IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received April 9, 2020, accepted April 30, 2020, date of publication May 4, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992275

Interference Aware Service Migration in Vehicular Fog Computing

SHUXIN GE[1], MENG CHENG[2], (Member, IEEE), AND XIAOBO ZHOU[1], (Senior Member, IEEE)

[1]Tianjin Key Laboratory of Advanced Networking, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[2]School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi 923-1292, Japan

Corresponding author: Meng Cheng (m-cheng@jaist.ac.jp)

This work was supported by the Natural Science Foundation of Tianjin under Grant 17JCQNJC00700, Grant 18ZXZNGX00040, and Grant 18ZXJMTG00290.

ABSTRACT Vehicular Fog Computing (VFC) is a promising technique to enable ultra low service latency by exploiting the computation and storage resources of both Roadside Units (RSUs) and Serving Vehicles (SVs) such as buses and trams with rich resources. To tackle with the mobility of vehicles, the services are usually migrated between RSUs and SVs, i.e., follow the vehicle, to maintain the benefits of VFC. However, making optimal service migration decisions in VFC is challenging due to the mobility of SVs and the interference between vehicles. In this paper, we investigate multi-vehicle service migration problem in VFC. We propose an efficient online algorithm, called FEE, to optimize the service migration for each vehicle in each time slot, where the latency in the current time slot, the expected latency in future time slots, and the interference among vehicles are minimized. The expected latency in future times slots is obtained by trajectory prediction based on hidden Markov model, and the interference is measured based on the server load. Finally, a series of simulations based on real-world mobility traces of Rome taxis are conducted to verify the superior performance of the proposed FEE algorithm as compared with the state-of-the-art solutions.

INDEX TERMS Service migration, vehicular fog computing, hidden Markov model, interference detection.

I. INTRODUCTION

With the development of autonomous driving, a variety of novel computation-intensive and delay-sensitive vehicular services, e.g., surrounding vehicle perception, high definition (HD) mapping, has emerged and posed great challenges on the conventional vehicular networks that supported by cellular networks and cloud computing [1], [2]. By deploying edge servers at the Roadside Units (RSUs), it is possible for the RSUs to provide services to the passing by vehicles and hence greatly alleviates this problem, which is referred to as Vehicular Edge Computing (VEC) [3]. However, VEC is still far from perfect due to the limited resources of the edge servers, and poor mobile coverage in rural areas [4]. Given the tremendous amount of vehicles with additional communication and computation resources, it is beneficial to fully utilize these resources, which is the concept of Vehicular Fog Computing (VFC) [5], [6].

In VFC, besides the RSUs, we can also deploy the services on some vehicles, especially those with relative rich communication and computation resources such as buses and trams [7]. We refer this kind of vehicles as Serving Vehicles (SVs). Without otherwise specified, we use vehicle to refer to the rest of the other ordinary vehicles in the network [8]. VFC leverages SV mobility to enhance the service capacity as it increases the communication opportunities of vehicles, including the service range and resources utilization [9]. This is because that SVs play the role of communication hubs, where the near-located vehicles can be connected together and further connect to more access points [10]. With VFC, vehicles can request service from either RSUs or SVs with ultra-low latency.

To tackle with vehicle mobility, a proper service migration strategy is needed to achieve a balance between the migration cost and the service latency [13]. However, making optimal service migration decisions in VFC is challenging due to the following reasons [14]–[16]. First, the mobility of SVs affects the optimality of the service migration strategy. When a vehicle has a similar trajectory with a SV over a period of time, the real-time Vehicle-to-Vehicle (V2V) communications greatly decreases the service latency [11]. However, the service will be frequently migrated when the vehicle

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

requests this service from a SV which just passing by and heading to a different direction, and even lead to the loss of service data [12]. Thus, the migration cost of vehicles may be misestimated, resulting in invalid migration [17]. Second, the performance of migration strategy greatly relies on the accuracy of the trajectory prediction of both the SVs and vehicles [18]. However, the existing prediction methods, e.g. Markov Decision Process (MDP), become intractable with large number of vehicles [19]. Third, the interference among different vehicles decreases the quality of experience (QoE). When the vehicles make migration decisions independently, they suffer from interference from each other as the resources of the edge servers are limited [20]. To the best of our knowledge, an efficient service migration strategy in VFC is still missing.

In this paper, we investigate multi-vehicle service migration in VFC, and develop an efficient solution to make optimal service migration decision for each vehicle. The main contributions of this paper are summarized as follows:

- We formalize the service migration process in VFC networks as a mixed integer nonlinear programming MINP) program for minimizing the average service latency, where the QoE constraints and the mobility of SVs are taken into account. To our best knowledge, this is the first work that studies service migration with SVs in VFC networks.
- We propose an efficient online algorithm, called inter**F**erence awar**E**e s**E**rvice migration (FEE), to decouple the migration process between different vehicles, services and time slots, where the latency in the current time slot, the expected latency in future time slots, and the interference between vehicles are minimized. The expected latency in future times slots is obtained by trajectory prediction based on hidden Markov model, and the interference between vehicles is measured based on the server load. This makes our algorithm scalable to multi-vehicle networks with low time complexity.
- The effectiveness of the proposed FEE algorithm is verified by simulations on the basis of real-world taxi traces in Rome. The average service latency, and service deadline guarantee rate of FEE are compared with other four benchmarks. The simulation results demonstrate the superior performance of FEE with SVs mobility, especially with large vehicle numbers.

The rest of this paper is organized as follows. Section II reviews related work. The system model considered in this paper is present in Section III. Section IV propose FEE algorithm, along with the complexity analysis. Section V shows the performance evaluation and discussion. Finally, the conclusion is drawn in Section VI.

II. RELATED WORK

Recently, different architectures of VFC have been proposed in the literature. Satyanarayanan [21] regards each vehicle as a fog node and chooses a coordinator for each area. Xiao and Zhu [5] select the commercial fleet as the fog node. Hou *et al.* [8] suggest slow moving vehicle or parked vehicle provide its residual resources. In this section, we briefly survey existing literature in service migration from the perspective of VFC networks [22], [23], service latency and trajectory prediction [24].

