tourist and this itinerary respects the time constraint of the tourist;
- **R2**: the amount of network and computing resources allocated for each itinerary takes into account the minimum and maximum demands of the service applications chosen by the tourist;
- **R3**: the sum of the amount of resources allocated for the itineraries does not exceed network and computing capacity in the POIs at any given time of the tours;
- **R4**: the generated set maximizes the sum of the profits perceived by the tourists while prioritizing the MEC resource allocation.

Table 1 summarizes the notation used in the system model (Section 3) and the problem formulation (Section 4).

3.1 Main elements of the model

3.1.1 *POIs*

For a city with $|\mathcal{V}|$ POIs, we consider a complete non-oriented graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$ being the set of vertices representing the POIs and $\mathcal{E} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ is the set of edges connecting the nodes. Each edge $(v_i, v_j) \in \mathcal{E}$ is associated with a cost $c_{i,j}$ representing the travel time between vertex v_i to vertex v_j, employing a given mode of transportation. Each POI $v_i \in \mathcal{V}$ is characterized by the following attributes. The popularity of the POI, denoted by $Pop(v_i) \in \mathbb{Z}$. The expected time one should spend in the POI to enjoy what it has to offer (i.e., expected POI visiting time), denoted by $Dur(v_i) \in \mathbb{R}$. The category representing its nature (e.g., Shopping, Entertainment, Historical, Museum, Sport, Park), denoted by $Cat(v_i)$. Finally, the location expressed in terms of latitude and longitude, denoted by $Lat(v_i)$ and $Long(v_i)$ respectively. We denote by $\mathcal{C} = \{1, \ldots, C\}$ the set of all POI categories, so that $Cat(v_i) \in \mathcal{C}$.

3.1.2 Service Applications and ICT Infrastructure

The MTC offers a set $\mathcal{A} = \{1, \ldots, A\}$ of service applications to enrich the touristic experience in every POI. Each service $a \in \mathcal{A}$ has particular requirements in terms of network and computing resources, expressed as:

- λ_a^{min} and λ_a^{max}: the minimum and maximum network demand (i.e., demanded traffic volume) in bps, respectively;
- ψ_a^{min} and ψ_a^{max}: the minimum and maximum computing demand (i.e., demanded processing load) measured in reference core (RC)[1], respectively.

In each POI, the network resources are provided through a set of wireless access points or a base station. In order to keep the model generic, we decide to represent network resources simply by the capacity, i.e., not taking into account the wireless channel aspects. In fact, the whole model complexity and the time scale of the problem also suggest that this is a reasonable approach. We denote by λ_{v_i} the total network resource available at POI $v_i \in \mathcal{V}$. On the other hand, computing resources are provided by MEC hosts which are reachable from the access network, as shown in Figure 1. We assume that a POI can be served by any MEC host. Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of MEC hosts, we represent by ψ_m the total computing capacity available at MEC host $m \in \mathcal{M}$.

Whenever possible, service applications should run on the MEC hosts since this results in shorter response times and improved touristic experience. When no resource is available at the edge, services run at the remote cloud with degraded performance, in terms of communication delay.

3.1.3 Tourists

We consider a set $\mathcal{U} = \{1, \ldots, U\}$ of tourists, where each tourist $u \in \mathcal{U}$ is described by the following attributes. The tourist interest in each POI category $c \in \mathcal{C}$, denoted by $Int_u(c) \in \mathbb{R}$. The preferred day and starting and ending locations for the tour, denoted respectively by D_u, v_s^u, and v_f^u, with $v_s^u, v_f^u \in \mathcal{V}$. Finally, the time budget (i.e., maximum available time for visits) to complete the tour, represented by $B_u \in \mathbb{R}$. The tourist is also characterized by the set $\mathcal{A}_u \in 2^{\mathcal{A}}$, representing the applications she is willing to use during the tour, where $2^{\mathcal{A}}$ is the power set of $\mathcal{A}$. The set $\mathcal{A}_u$ determines the minimum and maximum demand in terms of network and computing resources required to run the tourist applications during the tour. We represent this demand as:

- $\lambda_u^{min} = \sum_{a \in \mathcal{A}_u} \lambda_a^{min}$: the minimum network demand, in bps, required by tourist u during a tour;
- $\lambda_u^{max} = \sum_{a \in \mathcal{A}_u} \lambda_a^{max}$: the maximum network demand, in bps, required by tourist u during a tour;
- $\psi_u^{min} = \sum_{a \in \mathcal{A}_u} \psi_a^{min}$: the minimum computing demand, in reference core (RC), required by tourist u during a tour;
- $\psi_u^{max} = \sum_{a \in \mathcal{A}_u} \psi_a^{max}$: the maximum computing demand, in reference core (RC), required by tourist u during a tour.

Next, we describe how we define visiting-related information such as the popularity of the POI ($Pop(v_i)$), expected POI visiting time ($Dur(v_i)$), and the interest of a tourist in a certain category ($Int_u(c)$).

3.2 Visiting-related information

Similar to other works [6], [10], [11], in this paper, we define $Pop(v_i)$, $Dur(v_i)$, and $Int_u(c)$ based on users' past travel histories extracted from location-based social media such as Facebook, Flickr, and Foursquare. Given a tourist $u \in \mathcal{U}$ who has visited n POIs in a city, we define her travel history as an ordered sequence $S_u = ((v_1, t_{v_1}^a, t_{v_1}^d), \ldots, (v_n, t_{v_n}^a, t_{v_n}^d))$, where each triple $(v_i, t_{v_i}^a, t_{v_i}^d)$ represents a visit at POI v_i, the arrival time $t_{v_i}^a$ at POI v_i and the departure time $t_{v_i}^d$ from POI v_i. The user's visit duration at POI v_i can be computed as the difference between $t_{v_i}^d$ and $t_{v_i}^a$.

A travel history S_u can be further divided into multiple travel sub-sequences. We divide a travel history into distinct travel sub-sequences if visits to consecutive POIs occur more than t_{split} time units apart, i.e., if $t_{v_{i+1}}^a - t_{v_i}^d > t_{split}$. We denote by S_u^1, S_u^2, …, S_u^k, a travel history S_u with k travel sub-sequences.

The popularity of a POI $v_i \in \mathcal{V}$, $Pop(v_i)$, is defined based on the number of times v_i has been visited. Formally, given the set $\mathcal{S}$ of all travel histories, i.e., $\mathcal{S} = \bigcup_{u \in \mathcal{U}} S_u$, the popularity of a POI v_i is defined as:

$$Pop(v_i) = \sum_{S_u \in \mathcal{S}} \sum_{v_x \in S_u} \delta(v_x, v_i), \ \ \forall v_i \in \mathcal{V}, \tag{1}$$

$$\text{where } \delta(v_x, v_i) = \begin{cases} 1, & \text{if } v_x = v_i \\ 0, & \text{otherwise} \end{cases}$$

Similarly, the expected visiting time of a POI $v_i \in \mathcal{V}$, $Dur(v_i)$, is defined as follows:

$$Dur(v_i) = \frac{\sum_{S_u \in \mathcal{S}} \sum_{v_x \in S_u} (t_{v_x}^d - t_{v_x}^a) \delta(v_x, v_i)}{Pop(v_i)}, \ \ \forall v_i \in \mathcal{V}. \tag{2}$$

While the popularity of a POI is the same for all tourists, the tourist interest in a certain POI depends on the category of the POI and it is different for different tourists. The

[1] unit of measure that represents the processing capacity of a reference CPU core.

interest level of a tourist $u \in \mathcal{U}$ in a category $c \in \mathcal{C}$, is defined as:

$$Int_u(c) = \sum_{v_x \in S_u} \frac{(t^d_{v_x} - t^a_{v_x})}{Dur(v_x)} \gamma(c, Cat(v_x)), \tag{3}$$

$$\text{where } \gamma(c, Cat(v_x)) = \begin{cases} 1, & \text{if } c = Cat(v_x) \\ 0, & \text{otherwise} \end{cases}$$

$Int_u(c)$ is based on the time tourist u spent at a POI of category c relative to the expected visiting time at that POI. The intuition here is to determine the interest level of the tourist in a certain category by computing the time she spent at POIs of that category compared to other tourists.

TABLE 1
Notations and definitions used in the problem formulation.