The major studies in VFC focus on satisfying the high-dynamic network topology caused by SVs. Liu *et al.* [23] use software-defined networking (SDN) technology to enhance the QoE of latency-sensitive service in a VFC-based networks. Zhang *et al.* [25] adaptively upload the tasks to the fog computing servers by directly uploading or relay transmission in VFC-enable vehicular ad hoc networks to minimize the average latency. Yan *et al.* [26] propose a centralized scheme to schedule forwarders to manage the service requests in SVs during its driving route, and thus achieve minimum transmission latency. Zhou *et al.* [27] use the multi-armed bandit theory to maintain the delay performances in the decision-making process, where the appearance time of SVs is taken into account. Yao *et al.* [6] propose a scheme to accommodate low latency requirement. This scheme caches service data at a set of vehicle nodes that may repeatedly visit the same hot spot. However, all above researches ignore the cost of service migration, as well as the different characteristics of services (e.g. latency-sensitive and computing-intensive) and SVs (e.g., velocity and type).

Many previous service migration strategies help to reduce the service latency [28]. Nadembeg *et al.* [14] selecting the optimal servers in terms of offered throughput and splitting user requested service into several portions, which the objective is minimizing the service latency. Ouyang *et al.* [12] introduce a scheme to reduce system costs over time in compliance with the time-varying resource situation. Zhang *et al.* [29] propose an online service migration method based on network efficiency optimization to maintain latency performance while optimizing energy efficiency. However, these works neglect the interference among users, which leads to QoE deterioration.

Meanwhile, to account for the mobility of vehicles, existing works balance routing latency and migration latency adaptively by predicting vehicle trajectory in the following time [30]. Assuming the user moves in a straight line, i.e., following a one-dimensional (1-D) mobility pattern, Ksentini *et al.* [31] propose a method to predict user trajectory by MDP model, and then decide the course of service migration. Plachy *et al.* [32] use low-level channel information to predict the user trajectory to choose a best communication path for migration, where the load of communication and computing resources of the target base station (BS) is taken into account. Considering the more realistic case, Wang *et al.* [24] make the service migration decision by predicting the user trajectory with a 2-D Markov Decision Process (MDP). However, it is extremely challenging to maintain the prediction accuracy of future information, such as user trajectory owing to the increasing number of users. Such observations motivate us to propose FEE algorithm, which is expected to

utilize the HMM and specific interference detection function to optimize latency performance.

III. SYSTEM MODEL

A. SYSTEM OVERVIEW

In this paper, we consider a VFC networks consisting of N_1 fixed position BSs and N_2 SVs, and V vehicles, which are geographically distributed over N_a regular square area as shown in Fig. 1. Note that SVs usually follow some fixed routines (e.g., buses, delivery trucks, and garbage trucks) [26]. In the following, we use BS to denote both fixed position BSs and SVs, and let S_n denote the storage capacity of BS $n, n \in [1, N]$, where $N = N_1 + N_2$.

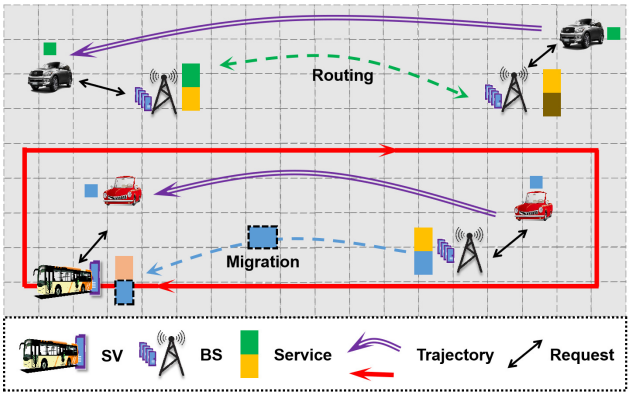


FIGURE 1. The overview of the system considered in this paper.

There are M different vehicular services deployed in the BSs. Let λ_m, γ_m, D_m and Ω_m denote the input data size, the computation intensity (i.e. CPU cycles/bit), the response deadline and the priority of service $m, m \in [1, M]$, respectively. Let $f_{m,v}, \theta_{m,v}$ denote the CPU cycle and storage requirements of a service request from vehicle $v, v \in [1, V]$ to service m, respectively. The vehicle around the N_a areas connects to the BS with the largest received signal strength indicator. When the BS receives a service request from a vehicle, it will response to the request once this service is deployed locally, otherwise it will forward the request to another BS that has this service. We divide the total time into T time slots, where each time slot lasts for τ seconds. In time slot $t, t \in [0, T-1]$, vehicles generate service requests according to some stochastic statistics. Table. 1 summarizes the key parameter notations in our paper.

B. SERVICE PLACEMENT AND MIGRATION

We use a vector $\boldsymbol{B}_t = \{B_t(v)|v = 1, 2, \cdots, V\}$ to denote the connected BSs of the vehicles in time slot t, where

$$B_t(v) \in [1, N], \quad \forall v, t. \tag{1}$$

We use another vector $\boldsymbol{b}_{t,m} = \{b_{t,m}(v)|v = 1, 2, \cdots, V\}$ to denote the serving nodes of the requests of vehicles to service m in time slot t, where

$$b_{t,m}(v) \in [0, N], \quad \forall v, t, m. \tag{2}$$

TABLE 1. Definitions of notations.

Notation	Definition
$\boldsymbol{B}_t(v)$	the connected BS of the vehicles v in time slot t
$\boldsymbol{b}_{t,m}(v)$	the serving node of the request of vehicles v to service m in time slot t
$\boldsymbol{b}'_{t,m}(v)$	the migration decision of vehicle v to service m in time slot t
S_n	the storage capacity in BS n
λ_m	the input data size of service m
γ_m	the computation intensity of service m
D_m	the response deadline of service m
Ω_m	the priority of service m
$f_{m,v}$	the CPU cycle requirements of a service m request from vehicle v
$\theta_{m,v}$	the storage requirements of a service m request from vehicle v
$R_t(i,j)$	The transmit rate between BS i and BS j in time slot t
$l^T_{t,m}(v)$	the transmission latency of vehicle v for service m in time slot t
$l^C_{t,m}(v)$	the computing latency of vehicle v for service m in time slot t
$l^M_{t,m}(v)$	the migration latency of vehicle v for service m in time slot t
$L_{t,m}(v)$	the total latency of vehicle v for service m in time slot t
Δ	the length of predicted serving node sequence
A_t	the state transition probability matrix in time slot t
B_t	the observation probability matrix in time slot t
$L^{\Delta}_{t,m}(v)$	the expected latency in future Δ time slots of vehicle v for service m
$\Psi(t,m)(v)$	the interference indicator in time slot t of vehicle v for service m