Symbol	Description
$\mathcal{V}$	set of POIs (in a city)
$\mathcal{E}$	set of edges connecting vertices (i.e., POIs)
$\mathcal{C}$	set of all POI categories
$\mathcal{U}$	set of tourists
$\mathcal{M}$	set of MEC hosts
$\mathcal{A}$	set of applications offered by the MTC
v_i	a POI in $\mathcal{V}$
$c_{i,j}$	travel time between POI v_i to POI v_j
$Pop(v_i)$	popularity of POI v_i
$Dur(v_i)$	expected visiting time of POI v_i
$Cat(v_i)$	category of POI v_i
$Lat(v_i)$	latitude of POI v_i
$Long(v_i)$	longitude of POI v_i
λ_{v_i}	total network resource available at POI v_i
ψ_m	total computing resource available at MEC m
λ_a^{min}	minimum network demand, in bps, of application $a \in \mathcal{A}$
λ_a^{max}	maximum network demand, in bps, of application $a \in \mathcal{A}$
ψ_a^{min}	minimum computing demand, in reference cores, of application $a \in \mathcal{A}$
ψ_a^{max}	maximum computing demand, in reference cores, of application $a \in \mathcal{A}$
u	a tourist in $\mathcal{U}$
B_u	time budget of tourist u to complete the tour
$\mathcal{A}_u$	set of applications tourist u is willing to use
$Int_u(c)$	interest of tourist u in the POI category c
D_u	date when the tour of tourist u will take place
v_s^u	place where the itinerary needs to begin
v_f^u	place where the itinerary needs to finish
λ_u^{min}	minimum network demand required by tourist u during a tour
λ_u^{max}	maximum network demand required by tourist u during a tour
ψ_u^{min}	minimum computing demand required by tourist u during a tour
ψ_u^{max}	maximum computing demand required by tourist u during a tour
$\mathcal{S}$	set of all travel histories
S_u	travel history of tourist u
S_u^k	the k-th travel sequence of tourist u
t_{split}	threshold to split a travel history into travel sequences
$\mathcal{U}_\mathcal{D}$	set of tourists whose tours are scheduled to day D
τ	index of the time slot in which MEC infrastructure resources are allocated

4 PROBLEM FORMULATION

In this section, we define the problem of recommending personalized tour itineraries in the context of next-generation touristic services. We call it Multi-access Edge Computing Aware Personalized Tour Recommendation (MEC-PTR) problem. Hereafter, we first revisit an existing formulation of the traditional Personalized Tour Recommendation (PersTour) problem, consistent with the one introduced in [6]. The PersTour problem finds a single optimal itinerary for a tourist given her preferences and constraints. Since only one itinerary is considered in the solution of the PersTour problem, there is no opportunity for optimizing the MEC resource allocation and to improve the virtual touristic experience. In addition, PersTour recommends personalized tour itineraries for an individual tourist. However, since the MEC resources are shared among the tourists during their tours, the recommendation problem must take into account a multi-user (tourist) perspective. Thus, we propose a new formulation, the MEC-PTR problem, to capture the joint problem of recommending personalized tour itineraries for multiple individual tourists while efficiently allocating MEC resources to enable advanced touristic service applications.

4.1 PersTour problem

The problem of recommending a personalized tour itinerary for a tourist interested in visiting multiple POIs (in a one-day trip), denoted PersTour, is a route-planning problem. Specifically, in this problem we are given a complete non-oriented graph $G = (\mathcal{V}, \mathcal{E})$ where nodes represent the POIs of the city and edges are the links connecting the nodes. As described in Section 3, each POI $v_i \in \mathcal{V}$ is characterized by a vector ($Cat(v_i)$, $Pop(v_i)$, $Dur(v_i)$). Each edge $(v_i, v_j) \in \mathcal{E}$ is associated with a cost $c_{i,j}$ representing the travel time between POI v_i to POI v_j. Similarly, a tourist $u \in \mathcal{U}$ is characterized by a vector (($Int_u(c) \forall c \in \mathcal{C}$), v_s^u, v_f^u, B_u).

The vast majority of the literature related to the PersTour problem defines it in the context of the OP and its integer programming problem formulation [6], [8], [9]. Formally, let $I_u = (v_1, \ldots, v_N)$, with $v_i \in \mathcal{V}, i = 1, \ldots, N$, be an itinerary for tourist $u \in \mathcal{U}$, i.e., a sequence of POI visits starting at $v_1 = v_s^u$ and ending at $v_N = v_f^u$. Each POI $v_i \in I_u$ has a profit $\alpha Int_u(Cat(v_i)) + (1 - \alpha) Pop(v_i)$ as perceived by tourist u. This profit depends on the value of $\alpha \in [0, 1]$, which can be chosen to give more emphasis either on the tourist interest or on the POI popularity. Thus, for each itinerary I_u, a score (profit) can be given by:

$$Prof(I_u) = \sum_{i=2}^{N-1} \sum_{j=2}^{N} x_{i,j} (\alpha Int_u(Cat(v_i)) + (1 - \alpha) Pop(v_i)), \tag{4}$$

where $x_{i,j} = 1$ if POIs v_i and v_j are visited in sequence, and $x_{i,j} = 0$ otherwise. On the other hand, let $Dur(v_j) Int_u(Cat(v_j))$ be the personalized recommended time for tourist u to spend at POI v_j. This time is based on the tourist interest level in category $Cat(v_j)$ and the expected visiting time at POI v_j. Since traveling from POI v_i to POI v_j and visiting v_j consumes the tourist budget, the following cost function can be defined:

$$Cost_u(i, j) = c_{i,j} + Dur(v_j) Int_u(Cat(v_j)). \tag{5}$$

The objective of the PersTour problem under the OP is to find an itinerary I_u that:

$$\text{maximize } (Prof(I_u)) \tag{6}$$

subject to:

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N} Cost_u(i,j)x_{i,j} \leq B_u \tag{7}$$

$$\sum_{j=2}^{N} x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1 \tag{8}$$

$$\sum_{i=1}^{N-1} x_{i,k} = \sum_{j=2}^{N} x_{k,j} \leq 1, \forall k = 2, ..., N-1 \tag{9}$$

$$2 \leq v_i \leq N, \forall i = 2, ..., N \tag{10}$$

$$v_i - v_j + 1 \leq (N-1)(1 - x_{i,j}), \forall i, j = 2, ..., N. \tag{11}$$

Equation (6) aims to maximize the total collected profit of tourist u without exceeding her budget (constraint represented by Equation (7)). The other constraints ensure the proper construction of the itinerary: Equation (8) ensures that the path starts at POI $v_1 = v_s^u$ and ends at POI $v_N = v_f^u$; Equation (9) ensures that the path is connected and no POI is visited more than once; Equations (10) and (11) ensure that there are no subtours.

4.2 MEC-PTR problem

The MEC-PTR problem adds the MEC resource availability and the demand of the service applications chosen by the tourists (defined in Section 3) to the PersTour problem. In addition, different from PersTour, the MEC-PTR problem seeks to recommend personalized tour itineraries for multiple individual tourists whose trips are scheduled for the same day. Specifically, let $\mathcal{U_D} \subseteq \mathcal{U}$ be the set of tourists whose tours are scheduled for day D, i.e., $\mathcal{U_D} = \{u \in \mathcal{U} \mid D_u = D\}$. Let $I_u = (v_1, \ldots, v_N)$, with $v_i \in \mathcal{V}, i = 1, \ldots, N$ and $v_1 = v_s^u$ and $v_N = v_f^u$, be an itinerary for tourist $u \in \mathcal{U_D}$. Let Equations (4) and (5) represent the profit associated with I_u and the cost function associated with u, respectively. The objective of the MEC-PTR problem is to find one itinerary for every tourist $u \in \mathcal{U_D}$, in which the set of chosen itineraries, denoted by $\mathcal{I}_*$, satisfies the requirements **R1** – **R4**.

As mentioned before, the output of the PersTour problem is only one optimal itinerary for a tourist $u \in \mathcal{U}$. This is so because the OP is a single-criterion variant of the Traveling Salesman Problem with Profits (TSPP or TSPwP) [8], [9]. However, the TSPP/TSPwP is a bi-criteria generalization of the TSP with two conflicting objectives: maximizing the total collected profit and minimizing the travel cost. Thus, by modeling the PersTour problem as a TSPwP, one can obtain the set $\mathcal{I}_{u*} = \{I_{u*}^1, \ldots, I_{u*}^k\}$ of non-dominated itinerary solutions (i.e., Pareto front) for a tourist $u \in \mathcal{U}$. Using this idea, we define the MEC-PTR as a *two-stage optimization problem.* The **first stage** finds the set $\mathcal{I}_{u*}$ for a user $u \in \mathcal{U_D}$ and is defined as:

Stage 1 (MEC-PTR):

$$\text{maximize } (Prof(I_u)) \tag{12}$$

$$\text{minimize } \left(\sum_{i=1}^{N-1}\sum_{j=2}^{N} Cost_u(i,j)x_{i,j}\right) \tag{13}$$

subject to:

Equations $(7) - (11)$.

Since the TSPwP belongs to the class of NP-hard problems [22], we present in Section 5 an efficient algorithm to generate the Pareto front ($\mathcal{I}_{u*}$) using a dynamic programming approach. Let denote by $\mathcal{I} = \bigcup_{u \in \mathcal{U_D}} \mathcal{I}_{u*}$ the set found by solving stage 1 of the MEC-PTR problem for all tourist $u \in \mathcal{U_D}$. We call $\mathcal{I}$ the set of *candidate itineraries*. Note that because of the constraint represented by Equation (7), all itineraries in $\mathcal{I}$ satisfy the budget constraint of the associated tourist. Since each itinerary $I_{u*}^j \in \mathcal{I}_{u*}$ has a profit $Prof(I_{u*}^j)$, we can formulate the **second stage** of MEC-PTR as *a Mixed Integer Linear Programming (MILP) problem*, as described below.

Assume that the indicator function $\phi(I_{u*}^j, v_i) \in \{0, 1\}$ represents POI visiting in an itinerary, with $\phi(I_{u*}^j, v_i) = 1$ if itinerary I_{u*}^j visits POI v_i and $\phi(I_{u*}^j, v_i) = 0$ otherwise. We assume that every itinerary in $\mathcal{I}$ starts and ends withing a period of time T, and that MEC infrastructure resources (computing and network) for these itineraries are allocated in discrete time slots of size Δt. The time slots are indexed by $\tau \in \mathbb{Z}$ such that $1 \leq \tau \leq T$.

Given an itinerary $I_{u*}^j \in \mathcal{I}$, a POI $v_i \in \mathcal{V}$, and a time index τ, we define the indicator function $\rho(I_{u*}^j, v_i, \tau) \in \{0, 1\}$, with $\rho(I_{u*}^j, v_i, \tau) = 1$ if itinerary I_{u*}^j is visiting POI v_i during τ, and $\rho(I_{u*}^j, v_i, \tau) = 0$ otherwise. This information can be derived since we know the start time, sequence of visits, and visit duration for each itinerary $I_{u*}^j \in \mathcal{I}_{u*}$ obtained from the first stage.

Let the set of decision variables $y(I_{u*}^j) \in \{0, 1\}$ represent itinerary choices, so that $y(I_{u*}^j) = 1$, if itinerary $I_{u*}^j \in \mathcal{I}$ composes the solution; and $y(I_{u*}^j) = 0$ otherwise. For each POI visit, a MEC host has to provide computing resources for the tourist. We define the decision variable $z(I_{u*}^j, v_i, m) \in \{0, 1\}$ to represent MEC host association, so that $z(I_{u*}^j, v_i, m) = 1$, if MEC host $m \in \mathcal{M}$ is responsible to provide computing resources during a visit to POI $v_i \in \mathcal{V}$ in itinerary $I_{u*}^j \in \mathcal{I}$; and $z(I_{u*}^j, v_i, m) = 0$ otherwise.

Let the decision variables $p(I_{u*}^j, v_i) \in \mathbb{R}$ and $q(I_{u*}^j, v_i, m) \in \mathbb{R}$ represent, respectively, the amount of network and the amount of computing resources (at MEC host $m \in \mathcal{M}$) allocated during a visit to POI $v_i \in \mathcal{V}$ in itinerary $I_{u*}^j \in \mathcal{I}$. Assuming $Norm(value)$ as a generic function that normalizes a value, we define the **second stage** of the MEC-PTR problem as:

Stage 2 (MEC-PTR):

$$\text{maximize} \sum_{I^j_{u*} \in \mathcal{I}} y(I^j_{u*}) Norm(Prof(I^j_{u*})) + \sum_{I^j_{u*} \in \mathcal{I}} \sum_{v_i \in \mathcal{V}} \frac{Norm(p(I^j_{u*}, v_i))}{2|I^j_{u*}|} + \sum_{I^j_{u*} \in \mathcal{I}} \sum_{v_i \in \mathcal{V}} \sum_{m \in \mathcal{M}} \frac{Norm(q(I^j_{u*}, v_i, m))}{2|I^j_{u*}|} \quad (14)$$

subject to:

$$\sum_{I^j_{u*} \in \mathcal{I}_{u*}} y(I^j_{u*}) = 1, \quad \forall \mathcal{I}_{u*} \in \mathcal{I} \quad (15)$$

$$\sum_{m \in \mathcal{M}} z(I^j_{u*}, v_i, m) = y(I^j_{u*}) \phi(I^j_{u*}, v_i), \quad (16)$$
$$\forall I^j_{u*} \in \mathcal{I}, \quad \forall v_i \in \mathcal{V}$$

$$p(I^j_{u*}, v_i) \geq \lambda^{min}_u y(I^j_{u*}) \phi(I^j_{u*}, v_i) \quad (17)$$

$$p(I^j_{u*}, v_i) \leq \lambda^{max}_u y(I^j_{u*}) \phi(I^j_{u*}, v_i), \quad (18)$$
$$\forall I^j_{u*} \in \mathcal{I}, \quad \forall v_i \in \mathcal{V}$$

$$q(I^j_{u*}, v_i, m) \geq \psi^{min}_u z(I^j_{u*}, v_i, m) \phi(I^j_{u*}, v_i) \quad (19)$$

$$q(I^j_{u*}, v_i, m) \leq \psi^{max}_u z(I^j_{u*}, v_i, m) \phi(I^j_{u*}, v_i), \quad (20)$$
$$\forall I^j_{u*} \in \mathcal{I}, \quad \forall v_i \in \mathcal{V}, \quad \forall m \in \mathcal{M}$$

$$\sum_{I^j_{u*} \in \mathcal{I}} \rho(I^j_{u*}, v_i, \tau) p(I^j_{u*}, v_i) \leq \lambda_{v_i}, \quad (21)$$
$$\forall v_i \in \mathcal{V}, \quad 1 \leq \tau \leq T$$

$$\sum_{I^j_{u*} \in \mathcal{I}} \sum_{v_i \in \mathcal{V}} \rho(I^j_{u*}, v_i, \tau) q(I^j_{u*}, v_i, m) \leq \psi_m, \quad (22)$$
$$\forall m \in \mathcal{M}, \quad 1 \leq \tau \leq T$$

Equation (14) aims to maximize simultaneously three objectives: (i) the aggregated total collected profit of all tourists in $\mathcal{U_D}$ given that each tourist $u \in \mathcal{U_D}$ must have only a single itinerary $I^j_{u*} \in \mathcal{I}_{u*}$; (ii) the average amount of allocated network resources; and (iii) the average amount of allocated computing resources at the edge. Together, these objectives ensure requirement **R4**. The $Norm(value)$ function assures that all objectives stay in the same interval of values. The aggregated profit of the itineraries and the overall average resource allocation have equal weights. Thus, Equation (14) may choose itineraries that do not have the highest individual profit, but provides the best balance with the resource allocation and, as a consequence, an improved experience to the set of tourists as a whole.

Although the constraint represented by Equation (7) ensures that every itinerary $I^j_{u*} \in \mathcal{I}_{u*}$ satisfies the time constraint of tourist u, in order to meet requirement **R1**, we still have to ensure, for each tourist $u \in \mathcal{U_D}$, that exactly one itinerary $I^j_{u*} \in \mathcal{I}_{u*}$ is chosen. This is accomplished by the constraint represented by Equation (15).

If an itinerary I^j_{u*} is chosen to compose a solution, we also need to associate one MEC host with each POI visit along this itinerary to provide computing resources. We achieve this with the constraint represented by Equation (16).

For each tourist $u \in \mathcal{U_D}$ and chosen itinerary I^j_{u*}, we also need to select at least the minimum amount of resources required for the set $\mathcal{A}_u \in 2^{\mathcal{A}}$ of chosen applications, so that the applications can execute properly. We have to take care of not allocating more than it is needed (requirement **R2**). We ensure proper allocation of network resources with Equations (17) and (18). Similarly, Equations (19) and (20) ensure proper allocation of computing resources.

Finally, we have to make sure that the performed allocations do not exceed the amount of available network and computing resources, at any given time slot (requirement **R3**). This is achieved with the constraints represented by Equations (21) and (22), respectively.

5 An efficient algorithm for MEC-PTR

5.1 Description of the algorithm

To solve the MEC-PTR problem, we propose the Multi-access Edge Computing Aware Personalized Itinerary Recommendation (MEC-PIR) algorithm described in Algorithm 1. As illustrated in the latter, in the initialization step (line 1) the set $\mathcal{I}$ of candidate itineraries is set to empty. Then, for each tourist $u \in \mathcal{U_D}$, we solve the first stage of the MEC-PTR problem by calling Algorithm 2 (line 3). The output of Algorithm 2 is the Pareto front $\mathcal{I}_{u*}$ for tourist u, i.e., the set of optimal solutions for the multi-objective optimization problem. This set is then added to the set of candidate itineraries (line 4). After the Pareto front has been computed for all tourist $u \in \mathcal{U_D}$, we use the set $\mathcal{I}$ of candidate itineraries as input to solve the second stage of the MEC-PTR problem (line 5). The result of the second stage is the set $\mathcal{I}_*$ of itineraries satisfying requirements **R1** – **R4** (line 6).

As described in Section 4.2, the first stage of the MEC-PTR problem is based on the TSPwP, which belongs to the class of NP-hard problems. To solve this part of the problem optimally and efficiently, MEC-PIR uses a variant of the shortest path problem with resource constraints (SPPRC) [23], summarized in Algorithm 2. To solve the second stage of the MEC-PTR problem, MEC-PIR uses a traditional optimization tool (i.e., solver). In the following, we describe in more detail how we use the variant of the SPPRC to solve the first stage of the MEC-PTR problem.