Note $b_{t,m}(v) = 0$ indicates vehicle v does not send request to service m in time slot t. Let $\boldsymbol{b}'_{t,m} = \{b'_{t,m}(v)|v = 1, 2, \cdots, V\}$ denote the migration decision of service m in time slot t, where

$$b'_{t,m}(v) \in [0, N], \quad \forall v, t, m. \tag{3}$$

Note that $b'_{t,m}(v)$ indicates a service migration happens. Thus, the serving node in next time update by

$$b_{t+1,m}(v) = b'_{t,m}(v) \tag{4}$$

When multiple services are placed at BS n, the total storage requirements should not exceed its storage capacity, therefore we have

$$\sum_{m=1}^{M}\sum_{v=1}^{V} \theta_{m,v}\mathcal{H}(b'_{t,m}(v), n) \le S_n, \quad \forall t, n. \tag{5}$$

where

$$\mathcal{H}(x, y) = \begin{cases} 1, & x = y, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

C. SERVICE LATENCY

Service latency consists of transmission latency between the vehicle and the BS, computing latency and migration latency. The transmit rate between BS i and BS j in time slot t is

$$R_t(i,j) = W \log(1 + \frac{P \cdot d(i,j,t)^{-2} \cdot |h|^2}{N_0}) \tag{7}$$

where W, P and N_0 are the channel bandwidth, transmit power and noise power, respectively. h is a complex fading vector modeled as circularly symmetric complex Gaussian random variables [33]. $d(i, j, t)$ is the distance between BS i and BS j in time slot t. Assuming vehicle v generates a request to service m at the beginning of time slot t, the request transmission latency is

$$l^T_{t,m}(v) = \frac{\lambda_m}{R_t(B_t(v), b_{t,m}(v))} + C, \tag{8}$$

where C is a constant value to approximate the transmission time between vehicle v and BS $B_t(v)$.

The computation latency of vehicle v for service m is

$$l^C_{t,m}(v) = \frac{\lambda_m \gamma_m}{f_{m,v}}. \tag{9}$$

Service m may be migrated from BS $b_{t,m}(v)$ to BS $b'_{t,m}(v)$ in time slot t. The service migration latency in time slot t can be expressed as

$$l^M_{t,m}(v) = \frac{\theta_{m,v}}{R_t(b_{t,m}(v), b'_{t,m}(v))}. \tag{10}$$

Note in the case that $b_{t,m}(v) = b'_{t,m}(v)$, service m is not migrated in time slot t, and thus we have $l^M_{t,m}(v) = 0$.

In summary, the service latency of vehicle v to service m in time slot t can be expressed as

$$L_{t,m}(v) = l^T_{t,m}(v) + l^C_{t,m}(v) + l^M_{t,m}(v). \tag{11}$$

D. PROBLEM FORMULATION

Our goal is to minimize the long-term average latency by selecting the optimal migration strategy $b'_{t,m}(v)$, which can be expressed as

$$\mathbf{P1}: \min_{\boldsymbol{b}'_{t,m}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{v=1}^{V} \sum_{m=1}^{M} L_{t,m}(v) \tag{12}$$

$$s.t.\ L_{t,m}(v) \le D_m, \forall t, v, m$$

$$(1), (2), (3), (5) \tag{13}$$

where constraint (13) ensures the request can be completed before its response deadline. The major challenge that impedes the derivation of the optimal solution is the lack of future information, e.g., the trajectories of SVs and vehicles, and the request distribution. Moreover, even if the future information is known, **P1** is still a MINP problem that is proven to be NP-hard.

IV. ONLINE SERVICE MIGRATION ALGORITHM

In this section, we present FEE algorithm, which is an online online service migration decision approach, to solve **P1**. First, we introduce a trajectory prediction algorithm to predict the trajectory of both the SVs and the vehicles in the next Δ time slots. Next, given the predicted trajectory, we obtain the expected service latency in the next Δ time slots. To decouple the migration decisions in different time slots, in each time slot, we minimize the current service latency and the expected service latency in the next Δ time slots by selecting the optimal migration decision. Finally, a specific interference prediction function is designed to avoid the interference among vehicles. Based on this, we can further convert the optimization problem in each time slot into a series of per-vehicle per-service sub-problems which can be solved.

A. PROBLEM TRANSFORMATION

In order to solve **P1** online, service migration in each time slot must take into account the service latency in the future under this service migration. Thus, the migration which migrate service to the SVs that just passing can be avoided. Let $L^{\Delta}_{t,m}(v)$ denote the service latency in next Δ time slot under the service migration of vehicle v for service m in time slot t, called by long-term latency, which will be introduced in Section. IV-B. Based on this, the objective function of **P1** is equivalent to minimize the service latency in current time slot and the long-term service latency, which is expressed as

$$\mathbf{P2}: \min_{\boldsymbol{b}'_{t,m}} \sum_{v=1}^{V} \sum_{m=1}^{M} \left[L_{t,m}(v) + \alpha L^{\Delta}_{t,m}(v) \right]$$

$$s.t.\ (1), (2), (3), (5), (13) \tag{14}$$

where $\alpha > 0$ is the weighting parameter, a high value of α mean the service migration decision take more into account long-term latency.

By eliminating the resource constraints in each BS, the migration decision of each vehicle and each service can be considered to be independent, i.e., the objective function of **P2** can be divided into $V \times M$ sub-problems. However, the optimal solution of these sub-problems may be invalid for **P2** if the resource constraints of the BSs are not satisfied. Thus, we define a interference indicator $\Phi_{t,m}(v)$ to represent the interference level of the service migration decision for vehicle v to service m in time slot t, which will be introduced in Section. IV-C.