Algorithm 1: The MEC-PIR Algorithm

```
Input  : $\mathcal{U_D}$; $G = (\mathcal{V}, \mathcal{E})$; $\alpha$;
         $(Pop(v_i), Dur(v_i), Cat(v_i))\ \forall v_i \in \mathcal{V}$;
         $((Int_u(c)\ \forall c \in \mathcal{C}), v^u_s, v^u_f, B_u)\ \forall u \in \mathcal{U_D}$
Output: $\mathcal{I}_*$ satisfying R1-R4

1 $\mathcal{I} \leftarrow \emptyset$
2 forall $u \in \mathcal{U_D}$ do
      // Stage 1
3     $\mathcal{I}_{u*} \leftarrow$ Algorithm 2 $(G = (\mathcal{V}, \mathcal{E})$; $\alpha$;
      $(Pop(v_i), Dur(v_i), Cat(v_i))\ \forall v_i \in \mathcal{V}$;
      $((Int_u(c)\ \forall c \in \mathcal{C}), v^u_s, v^u_f, B_u))$
4     $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_{u*}$
  // Stage 2
5 Find $\mathcal{I}_*$ by solving Stage 2 (MEC-PTR) using $\mathcal{I}$ as input
6 return $\mathcal{I}_*$
```

5.2 Solution for the first stage

The SPPRC finds a shortest path among all paths that start from a source node, end at a destination node, and satisfy a set of constraints defined over a set of resources. A resource, in this context, corresponds to a quantity, for example, time or money, which varies along a path according to resource extension functions. A resource extension function is defined for every edge in the graph and every resource involved in the problem. Each function provides a lower bound on its corresponding resource related to the next vertex, given the value accumulated in the present vertex. The resource constraints are given as intervals, also known as resource windows, which limit the values that can be taken by the resources at every vertex along a path. These constraints are defined for every vertex and every considered resource. Thus, if multiple resources are involved in the problem, the SPPRC becomes very close to a multi-criteria problem, since the paths may not be comparable for different resources. In this context, by solving the SPPRC, we obtain a set of optimal solutions, i.e., a Pareto front.

Indeed, due to the constraint represented by Equation (9), to solve the first stage of the MEC-PTR, MEC-PIR employs the Elementary SPPRC (ESPPRC), a variant of the SPPRC algorithm that finds only elementary paths, i.e., paths in which no vertex is visited more than once. Additionally, we use two types of resource: a constrained resource (i.e., time) and an unconstrained one (i.e., score or profit). In the following, we describe how we map the first stage of MEC-PTR into the ESPPRC framework. We use a dynamic programming methodology.

ESPPRC procedure. For a given itinerary or path $P = (v_0, v_1, ..., v_{p-1}, v_p)$, let $Res(P) = v_p$ be the resident vertex of P (i.e., the last vertex of P) and $(v_0, v_1, ..., v_{p-1})$ be an example of prefix path of P. For the sake of efficiency, itinerary (or paths) in dynamic programming algorithms are represented through labels. Thus, associated with every itinerary P there is a label R. A label stores multiple information, for example, its resident vertex, its predecessor edge, its predecessor label, and its current vector of resource values. For the first stage of MEC-PTR, two resources are critical to be stored in the labels:

- $Spent(R) \in \mathbb{R}$: constrained resource that represents the amount of time spent along the itinerary P. Time is consumed whenever the tourist visits a POI or move from one POI to another. As described in Section 4.2, for every tourist $u \in \mathcal{U_D}$, the time consumed moving from POI i to POI j and visiting j is represented by $Cost_u(i, j)$.
- $Profit(R) \in \mathbb{R}$: unconstrained resource that represents the profit (or score) of the itinerary P. As described in Section 4.2, for every tourist $u \in \mathcal{U_D}$, each POI $v_i \in P$ has a profit equals to $\alpha Int_u(Cat(v_i)) + (1-\alpha)Pop(v_i)$.

In Algorithm 2, we call a function `getLabel` to obtain a label from a path, and a function `getPath` to obtain a path from a label. Algorithm 2 operates over two main sets: $\mathcal{P}$, the set of useful paths, and $\mathcal{Q}$, the set of unprocessed paths, i.e., paths that have not yet been extended. The useful paths $P \in \mathcal{P}$ have already been processed and they are Pareto-optimal paths or prefixes of Pareto-optimal paths. The set $\mathcal{P}$ is initially empty (line 1), while the set $\mathcal{Q}$ is initiated with the trivial path (v_s^u) (line 2). In summary, the main loop (line 3) consists of: **1)** selecting and removing a path $Q \in \mathcal{Q}$ (line 4), **2)** making all feasible extensions from Q (lines 5-10), **3)** adding Q to $\mathcal{P}$ (line 11), and **4)** identifying and removing dominated paths from $\mathcal{P} \cup \mathcal{Q}$ (lines 12-17). In the end, the algorithm obtains the set of non-dominated paths (or itineraries) that end in vertex v_f^u (line 18), discarding eventual prefixes contained in P.

Algorithm 2: Dynamic programming ESPPRC – Stage 1 (MEC-PTR)

```
Input : $G = (\mathcal{V}, \mathcal{E})$; $\alpha$;
        $(Pop(v_i), Dur(v_i), Cat(v_i))\ \forall v_i \in \mathcal{V}$;
        $((Int_u(c)\ \forall c \in \mathcal{C}), v_s^u, v_f^u, B_u)$
Output: set $\mathcal{I}_{u*} = \{I_{u*}^1, \dots, I_{u*}^k\}$ of non-dominated
        itinerary (or path) solutions for a tourist
        $u \in \mathcal{U_D}$

1  $\mathcal{P} \leftarrow \emptyset$
2  $\mathcal{Q} \leftarrow \{(v_s^u)\}$
3  while $\mathcal{Q} \neq \emptyset$ do
4      Choose a path $Q \in \mathcal{Q}$ and remove $Q$ from $\mathcal{Q}$
5      forall $v_j \in \mathcal{V} \mid (v_j \neq Res(Q)$ and $v_j \notin Q)$ do
6          $R \leftarrow$ getLabel($Q$)
7          $feasible, R' \leftarrow$ Algorithm 3
             $(R, v_j, \alpha, Pop(v_j), Dur(v_j), Int_u(Cat(v_j)), B_u)$
8          if $feasible = true$ then
9              $Q' \leftarrow$ getPath($R'$)
10             Add the path $Q'$ to $\mathcal{Q}$
11     Add the path $Q$ to $\mathcal{P}$
12     forall pair of paths $(P_1, P_2) \in \mathcal{P} \cup \mathcal{Q}$ and
         $(Res(P_1) = Res(P_2))$ do
13         $R_1 \leftarrow$ getLabel($P_1$)
14         $R_2 \leftarrow$ getLabel($P_2$)
15         dominates $\leftarrow$ Algorithm 4 $(R_1, R_2)$
16         if dominates $= true$ then
17             Remove the path $P_2$
18 Filter $\mathcal{P}$ such that $Res(P) = v_f^u\ \forall P \in \mathcal{P}$
19 return $\mathcal{P}$
```

Let Ξ be the set of all solutions of the ESPPRC, where each element $\mathcal{X} \in \Xi$ is a set of Pareto-optimal paths, such that:

$$\exists \mathcal{X} \in \Xi \mid \mathcal{X} \subseteq \{(Q, P) \mid Q \in \mathcal{Q}, P \in \mathcal{Z}(Q)\} \cup \mathcal{P}, \quad (23)$$

where $\mathcal{Z}(Q) = \{P \mid (Q, P) \in \mathcal{F}(v_i, v_j)) \cap \mathcal{G}\}$ represents the set of all feasible extensions. The set $\mathcal{F}(v_i, v_j)$ has all resource-feasible paths from vertex v_i to vertex v_j.

The condition described by Equation (23) is held for the initialization of Algorithm 2 (lines 1-2), since $v_i = v_s^u$ and $v_j = Res(P) = \emptyset$. For every vertex $v_j \in \mathcal{V}$ (line 5), the condition described by Equation (23) is verified (line 8) before adding the extended path Q' to $\mathcal{Q}$ (line 10). This depends on Algorithm 3 (called in line 7) described latter in this section. In addition, to be efficient, Algorithm 2 must avoid making extensions over dominated prefix paths, i.e., paths that are not part of the Pareto front. Thus, after processing every path $Q \in \mathcal{Q}$ (lines 4-11), the algorithm takes every pair of paths $(P_1, P_2) \in \mathcal{P} \cup \mathcal{Q}$ where $Res(P_1) = Res(P_2)$ (line 12), verifies if P_1 dominates P_2 (line 16) and, if this is true, removes the path P_2 (line 17). This procedure depends on Algorithm 4 (called in line 15) described latter in this section.