Given the interference indicator, **P2** can be converted into $V \times M$ smaller problems **P3** with lower complexity.

$$\mathbf{P3}: \min_{\boldsymbol{b}'_{t,m}} L_{t,m}(v) + \alpha L^{\Delta}_{t,m}(v) + \beta \Omega_m \Phi_{t,m}(v)$$

$$s.t.\ (1), (2), (3), (13) \tag{15}$$

where $\beta > 0$ is the weighting parameter of the interference indicator. The detail of our proposed FEE scheme is shown in Algorithm. 1. For each time slot and each vehicle, we firs generate the predicted trajectory and serving node sequence by TP algorithm and SNP algorithm, which are introduced in Section. IV-B (Lines 3-4). For each service, we calculate the long-term latency and interference indicator, and thus we can directly solve **P3** to obtain the migration decision (Lines 5-10). Finally, we update the system state in next time slot according to our migration decisions (Line 11). As we greatly reduced the size of the state space, the overall complexity is reduced from $\mathbf{O}(V^{NM})$ to $\mathbf{O}(VNM)$ in each time slot t.

Note that there is a small probability that the migration decisions obtained from **P3** may be invalid if the resources

Algorithm 1 FEE Algorithm

Input: : Long-term Trajectory, α, β ω_1, ω_2, K, Δ, $\boldsymbol{B}_0$, $\boldsymbol{b}_{0,m}$
Output: Migration Decision $\boldsymbol{b}'_{t,m}$

```
for t = 1 to T do
    for v = 1 to V do
        Predict the trajectory of SVs and vehicles according to TP Algorithm;
        Predict serving node sequence according to SNP Algorithm;
        for m = 1 to M do
            Obtain L^Δ_{t,m}(v) according to (24);
            Obtain Φ_{t,m}(v) according to (31);
            Obtain b'_{t,m} by solving P3;
        end for
    end for
    b_{t+1,m} = b'_{t,m};
end for
```

TABLE 2. Historical trajectories of a vehicle.

Index	Trajectory	Probability
1	$\boldsymbol{G}_1 = \{g_1^1, g_1^2, \cdots, g_1^{l_1}\}$	p_1
⋮	⋮	⋮
h	$\boldsymbol{G}_h = \{g_h^1, g_h^2, \cdots, g_h^{l_h}\}$	p_h
⋮	⋮	⋮
H	$\boldsymbol{G}_H = \{g_H^1, g_H^2, \cdots, g_H^{l_H}\}$	p_H

constraints are not satisfied. The probability can be reduced by choosing a proper β value, as shown in Section V.

B. EXPECTED LATENCY PREDICTION BASED ON HIDDEN MARKOV MODEL

The process of predicting future delay is mainly divided into two steps. The first step is to predict the trajectory of SVs and vehicles, and the second step is to predict the serving nodes sequence to calculate the long-term latency.

1) TRAJECTORY PREDICTION

We collect the historical trajectories of each vehicle to predict its current trajectory. A set of H historical trajectories of a vehicle is shown in Table 2, where $\boldsymbol{G}_h = \{g_h^1, g_h^2, \cdots, g_h^{l_h}\}$, $h \in [1, H]$ is one trajectory. The elements of $\boldsymbol{G}_h$ are positions sequentially obtained from the GPS receiver when the vehicle moves along a certain route. p_h indicates the probability that the vehicle follows $\boldsymbol{G}_h$, which can be calculated by

$$p_h = n_h / N_H, \tag{16}$$

where n_h and N_H are the number of occurrences of trajectory $\boldsymbol{G}_h$, and the total number of trajectories collected. Once a latest trajectory is obtained, we add it to the historical trajectory table and update the probabilities accordingly. Two

Algorithm 2 Trajectory Prediction (TP) Algorithm

Input: $(g_*^1, g_*^2, \cdots, g_*^t)$, Δ, l_0
Output: $(g_*^{t+1}, g_*^{t+2}, \cdots, g_*^{t+\Delta})$

```
for h = 1 To H do
    Obtain l'_h according to (17);
end for
Obtain the candidate trajectory G_{h*} according to (18);
Return (g_{h*}^{l̄_{h*}+1}, g_{h*}^{l̄_{h*}+2}, ⋯, g_{h*}^{l̄_{h*}+Δ})
```

trajectories, $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$, are regarded as the same if $l_i = l_j$ and $\sum_{l=1}^{l_i} \|g_i^l, g_j^l\| \le \epsilon$, where ϵ is a threshold obtained empirically.

Given the historical trajectory table and the current trajectory $(g_*^1, g_*^2, \cdots, g_*^t)$, we use trajectory prediction (TP) algorithm to predict the future trajectory, which works as follows.

- First, we find the *matching length* of each trajectories in the historical trajectory table. Let l'_h denote the *matching length* of $\boldsymbol{G}_h$, we have

$$\sum_{l=1}^{l'_h} \|g_*^{t-l+1}, g_h^{\bar{l}_h - l+1}\| \le \epsilon, \tag{17}$$

  where $1 \le \bar{l}_h \le l_h$. Note that if (17) can not be satisfied, we set $l'_h = 0$.
- Next, we choose the candidate trajectory as

$$h^* = \arg\max_h \quad p_h \cdot \frac{l'_h \cdot \phi(l'_h - l_0)}{\sum_{h=1}^{H} l'_h \cdot \phi(l'_h - l_0)}, \tag{18}$$

  where l_0 is the minimum *matching length*, and

$$\phi(x) = \begin{cases} 1, & \text{if } x \ge 0, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

- Finally, the sub-trajectory $(g_{h^*}^{\bar{l}_{h^*}+1}, g_{h^*}^{\bar{l}_{h^*}+2}, \cdots, g_{h^*}^{\bar{l}_{h^*}+\Delta})$ of $\boldsymbol{G}_{h^*}$ is regarded as the predicted trajectory $(g_*^{t+1}, g_*^{t+2}, \cdots, g_*^{t+\Delta})$ in the next Δ time slots.

The TP algorithm is summarized in Algorithm 2.

2) SERVING NODE PREDICTION

In this part, we estimate the long-term service latency in the next Δ time slots by predicting the serving nodes, where the HMM technique is utilized.

For each vehicle, by replacing the GPS position in predicted trajectory with the corresponding area index, we can obtain a observation sequence O with length Δ. We take the serving node as the state of the vehicle, then we can use HMM to obtain the serving node sequence with length Δ. We introduce a 5-tuple $< \pi, \mathcal{N}, \mathcal{N}_a, A_t, B_t >$ to denote the HMM that describe the service migration process.

- **Initial State π:** π is N-dimension vector, which denotes the probability distribution of the initial state. Each element $\pi(n), n \in [1, N]$ is the probability of the vehicle request service from BS n.