Extension procedure. The goal of Algorithm 3 is to verify if path P (described by label R) can be extended to vertex v_j. After identifying the path P associated with label R (line 1) and the resident vertex $v_i = Res(P)$ of this path (line 2), the algorithm includes into label R' (associated with the extended path P') the information about the time consumed for travelling from v_i to v_j and for visiting the vertex (or POI) v_j (lines 3-4). The algorithm also includes into label R' the information about the profit of the path P' (line 5). Finally, the algorithm verifies if the constrained resource $Spend(R')$ is inside its resource window (line 6), which can be seen as the interval $[0, B_u - Cost_u(i,j)]$. The algorithm returns **true**, if the extension is feasible (line 7), or **false** (line 9), otherwise.

```
Algorithm 3: Extension
   Input  : R (label), $v_j$ (vertex for path extension),
            $\alpha$, $Pop(v_j)$, $Dur(v_j)$, $Int_u(Cat(v_j))$, $B_u$
   Output: true or false to indicate the feasibility or not of
            the extension, R′ (label of the path extension)
1  $P \leftarrow$ getPath($R$)
2  $v_i \leftarrow Res(P)$
3  $Cost_u(i,j) \leftarrow c_{i,j} + Dur(v_j)Int_u(Cat(v_j))$
4  $Spent(R') \leftarrow Spent(R) + Cost_u(i,j)$
5  $Profit(R') \leftarrow$
       $Profit(R) + (\alpha Int_u(Cat(v_j)) + (1-\alpha)Pop(v_j))$
6  if $Spent(R') \leq B_u$ then
7      return (true,R′)
8  else
9      return (false,R′)
```

Dominance procedure. In general terms, dominance rules identify non-useful paths by comparing the resource vector $T(Q)$ and the set of all feasible extensions $\mathcal{Z}(Q)$ with the corresponding $T(P)$ and $\mathcal{Z}(P)$, such that $Res(P) = Res(Q)$. In the context of the first stage of MEC-PTR, as described by Algorithm 4, this comparison consists in verifying if label R_1 (corresponding to path P_1) provides a higher or equal profit and consumes less or equal time than label R_2 (that corresponds to path P_2) (line 1). The algorithm returns **true**, if the condition is satisfied (line 2), or **false** (line 4), otherwise. When the condition is true, it means that path P_2 (represented by label R_2) must be removed, since it was identified as equal or worse than P_1 (represented by label R_1).

```
Algorithm 4: Dominance
   Input  : $\mathbf{R_1}$ (label), $\mathbf{R_2}$ (label)
   Output: true if $\mathbf{R_1}$ dominates $\mathbf{R_2}$, false otherwise
1  if $(Profit(R_1) \geq Profit(R_2))$ and
      $(Spent(R_1) \leq Spent(R_2))$ then
2      return true
3  else
4      return false
```

6 Experimental evaluation

This section evaluates the performance of MEC-PIR using real-world travel histories. In the following, we first describe how we obtain these travel histories and then present an analysis of the visiting-related statistics found in our dataset (Section 6.1). Later, we outline the baseline algorithm (Section 6.2) used for comparison. Finally, we detail the experimental methodology (Section 6.3) and discuss the obtained results (Section 6.4).

6.1 Mining tourist travel histories from real data

In order to achieve a realistic evaluation, we use real data to derive POI information, tourist travel histories, and visiting-related statistics. More specifically, we use the Google Places API[2] to obtain a list of POIs from four cities in Europe, namely: London, Madrid, Barcelona, and Athens (one list per city). These cities are important from a touristic point of view and guarantee variety and diversity in our evaluation. For each city, we select the first 30 POIs with the highest rating, i.e., we use $|\mathcal{V}| = 30$. In each list, each POI is identified by an id, name, latitude and longitude $(Lat(v_i), Long(v_i))$, and the category $(Cat(v_i))$ it belongs to. The cost $c_{i,j}$ of an edge connecting two POIs v_i and v_j in the same list is computed using the Google Matrix Distance API[3] in the walking mode.

Following the approach proposed in [6], [10], [11], we use the Flickr API[4] to extract geotagged photos and derive user travel histories. We collect photos taken from 01-01-2017 to 12-31-2018 for the four considered cities. Each photo is tagged with the tourist (user) id of the photo owner, timestamp, and latitude/longitude geo-coordinates.

For each city, we combine its photo dataset with its POI list to generate the set $\mathcal{S}$ of travel histories in the city. First, we match each photo to the corresponding POI using the geo-coordinates of the photo and of the POIs. A photo is mapped to a POI if their geo-coordinates differ by less than 100 meters according to the Haversine formula. If this condition holds for more than one POI, the photo is associated with the nearest POI, so that each photo is mapped to a single POI. Then, for each user, we construct the travel history S_u sorting the photos by user id and timestamp. After that, for each user, we group consecutive POI visits as an individual travel sequence if the consecutive POI visits differ by at most 8 hours, i.e, we use $t_{split} = 8$. For each travel sequence $S_u^k \subseteq S_u$ and for each POI $v_i \in S_u^k$, we take the time of the first and the last photo taken by user u at v_i as the arrival $(t_{v_i}^a)$ and departure $(t_{v_i}^d)$ time in that POI, respectively. Finally, we discard travel sequences with cycles (repeated POI visits) or with less than 3 POI visits.

Table 2 summarizes the number of photos, number of users, number of travel sequences, and number of travel sequences with at least 3 POI visits (valid sequences) found in our dataset. Indeed, we do not observe travel sequences with cycles. However, the number of sequences with 3 or more POIs is small. Fig. 2 shows the cumulative distribution function (CDF) of the length of the travel sequences found

[2] https://developers.google.com/places/web-service/intro

[3] https://developers.google.com/maps/documentation/distance-matrix/start

[4] https://www.flickr.com/services/developer/api/

in our dataset for each city. Sequences with 3 or more POIs represent less than 5% of the sequences found in each city. The longest sequence in Athens has 5 POIs, while in Barcelona, London, and Madrid the longest sequence has 6, 9, and 8 POIs, respectively.

TABLE 2
Dataset summary.

City	No. of Photos	No. of Users	No. of Sequences	No. of Valid Sequences
Athens	73,548	1,576	516	17
Barcelona	237,858	4,409	1,654	47
London	637,504	10,958	5,018	120
Madrid	180,435	3,533	1,396	44

Based on the obtained set of travel histories ($\mathcal{S}$) of each city, we can compute the visiting-related statistics: popularity of the POI ($Pop(v_i)$), expected POI visiting time ($Dur(v_i)$), and tourist's interests in a given category ($Int_u(c)$) using the Equations (1), (2) and (3), respectively. In the following, we present an analysis of the visiting-related statistics found in our database. These statistics were computed after removing all sequences with less than 3 POIs.

Fig. 3 shows the distribution of the popularity of the POIs per city. Recall that the popularity of a POI $v_i \in \mathcal{V}$ is calculated using Equation (1). In each graphic of the figure, we present the histogram (y-axis on the left) and the CDF (y-axis on the right) of the popularity of the POIs. The x-axis presents the POIs. For comparison purpose, in each city, the popularity is displayed as a percentage of the total number of visits (i.e., relative values). The top 10 POIs (in terms of popularity) represents 89%, 83%, 79% and 80% of all POI visits in Athens, Barcelona, London, and Madrid, respectively. Indeed, in Athens, all the visits are concentrated in only 15 POIs (i.e., 50% of the POIs). The number of POIs with no visits in Barcelona, London, and Madrid are 7, 5, and 10, respectively.

Fig. 4 shows the CDF of the expected POI visiting time (in minutes) per city. Recall that the expected visiting time at a POI $v_i \in \mathcal{V}$ is computed using Equation (2). In all cities, the expected POI visiting time is short, for example, 50% lasts less than 20 minutes and 90% lasts less than 1 hour. Athens presented the distribution with the shortest expected POI visiting time, in which 60% lasts less than 4 minutes. The longest expected POI visiting time found in Athens, Barcelona, London, and Madrid are 91, 117, 155, and 108 minutes, respectively.

Finally, we analyze the distribution of the interest in the categories of the POIs. Recall that the interest of a tourist $u \in \mathcal{U}$ in a category $c \in \mathcal{C}$ is calculated using Equation (3). Fig. 5 presents the tourist's interest per POI category in our database for each city. In total, there are 13 POI categories. For comparison purposes, the interest is displayed as a percentage of the sum of the interests. In Athens and Barcelona, tourist's interest is concentrated in a few categories. For example, in Athens, all the interest is related to Education, Museum, and Historical. On the other hand, tourist's interest in London and Madrid is more diffused among the categories. For example, in London, no category receives more than 25.5% of the interest.

In the next sections, we use the obtained travel histories and the visiting-related statistics obtained from our dataset as inputs to MEC-PIR. We also use these travel histories as ground truth for the subsequent evaluation.

6.2 Baseline algorithm

We compare MEC-PIR against PersTour [6], a state-of-the-art algorithm for recommending personalized itinerary for an individual tourist based on POI popularity and tourist interest preferences. PersTour solves the PersTour problem using an Integer Programming Model. However, it does not take into account MEC resources and service application demands while generating the itinerary. Thus, in order to use PersTour to solve the MEC-PTR problem, we adapt the algorithm as follows:

- Since PersTour is designed for a single tourist, in order to produce itineraries for every tourist $u \in \mathcal{U_D}$, we run $|\mathcal{U_D}|$ independent instances of PersTour, i.e., one instance per tourist.
- We use the single itinerary per tourist that is generated by PersTour as input to our implementation of the second stage of MEC-PTR. Since PersTour is unaware of the limited resources, this approach may lead to infeasible solutions. When this happens, we solve the problem by relaxing the constraints dealing with minimum allocation for service applications. Thus, in terms of user experience, *the effective performance of PersTour is even worse than will be shown in the following section.*

We call this modified version of PersTour as Resource-aware PersTour (RA-PersTour).

6.3 Evaluation methodology

In our evaluation, we consider that two service applications are available for tourists: augmented reality and video streaming (i.e., $\mathcal{A} = \{\text{augmented reality}, \text{video streaming}\}$). The requirements of an augmented reality MEC service were taken from [24], while the requirements of a video streaming application were based in the Netflix stream service website[5]. Table 3 shows the network and processing requirements for both service applications.

TABLE 3
Requirements of the service applications.

Application	λ_a^{min}	λ_a^{max}	ψ_a^{min}	ψ_a^{max}
Augmented reality	1 Mbps	10 Mbps	0.1 RC	1 RC
Video streaming	1.5 Mbps	25 Mbps	0 (None)	0 (None)

Based on [25], we assume the amount of connectivity resources available at each POI is $\lambda_{v_i} = 75$ Mbps. As for computing resources, we consider two MEC hosts ($|\mathcal{M}| = 2$), each one with 37.5 Reference Cores (RCs), i.e., $\psi_m = 37.5$ RC. Based on [24], we assume that one RC is equivalent to the processing power of an Intel Haswell i7-4770 3.40GHz.

For each city, we use the valid travel sequences found in our dataset and the tourist interests derived from them as tourist profiles. Based on theses profiles, we generate a set

[5] https://help.netflix.com/en/node/306

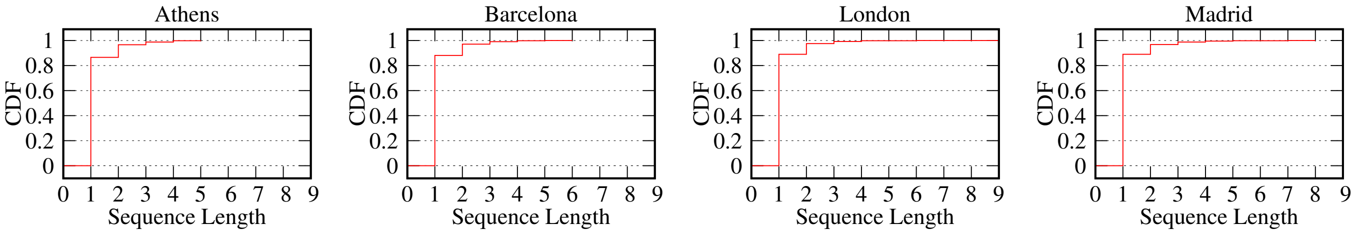

Fig. 2. Distribution of the length of the travel sequences.

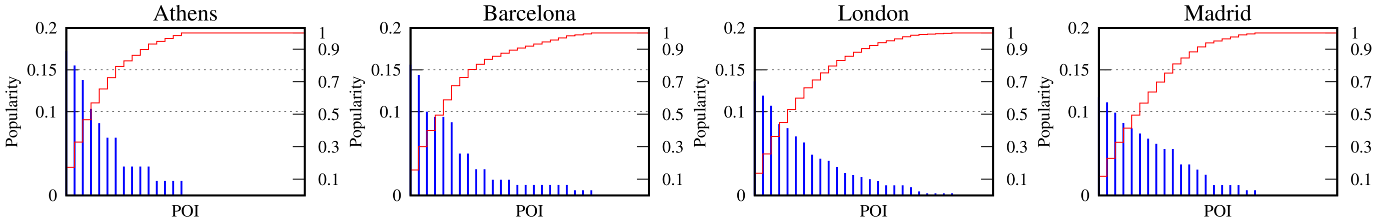

Fig. 3. Distribution of the popularity of the POIs. Histograms are shown in blue. CDFs are shown in red.

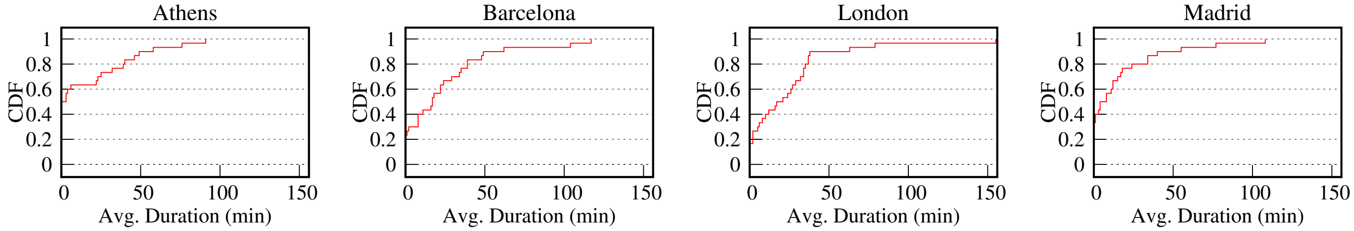

Fig. 4. Distribution of the expected POI visiting time.

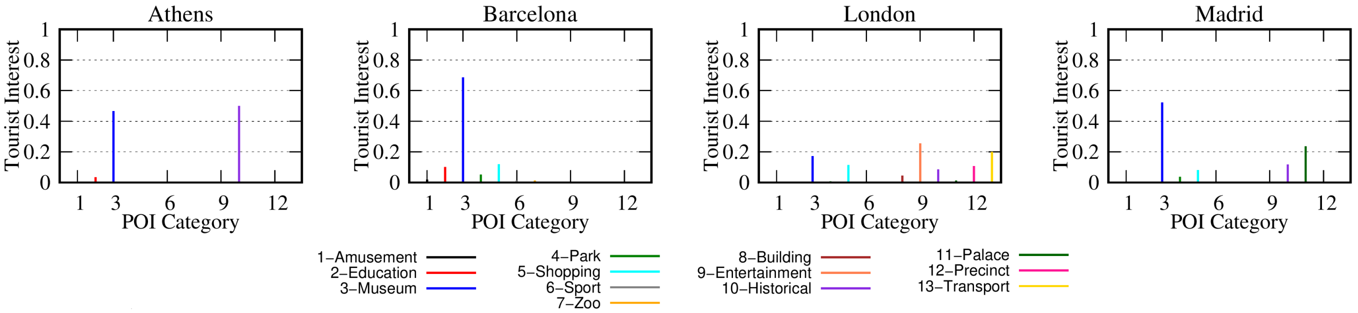

Fig. 5. Distribution of the tourist interest in the POI categories.

with 250 tourists for each city, i.e., for each city, we consider $|\mathcal{U}_\mathcal{D}|$ =250. For each tourist $u \in \mathcal{U}_\mathcal{D}$, we associate one service application profile selected randomly from the set $2^{|\mathcal{A}|}$, the power set for $\mathcal{A}$. By applying this approach, some tourists may be associated with no service application, which is also the truth in the real world.

For each city, we provide the set of POIs ($\mathcal{V}$), the ICT infrastructure information, and the set of tourists along with their preference and application profiles to MEC-PIR and RA-PersTour. We evaluate both algorithms using leave-one-out cross-validation [26]. To this end, we use all travel sequences of a tourist to build her interest ($Int_u(c), \forall c \in \mathcal{C}$), except the one we want to test. We also use the starting POI (v^u_s), ending POI (v^u_f), and budget (B_u) of the testing travel sequence as the corresponding starting POI, ending POI, and budget of the tourist profile.

Similar to other works related to recommendation algorithms [6], [11], [12], we use metrics such as **Recall**, **Precision**, and **F-score** to evaluate MEC-PIR against RA-PersTour. We also introduce two new metrics, namely, **Allocation Efficiency** and **User Experience** to assess the performance of the algorithms on allocating resources in the edge and the overall user experience provided by the recommended itinerary. The metrics are detailed in the following.

- **Recall**: the fraction of POIs from the real-world testing travel sequence that also exists in the generated itinerary. Let S^k_u be the real-world testing travel sequence and I^j_{u*} be the generated itinerary. The **Recall** is defined as: $Recall(I^j_{u*}) = |I^j_{u*} \cap S^k_u| \;/\; |S^k_u|$.
- **Precision**: the fraction of POIs from the generated itinerary (I^j_{u*}) that also exists in the real-world testing travel sequence (S^k_u), i.e., $Precision(I^j_{u*}) = \frac{|I^j_{u*} \cap S^k_u|}{|I^j_{u*}|}$.
- **F-score**: the model accuracy represented by:

$$\text{F-score}(I^j_{u*}) = \frac{2 \times Precision(I^j_{u*}) \times Recall(I^j_{u*})}{Precision(I^j_{u*}) + Recall(I^j_{u*})}.$$

- **Allocation Efficiency (AE)**: the amount of computing and network resource allocated to the generated itinerary (I^j_{u*}), relative to the maximum amount demanded by the set of chosen service applications (λ^{max}_u and ψ^{max}_u). This is only calculated for those tourists that make use of at least one service application and it is defined as:

$$AE(I^j_{u*}) = \sum_{v_i \in \mathcal{V}} \frac{p(I^j_{u*}, v_i)}{2|I^j_{u*}|\lambda^{max}_u} + \sum_{v_i \in \mathcal{V}} \sum_{m \in \mathcal{M}} \frac{q(I^j_{u*}, v_i, m)}{2|I^j_{u*}|\psi^{max}_u}.$$

- **User Experience (UE)**: determines the relation between the allocation efficiency associated with the generated itinerary (AE(I^j_{u*})) and the profit as perceived by the tourist for this itinerary ($Prof(I^j_{u*})$). The rationale is that high **AE** alone does not make an itinerary good/better, and **UE** also embeds the tourist interest within it. Formally, $UE(I^j_{u*}) = AE(I^j_{u*})Norm(Prof(I^j_{u*}))$.