- **State Set $\mathcal{N}$:** The state set $\mathcal{N}$ includes all BSs, i.e. $\mathcal{N} = \{1, 2, \cdots, N\}$.
- **Observation Set $\mathcal{N}_a$:** The state set $\mathcal{N}_a$ includes all possible area n_a vehicle may locate, i.e. $\mathcal{N}_a = \{1, 2, \cdots, N_a\}$.
- **State Transition Probability Matrix A_t:** A_t is a N-by-N probability matrix, where each element $A_t(i,j), i \in [1,N], j \in [1,N]$ is the transition probability that service is migrated from BS i to BS j in time slot t. Due to the mobility of SVs, the state transition probability varies between different time slots. More specifically, if the distance between BS i and BS j decreases in time slot t, $A_t(i,j)$ will increase, and vise versa. Thus, we update $A_t(i,j)$ as

$$A_t(i,j) = \frac{A_{t-1}(i,j)d(i,j,t-1)}{d(i,j,t)\sum_{j=1}^{N} A_{t-1}(i,j)\frac{d(i,j,t-1)}{d(i,j,t)}}. \quad (20)$$

- **Observation Probability Matrix B_t:** B_t is a N_a-by-N probability matrix, where each item $B_{n,t}(i,j), i \in [1,N], j \in [1,N]$ is the probability that service is served by serving node j when the vehicle is located in area i in time slot t. Similar to (20), $B_t(i,j)$ is update as

$$B_t(i,j) = \frac{B_{t-1}(i,j)\bar{d}(i,j,t-1)}{\bar{d}(i,j,t)\sum_{j=1}^{N} B_{t-1}(i,j)\frac{\bar{d}(i,j,t-1)}{\bar{d}(i,j,t)}}, \quad (21)$$

  where the $\bar{d}(i,j,t)$ denote the distance between area i and BS j in time slot t.

Let $\delta_{t'}(n)$ denote the maximum probability that the serving node is n in the time slot t', $t' \in [t, t+\Delta]$ given the observing sequence, which can be obtained as

$$\delta_{t'}(n) = \left\{ \max_{1 \le n' \le N} \delta_{t'-1}(n') A_{t'}(n', n) \right\} B_{t'}(O(t'), n). \quad (22)$$

Let $\psi_{t'}(n)$ denote the corresponding serving node in previous time slot, which can be obtained as

$$\psi_{t'}(n) = \arg \max_{1 \le n' \le N} [\delta_{t'-1}(n') A_{t'}(n', n)], \quad (23)$$

where $\psi_{t'}(n) = n'$ means the service migrates from BS n' to BS n in time slot t'.

We use a serving node prediction (SNP) algorithm utilizing the HMM model to predict the top-K serving nodes in the next Δ time slots, where the details are shown in Algorithm 3. The SNP algorithm works as follows. First, we initialize $\delta_1(n)$ and $\psi_1(n)$ (Lines 1-2). Second, we update $\boldsymbol{\delta}$ and $\boldsymbol{\psi}$ with the time slot t' increases (Lines 3-8). Next, to find the top-K probability serving node sequence, we record the kth highest probability $\delta_\Delta(n)$ in time slot $t+\Delta$ (Lines 10), and obtain the corresponding serving node sequence by recalling $\boldsymbol{\psi}'_t$ (Lines 11-14). As the value of $\delta_\Delta(n)$ has been recorded, we replace it with the sub-optimal value according to (22), where the $n' \neq Q_k(\Delta)$, and further update the value of $\psi_\Delta(Q_k(\Delta))$ (Lines 15).

Thus, we can find the serving node sequence with top-K probability by SNP algorithm. Take the observation sequence

Algorithm 3 Serving Node Prediction (SNP) Algorithm

Input: $\pi, O, A_0, B_0, \Delta, K$
Output: state sequence Q_K, δ^*

1: Initialize: $\delta_1(n) = \pi B_t(O(1), n), n \in [1, N]$;
2: Initialize: $\psi_1(n) = 0, n \in [1, N]$;
3: **for** $t' = 2$ to Δ **do**
4: **for** $n = 1$ to N **do**
5: Update $\delta_{t'}(n)$ according to (22);
6: Update $\psi_{t'}(n)$ according to (23);
7: **end for**
8: **end for**
9: **for** $k = 1$ to K **do**
10: Obtain $\delta^*(k) = \max_{1 \le n \le N} \delta_\Delta(n)$;
11: Obtain $Q_k(\Delta) = \arg \max_{1 \le n \le N} \psi_\Delta(n)$;
12: **for** $t' = \Delta - 1$ to 1 **do**
13: Obtain $Q_k(t') = \psi_{t'+1}(Q_k(t'+1))$;
14: **end for**
15: Update $\delta_\Delta(Q_k(\Delta))$ and $\psi_\Delta(Q_k(\Delta))$ according to (22),(23) with constraint $n' \neq \{Q_{\bar{k}}(\Delta - 1) | Q_{\bar{k}}(\Delta) = Q_k(\Delta), 1 \le \bar{k} \le k\}$;
16: **end for**

O and K serving node sequences as the trajectory and the serving node in next Δ time slot, the expected latency in future times slots $L^{\Delta}_{t,m}(v)$ is defined as

$$L^{\Delta}_{t,m}(v) = \sum_{k=1}^{K} \delta^*(k) \sum_{t'=1}^{\Delta} L_{t+t',m}(v) \quad (24)$$

C. INTERFERENCE INDICATOR

In this subsection, we propose a interference indicator based on the load in each BS. At the beginning of each slot, the following is performed:

- With the vehicles move, the load in each BS n is constantly changing. The storage load of each BS n in time slot t is

$$\rho_t(n) = \sum_{v=1}^{V} \sum_{m=1}^{M} \frac{\theta_{m,v} \mathcal{H}(b'_{t,m}(v), n)}{S_n}. \quad (25)$$

- BSs confirm the identities of its associated vehicles. The number of vehicles which migrate service m from BS n to another BS (compared to the beginning of the previous time slot) is

$$\pi^I_t(n,m) = \sum_{v=1}^{V} \left[\mathcal{H}(b'_{t,m}(v), n) - \mathcal{H}(b_{t,m}(v), n) \right]^+, \quad (26)$$

  The number of vehicles which migrate service m to BS n (compared to the beginning of the previous time slot) is

$$\pi^O_t(n,m) = \sum_{v=1}^{V} \left[\mathcal{H}(b_{t,m}(v), n) - \mathcal{H}(b'_{t,m}(v), n) \right]^+, \quad (27)$$

TABLE 3. Service parameters.