Table 4 along with Table 3 summarize the main parameters and their values employed in the evaluation.

TABLE 4
Model parameters.

Parameter	Value
$\mathcal{A}$	{augmented reality, video streaming}
λ_{v_i}	75 Mbps
ψ_m	37.5 RCs
$\|\mathcal{U}_{\mathcal{D}}\|$	250
$\|\mathcal{V}\|$	30
$\|\mathcal{M}\|$	2
α	0.5
t_{split}	8 hours
Δt	1 minute
T	480 minutes

All experiments were performed in a virtual machine (VM) that runs Debian 9 GNU/Linux and is configured with 16 vCPUs, 64 GB RAM, and 40 GB of virtual disk. The VM is hosted in a server with 2 Intel Xeon Silver 4114 @ 2.20GHz. In MEC-PIR, the first stage of the MEC-PTR problem is implemented using C++. The second stage of the problem is implemented using Python 2.7.12, docplex 2.8.125, and IBM CPLEX 12.8.0 (as the solver). The PersTour algorithm is also implemented using Python 2.7.12, docplex 2.8.125, and IBM CPLEX 12.8.0.

6.4 Results and discussion

As described in Section 6.3, we provide the same input to MEC-PIR and to RA-PersTour. The output of both algorithms is a set $\mathcal{I}_*$, containing one itinerary for each tourist $u \in \mathcal{U}_{\mathcal{D}}$. For both sets, we compute the metrics **Recall**, **Precision**, **F-score**, **AE**, and **UE** for every itinerary $I^j_{u*} \in \mathcal{I}_*$. Table 5 summarizes the performance achieved by both algorithms in each metric, for each city considered in the evaluation. These results correspond to the average of the values found for the itineraries.

By design, *our algorithm may accept some penalty in the scores in order to improve resource allocation and user experience.* In practice, this approach has shown that the negative impact is negligible. MEC-PIR outperforms RA-PersTour in all metrics, except **Recall** of three cities. Even though, the difference was very low. The advantages in performance are higher for the metrics **AE** and **UE**, meaning that *MEC-PIR has the ability to choose itineraries that provide better resource allocation and better overall user experience.* Since our algorithm achieves better performance also in the metric **F-score**, this means that *MEC-PIR is also satisfying the user interest.* In the evaluated dataset, MEC-PIR improved in at least 19% the **UE**, which corresponds to the city of London. In the city of Athens, our algorithm improved in more than 100% the **UE**.

TABLE 5
Average performance of MEC-PIR and RA-PersTour.

		Athens			
Algorithm	**Recall**	**Precision**	**F-score**	**AE**	**UE**
MEC-PIR	0.859	0.325	0.456	0.564	0.485
RA-PersTour	0.863	0.301	0.432	0.261	0.234
Gain	-0.47%	**+7.98%**	**+5.61%**	**+116%**	**+107%**
		Barcelona			
Algorithm	**Recall**	**Precision**	**F-score**	**AE**	**UE**
MEC-PIR	0.772	0.478	0.555	0.665	0.397
RA-PersTour	0.771	0.464	0.546	0.501	0.307
Gain	**+0.04%**	**+2.91%**	**+1.71%**	**+33%**	**+29%**
		London			
Algorithm	**Recall**	**Precision**	**F-score**	**AE**	**UE**
MEC-PIR	0.816	0.45	0.543	0.817	0.42
RA-PersTour	0.845	0.433	0.541	0.663	0.351
Gain	-3.59%	**+4.03%**	**+0.42%**	**+23%**	**+19%**
		Madrid			
Algorithm	**Recall**	**Precision**	**F-score**	**AE**	**UE**
MEC-PIR	0.77	0.533	0.592	0.57	0.248
RA-PersTour	0.788	0.496	0.574	0.459	0.206
Gain	-2.33%	**+7.33%**	**+3.12%**	**+24%**	**+20%**

To illustrate how the metric values are distributed over the itineraries belonging to the solution ($\mathcal{I}_*$), Fig. 6 shows the CDF of the evaluation metrics for both algorithms. According to the CDFs of **AE** and **UE**, in addition to achieving better average performance than RA-PersTour, MEC-PIR also systematically performs better resource allocation and better overall user experience. In other words, *the average improvement of MEC-PIR does not come from punishing certain tourists in order to benefit others.* Naturally, the level of improvement varies according to the resource constraints faced by each tourist along her tour. On the other hand, MEC-PIR works with the Pareto front solutions for a tourist, which means a set of solutions in which may exist itineraries with scores lower the highest. This means that we were aware of some impact in the traditional metrics, i.e., **Recall**, **Precision**, and **F-score**. However, the size and the extent of this impact was unknown in advance. As shown in Fig. 6, the negative impact is negligible for all tourist, i.e., no one has her score notably affected in order to improve the overall gain. Actually, in most of the traditional metrics, *MEC-PIR outperforms RA-PersTour.*

Finally, Fig. 7 shows the execution time of MEC-PIR

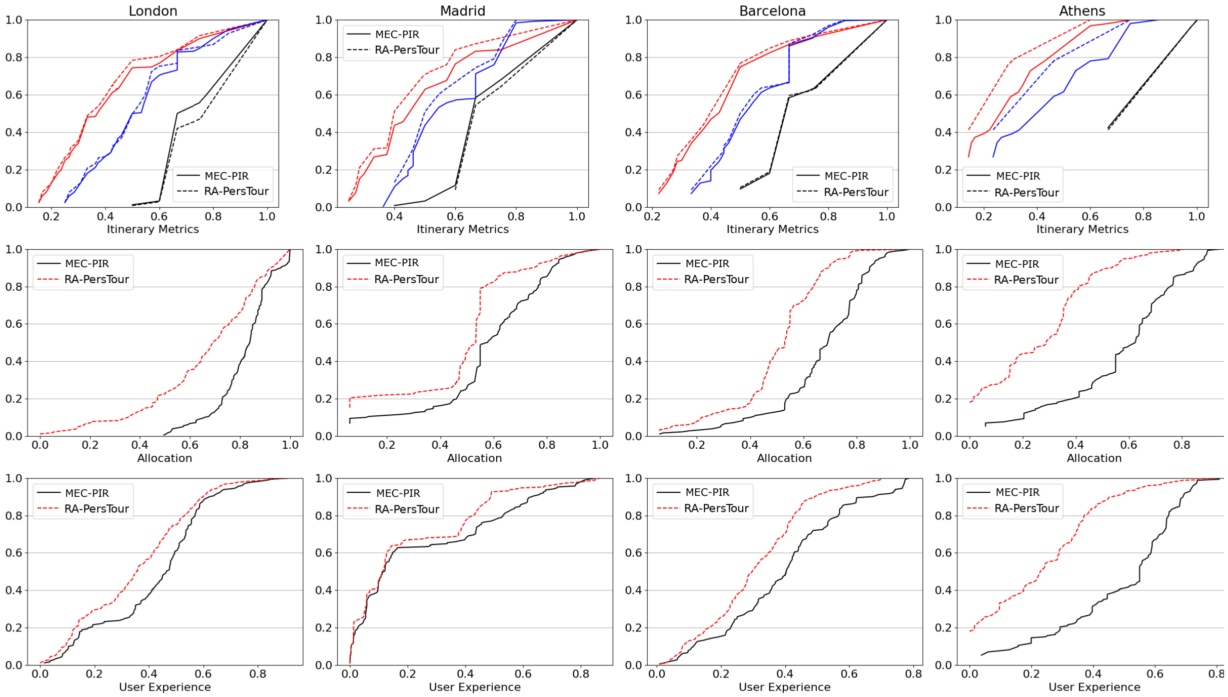

Fig. 6. CDF of the evaluation metrics. In the CDF of the Itinerary Metrics, black, red and blue colors representing, respectively, the **Recall**, **Precision** and **F-score**, with the performance of MEC-PIR being displayed with continuous lines and that of RA-PersTour represented by dashed lines. Each column represents the results for a city.