Type	D_m	λ_m	γ_m (KB)	Ω_m	$f_{m,v}$ (GHz)	$\theta_{m,v}$ (MB)
ES	0.1	3200	36	0.9	[1.6, 3.2]	[1, 3]
CR	0.1	4800	40	0.9	[2.4, 4.8]	[1, 3]
AR	0.5	4800	28	0.9	[2.4, 4.8]	[2, 3]
PA	0.1	1200	80	0.5	[0.6, 1.2]	[3, 5]
TC	1	1200	45	0.8	[0.6, 1.2]	[4, 5]
PL	0.5	4800	88	0.6	[2.4, 4.8]	[4, 6]
FD	0.5	3200	50	0.3	[1.6, 3.2]	[5, 10]

The number of vehicles which currently request service m from BS n is

$$\pi_t(n, m) = \sum_{v=1}^{V} \mathcal{H}(b_{t,m}(v), n). \quad (28)$$

- BSs update the probability that services m migrate to BS n by

$$\Pr\left\{I|n, m\right\} = \omega_1 \Pr\left\{I|n, m\right\} + \omega_2 \frac{\pi_t^I(n, m)}{\pi_t(n, m)}, \quad (29)$$

and update the probability that services m migrate from BS n to another BS by

$$\Pr\left\{O|n, m\right\} = \omega_1 \Pr\left\{O|n, m\right\} + \omega_2 \frac{\pi_t^O(n, m)}{\pi_t(n, m)}, \quad (30)$$

where ω_1 and ω_2 is the weight parameter ($\omega_1 + \omega_2 = 1$).
- To avoid the interference among vehicles, the services are willing to migrate from the BS with high load to the BS with low load. Therefore, the interference indicator is defined as

$$\Phi_{t,m}(v) = \Pr\left\{I|b_{t,m}(v), m\right\}\rho_t(b_{t,m}(v)) - \Pr\left\{O|b'_{t,m}(v), m\right\}\rho_t(b'_{t,m}(v)) \quad (31)$$

V. PERFORMANCE EVALUATION

A. SIMULATION SETUP

In this section, we evaluate the performance of FEE algorithm with simulations. We simulate an VFC system with 9 fixed BSs and 4 SVs deployed on a regular cellular network, which is divided into 625 geographical parts. The communication radius of BSs is 350m, while 150m for SVs. The vehicle trajectories are collected from the real-world Rome taxi traces obtained in 2014 [34]. Seven typical vehicular services, i.e., emergency stop (ES), collision risk (CR), accident report (AR), parking (PA), traffic control (TC), platoon(PL), and face detection (FD), are deployed on the serving nodes. The parameters of the services are listed in Table 3 [20], [35], [36]. We set W, P, and N_0 to 10 MHz, 0.5 W and 10^{-15}, respectively [11].

We compare FEE with four benchmarks.

- **Always-migrate scheme (AM)**: The service is always migrated to the BS that is nearest to the vehicle.
- **Never-migrate scheme (NM)**: The service is deployed in the original BS and never migrate to other BS.

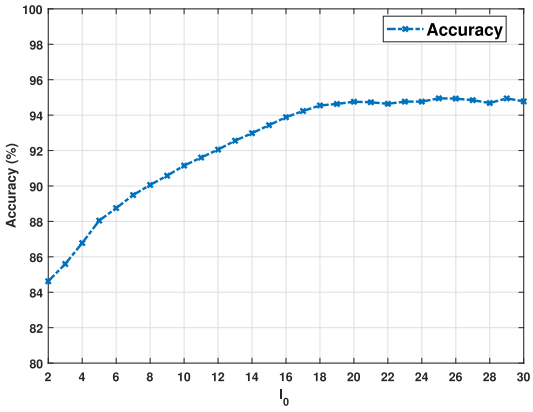


FIGURE 2. The accuracy of TP algorithm with different l_0 values.

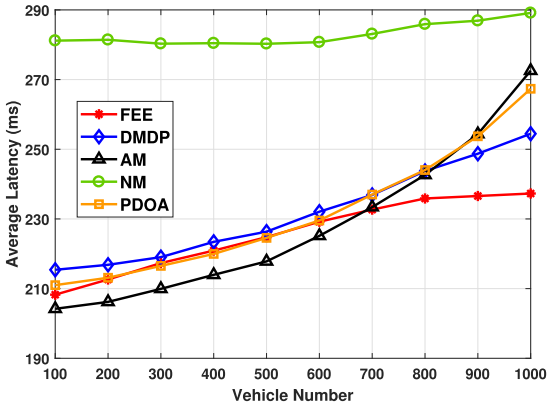
- **Dynamic Markov Decision Process (DMDP)** [24]: This is a single-user service migration algorithm, where the trajectory is predicted using MDP, and the optimal service migration decision is made to minimize the overall cost.
- **Partial dynamic optimization algorithm (PDOA)** [36]: Each vehicle selects a fraction of the services according to the priority queue, and only migrates these services to reduce the negative effect of trajectory prediction error.

B. THE ACCURACY OF TP ALGORITHM

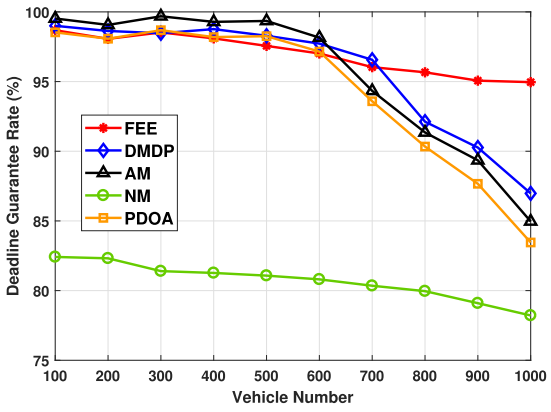
Fig. 2 illustrates the accuracy of TP algorithm with different l_0 values. The predicted trajectory is defined as accurate if it can be regarded as same with the real trajectory. It can be observed that the accuracy increases with the growth of l_0. This is because a larger l_0 value indicates the candidate trajectory chosen by TP algorithm has a higher similarity with the current vehicle trajectory. It is also shown that when l_0 is larger than 18, the accuracy becomes stable, which is around 95%.

C. IMPACT OF DIFFERENT VEHICLE NUMBER

The average latency with different number of vehicles are shown in Fig. 3(a). As can be seen from Fig. 3(a), when the number of vehicles increases from 100 to 1000, the average latency with NM is the highest. This is reasonable as the services never migrate to follow the vehicles. On the other hand, the lowest average latency is achieved by AM when the vehicle number below 600, since the services are always successfully migrated to the nearest BSs to follow the vehicles. However, as the number of vehicles exceeds 600, the average latency with AM increases exponentially. This is because the servers run out of resources with too many vehicles, and the interference among the vehicles begins to affect the performance. The average service latency with PDOA is quite close to that with AM, whose rising tendency is lower than AM due to PDOA only migrate a fraction of services. It is found that the average latency with FEE is is slightly larger than AM and DMDP when the number of vehicles is smaller than 600.



(a) The average latency



(b) The deadline guarantee rate

FIGURE 3. The average latency and deadline guarantee rate with different number of vehicles.

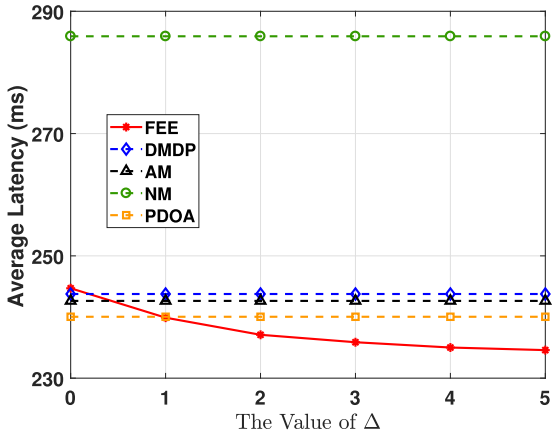


FIGURE 4. The impact of Δ on the average latency, where the number of vehicles is 800.

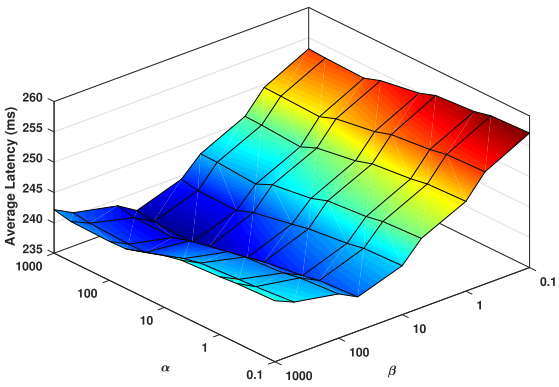


FIGURE 5. The impact of parameter α and β on the average latency, where the number of vehicle is 800.

However, as the number of vehicles exceeds 600, the lowest average latency can be achieved with FEE. This is because the interference among the vehicles is taken into account in the optimization problem formulated. This indicates that FEE is suitable for VFC networks.

The deadline guarantee rate of the four methods is depicted in Fig. 3(b). As an important performance metric, the deadline guarantee rate indicates the percentage of request that get response before its deadline. It can be observed that the deadline guarantee rate of NM is the lowest due to the long transmission path. When the vehicle number is less than 700, the deadline guarantee rate with AM, DMDP,PDOA and FEE are quite close to each other. However, the deadline guarantee rate with AM, DMDP and PDOA reduce rapidly when the user number exceeds 500. The deadline guarantee rate with FEE is the highest with large user numbers, e.g., it is approximately 95% with 1000 users. The results in Fig. 3(b) are consistent with that in Fig. 3(a).

D. IMPACT OF PARAMETERS Δ

Fig. 4 presents the average latency of FEE with different Δ values, where the number of vehicles is 800. It is surprising that the average latency of FEE decreases as Δ increases. Note $\Delta = 0$ means FEE only consider the latency in the current time slot. The average latency of FEE is higher than DMDP and AM when $\Delta = 0$, however, the average latency of FEE becomes the lowest among the 5 methods after Δ exceeds 1. This indicates by taking the long-term service latency in the following time slots into account, better service migration decisions can be made with FEE. It can also be observed from the figure that the average latency of FEE becomes stable if Δ exceeds 4. Therefore, a trade-off between the average service latency and the computation complexity can be achieved by selecting a proper Δ value.

E. IMPACT OF PARAMETERS α AND β

Fig. 5 shows the average latency of FEE with different α and β values, where the number of vehicles is 800. As can be seen from the figure, if we fix β, the average latency decreases as α increase from 0.1 to 100. This is because by taking the long-term service latency in the next Δ time slots into account helps to make a better migration decision. However, when α exceeds 100 which means we put more emphasis on the long-term service latency, the average latency slightly increases. Similarly, if we fix α, the average latency decreases as β increases from 0.1 to 50. This is because by taking the

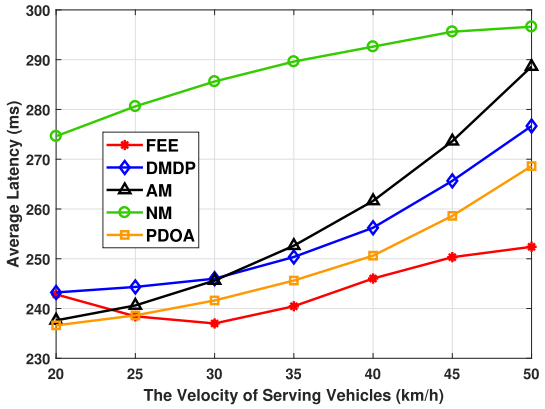


FIGURE 6. The average latency with velocity of SV, where the number of vehicle is 800.

interference among vehicles into consideration, better service migration decisions can be obtained. However, the average latency increases if β exceed 50, where more conservative service migration decisions might be made to avoid the potential interference. It is found that with $\alpha = 100$ and $\beta = 50$, the lowest average latency can be achieved.