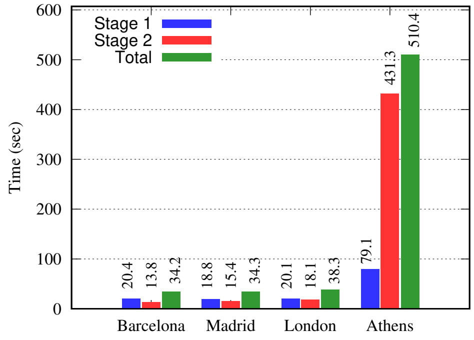

Fig. 7. MEC-PIR execution time.

in each stage per city. For all cities, with the exception of Athens that contains a large set of candidate itineraries to choose from (with more than 9,000 paths), MEC-PIR takes less then 40 seconds to solve the entire MEC-PTR problem. For Athens, the entire problem takes less than 10 minutes to be solved and the vast majority of the time is spent to solve the second stage of the problem, which depends on the solver performance. Indeed, *our approach to solve the first stage of the problem using a dynamic programming version of the ESPPRC has proven to be very efficient*, taking less than 2 minutes for Athens and less than 30 seconds for the other cities. This is an important result since the databases of other cities may demand scalability similar to the one observed in Athens. In summary, *the experimental results demonstrate that MEC-PIR achieves gains from near 20% up to more than 100% in both resource allocation efficiency and user experience, while presenting performance similar to RA-PersTour in the traditional metrics for evaluation of personalized itinerary recommendation.* In addition, MEC-PIR solves the MEC-PTR problem for instances of realistic size in a reasonable amount of time.

7 Conclusion

Our work is motivated by the next-generation touristic services, where tourists will augment their visiting experience with service applications employing Multi-access Edge Computing. In this context, we formalize the problem of recommending personalized itineraries for tourists, i.e., taking into consideration not only the POIs and user interests, but also (i) the resource demands of the used service applications; and (ii) the availability of MEC resources to support the services, defined as MEC-PTR.

The objective of MEC-PTR is to improve the overall physical and virtual experience of the tourists by maximizing the sum of scores for the chosen itineraries and prioritizing resource allocation at the network edge, while respecting the capacity of the computing and network resources. We introduced the MEC-PIR algorithm, which efficiently solves the MEC-PTR problem. We compare MEC-PIR with RA-PersTour, a state-of-the-art solution adapted to this scenario. Results show that MEC-PIR performs better than RA-PersTour from near 20% up to 100% in resource allocation efficiency and also in user experience, while presenting a similar performance in traditional metrics of personalized itinerary recommendation.

Although satisfactory results were obtained with the Dynamic Programming version of the ESPPRC for the first stage of the problem, the second stage may face scalability issues under specific conditions, as suggested by the evaluation of Athens city. As a future work, we intend to investigate some meta-heuristic approaches, e.g., Ant Colony Optimization, in order to speed up the second stage. We also plan to include a degree of uncertainty in the model, since not all tourists may follow the recommendation and may, for example, extrapolate the visiting time, affecting (or being affected by) the resource allocation. We are also interested in other variants of the personalized tour problem that can be investigated under the context of the next-generation touristic services. For example, the multi-day tours [12], in which the maximization of the overall score is not simply maximizing the score of each day isolated.

References

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[2] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[3] World Travel and Tourism Council, "Travel & Tourism Performance, 2019," https://wttc.org/Research/Economic-Impact, 2019, [Last access: May-24-2020].

[4] K. Katsaros, D. Gkounis, D. Kaleshi, B. Thomas, J. Harris, H. Falaki, and D. Simeonidou, "Enhancing Tourist Experiences through 5G - The 5G Smart Tourism Case Study," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 471–476.

[5] 5GBarcelona, "Case Studies," https://5gbarcelona.org/pilots/5g-interactive-city/, 2020, [Last access: May-27-2020].

[6] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized trip recommendation for tourists based on user interests, points of interest, visit durations and visit recency," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 375–406, 2018.

[7] T. Tsiligirides, "Heuristic Methods Applied to Orienteering," *Journal of the Operational Research Society*, vol. 35, pp. 797–809, 1984.

[8] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[9] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, pp. 291–328, 2014.

[10] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu, "Automatic Construction of Travel Itineraries Using Social Breadcrumbs," in *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, 2010, p. 35–44.

[11] I. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, "Where Shall We Go Today?: Planning Touristic Tours with Tripbuilder," in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, 2013, pp. 757–762.

[12] Z. Friggstad, S. Gollapudi, K. Kollias, T. Sarlos, C. Swamy, and A. Tomkins, "Orienteering Algorithms for Generating Travel Itineraries," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 180–188.

[13] F. F. Fonseca, L. Mamatas, A. C. Viana, S. L. Correa, and K. V. Cardoso, "Personalized Travel Itineraries with Multi-Access Edge Computing Touristic Services," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, p. 1–6.

[14] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, "Tour recommendation and trip planning using location-based social media: a survey," *Knowledge and Information Systems*, vol. 60, no. 3, pp. 1247–1275, 2018.

[15] P. Vansteenwegen and D. Van Oudheusden, "The Mobile Tourist Guide: An OR Opportunity," *OR Insight*, vol. 20, no. 3, pp. 21–27, 2007.

[16] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next Location: A Recurrent Model with Spatial and Temporal Contexts," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[17] Y. Su, X. Li, W. Tang, J. Xiang, and Y. He, "Next Check-in Location Prediction via Footprints and Friendship on Location-Based Social Networks," in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, 2018, pp. 251–256.

[18] L. Yao, Q. Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, and Q. He, "Context-Aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, p. 1007–1010.

[19] L. Ravi, V. Subramaniyaswamy, V. Vijayakumar, S. Chen, A. Karmel, and M. Devarajan, "Hybrid Location-based Recommender System for Mobility and Travel Planning," *Mobile Networks and Applications*, vol. 24, no. 4, pp. 1226–1239, 2019.

[20] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Towards next generation touring: Personalized group tours," in *Proc. of ICAPS'16*, 2016, pp. 412–420.

[21] Y. Yin, J. Xia, Y. Li, y. xu, W. Xu, and L. Yu, "Group-wise itinerary planning in temporary mobile social network," *IEEE Access*, vol. 7, pp. 83 682–83 693, 2019.

[22] D. Feillet, P. Dejax, and M. Gendreau, "Traveling Salesman Problems with Profits," *Transportation Science*, vol. 39, no. 2, pp. 188–205, 2005.

[23] S. Irnich and G. Desaulniers, *Shortest Path Problems with Resource Constraints*. Springer US, 2005, pp. 33–65.

[24] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

[25] F. Malandrino, C. F. Chiasserini, G. Avino, M. Malinverno, and S. Kirkpatrick, "From Megabits to CPU Ticks: Enriching a Demand Trace in the Age of MEC," *IEEE Transactions on Big Data*, 2018.

[26] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

Sand Luz Correa is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where she has been a professor and researcher since 2010. She holds a degree in Computer Science from Universidade Federal de Goiás (1994), has MSc (1997) in Computer Sciences from University of Campinas (UNICAMP) and PhD (2011) in Informatics from Pontifical Catholic University of Rio de Janeiro (PUC-Rio). In 2015, she spent her sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). Professor Sand has participated in some international research projects (including two from joint calls BR-EU). Her research interests include cloud computing, SDN, data plane programmability, and sensor systems and smart city platforms.

Kleber Vieira Cardoso is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where he has been a professor and researcher since 2009. He holds a degree in Computer Science from Universidade Federal de Goiás (1997), has MSc (2002) and PhD (2009) in Electrical Engineering from COPPE - Universidade Federal do Rio de Janeiro. In 2015, he spent his sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). He has participated in some international research projects and coordinated several national-sponsored research and development projects. His research is focused on the following topics: wireless networks, software-defined networks, virtualization, resource allocation, and performance evaluation.

Felipe F. Fonseca holds a degree in Computer Science from the Universidade Federal Goiás. From mid 2018 until the beginning of 2020, he was part of the LABORA research group (in UFG), where he developed works in the area of Resource Allocation in Wireless Networks.

Lefteris Mamatas is an Assistant Professor with the Department of Applied Informatics, University of Macedonia, Greece. He leads the Softwarized and Wireless Networks Research Group (http://swn.uom.gr), University of Macedonia. He has worked as a Researcher with the University College London, U.K., the Space Internetworking Center/Democritus University of Thrace, Greece, and DoCoMo Eurolabs, Munich. His research interests include software-defined networks, the Internet of Things, 5G networks, and multi-access edge computing. He has published more than 60 articles in international journals and conferences.

Aline Carneiro Viana is a permanent Reseach Director (DR) at Inria, where she leads the team TRiBE team. After a 1-year sabbatical leave at the TKN Group of the TU-Berlin, Germany, she got her habilitation degree from UPMC - Sorbonne Universités, France in 2011. Dr. Viana got her PhD in Computer Science from the UPMC - Sorbonne Universités in 2005. Her research addresses the design of solutions for tactful networking, smart cities, mobile and self-organizing networks with the focus on human behavior analysis. She is a recipient of the French Scientific Excellence award since 2015 and for 6 years now and was nominated in 2016 as one of the "10 women in networking/communications that you should Watch" (1st-year nomination of N2Women community).