F. IMPACT OF THE VELOCITY OF SVs

Fig. 6 shows the impact of the velocity of SVs on the average latency, where the number of vehicle is 800. There is a tendency that the average latency of the 5 methods increase as the velocity of SVs increase, as the high mobility of SVs brings great challenge to the service migration decisions. The average latency with NM is the highest, however, it is less sensitive to the mobility of SVs as the services never migrates. On the contrary, the average latency with AM and DMDP are much lower, but they are more sensitive to the mobility of SVs. The average latency of AM, DMDP and PDOA increases exponentially when the velocity of SVs exceeds 35 Km/h. The average latency with FEE decreases when the velocity of SVs is less than 30 km/h, since the low mobility of SVs reduces the service migration opportunities. Even the latency with FEE also increase with the velocity of SVs exceeds 30 km/h, the lowest average latency can be achieved with FEE.

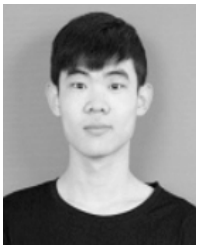
VI. CONCLUSION

In this paper, we investigate the service migration problem in VFC networks consists of fixed position BSs, moving SVs and vehicles. We formulate the service migration process as a MINP optimization problem to minimize the average latency, where the mobility of SVs and the interference among the vehicles are taken into account. An online algorithm, FEE, is proposed to solve the MINP problem based on HMM technique and a interference detection function. The simulations results based on the real world taxi trajectory in Rome demonstrated that the proposed solution can reduce the average latency by up to 7.6% and improve the deadline guarantee rate by up to 8.0%, as compared with the state-of-the-art solutions.

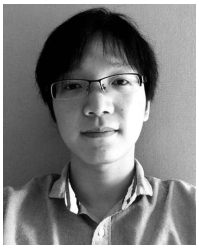
REFERENCES

[1] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 111–117, Jan. 2018.
[2] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.
[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Dec. 2017.
[4] T. Qiu, B. Li, W. Qu, E. Ahmed, and X. Wang, "TOSG: A topology optimization scheme with global small world for industrial heterogeneous Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3174–3184, Jun. 2019.
[5] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 6–9.
[6] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, Jun. 2018.
[7] T. Qiu, B. Li, X. Zhou, H. Song, I. Lee, and J. Lloret, "A novel shortcut addition algorithm with particle swarm for multisink Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3566–3577, May 2020.
[8] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[9] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 91–95, Nov. 2019.
[10] C. Li, Q. Luo, G. Mao, M. Sheng, and J. Li, "Vehicle-mounted base station for connected and autonomous vehicles: Opportunities and challenges," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 30–36, Aug. 2019.
[11] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.
[12] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.
[13] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1028–1042, Jun. 2019.
[14] A. Nadembega, A. S. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.
[15] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.
[16] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, "Optimized placement of scalable iot services in edge computing," in *IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Washington, DC, USA, Apr. 2019, pp. 189–197.
[17] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug. 2017.
[18] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-me cloud: When cloud services follow mobile users," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 369–382, Apr. 2019.
[19] A. Sadilek and J. Krumm, "Far out: Predicting long-term human mobility," in *Proc. Int. Conf. Artif. Intell. (AAAI)*, Toronto, ON, Canada, Jul. 2012, pp. 814–820.
[20] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.
[21] M. Satyanarayanan, "Edge computing for situational awareness," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Jun. 2017, pp. 1–6.
[22] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

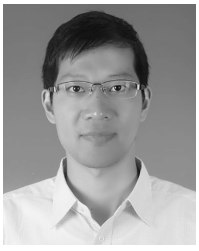
[23] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.
[24] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on Markov decision process," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019.
[25] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[26] L. Yan, H. Shen, and K. Chen, "MobiT: Distributed and congestion-resilient trajectory-based routing for vehicular delay tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1078–1091, Jun. 2018.
[27] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 49–55, May 2019.
[28] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Migrating running applications across mobile edge clouds: Poster," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, New York City, NY, USA, Oct. 2016, pp. 435–436.
[29] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3100–3112, Apr. 2019.
[30] C. Ferguson and P.-S. Seow, "Accounting information systems research over the past decade: Past and future trends," *Accounting Finance*, vol. 51, no. 1, pp. 235–251, Mar. 2011.
[31] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for follow me cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 1350–1354.
[32] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.
[33] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[34] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. (Jul. 17, 2014). *CRAWDAD Dataset Roma/Taxi*. [Online]. Available: https://crawdad.org/roma/taxi/20140717
[35] W. Nasrin and J. Xie, "SharedMEC: Sharing clouds to support user mobility in mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.
[36] X. Yu, M. Guan, M. Liao, and X. Fan, "Pre-migration of vehicle to network services based on priority in mobile edge computing," *IEEE Access*, vol. 7, pp. 3722–3730, 2019.



SHUXIN GE received the bachelor's degree from the School of Mechanical Engineering, Hebei University of Technology (HEBUT), Dalian, China, in 2018. He is currently pursuing the Ph.D. degree in computer science and technology with the College of Intelligence and Computing, Tianjin University (TJU), Tianjin, China. He is also a member of the Tianjin Advanced Networking Key Laboratory (TANK Lab). His researches focus on the mobile edge computing, the Internet of Things, vehicular fog computing, and intelligent connected vehicle.



MENG CHENG (Member, IEEE) received the B.Eng. degree in telecommunication engineering from the Anhui University of Technology, Anhui, China, in 2009, the M.Sc. degree (Hons.) in wireless communications from the University of Southampton, Southampton, U.K., in 2010, and the Ph.D. degree in information science from the Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2014. He has served as a 5G Research Engineer with the Shanghai Research Center, Huawei Technologies Company, Ltd., from 2014 to 2017. After that, he returned to JAIST as a Postdoc Researcher. His research interests are network information theory, non-orthogonal multiple access (NOMA), iterative coding/decoding, and wireless geolocation techniques.



XIAOBO ZHOU (Senior Member, IEEE) received the B.Sc. degree in electronic information science and technology from the University of Science and Technology of China (USTC), Hefei, China, the M.E. degree in computer application technology from Graduate University of Chinese Academy of Science (GUCAS), Beijing, China, and the Ph.D. degree in information science from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2007, 2010, and 2013, respectively. From April 2014 to March 2015, he was a Researcher with the Department of Communications Engineering, University of Oulu, Oulu, Finland. He is currently an Associate Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include cooperative wireless communications, wireless networks, data center networks, vehicular networks, and mobile edge computing.