sensors

MDPI

Article

A Two-Stage Service Migration Algorithm in Parked Vehicle Edge Computing for Internet of Things

Shuxin Ge [1], Meng Cheng [2], Xin He [3,*] and Xiaobo Zhou [1]

[1] Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China; cecilge@tju.edu.cn (S.G.); xiaobo.zhou@tju.edu.cn (X.Z.)
[2] School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi, Ishikawa 923-1292, Japan; m-cheng@jaist.ac.jp
[3] School of Computerand Information, Anhui Normal University, Wuhu 241001, China
* Correspondence: xin.he@ahnu.edu.cn

Received: 18 March 2020; Accepted: 12 May 2020; Published: 14 May 2020

Abstract: Parked vehicle edge computing (PVEC) utilizes both idle resources in parked vehicles (PVs) and roadside units (RSUs) as service providers (SPs) to improve the performance of vehicular internet of things (IoT). However, it is difficult to make optimal service migration decisions in PVEC networks due to the uncertain parking duration and resources heterogeneity of PVs. In this paper, we formulate the service migration of all the vehicles as an optimization problem with the objective of minimizing the average latency. We propose a two-stage service migration algorithm for PVEC networks, which divides the original problem into the service migration between SPs and the serving PV selection in parking lots. The service migration between SPs is transformed to an online problem based on Lyapunov optimization, where the expected parking duration of PVs is utilized. A modified Hungarian algorithm is proposed to select the PVs for migration. A series of simulation experiments based on the real-world vehicle traces are conducted to verify the superior performance of the proposed two-stage service migration (SEA) algorithm as compared with the state-of- art solutions.

Keywords: internet of things; parking duration; service provider; Lyapuunov optimization; Hungarian algorithm

1. Introduction

The development of the automotive industry and the popularity of private cars have led to various issues such as traffic congestion, accidents, air pollution, etc. [1,2]. To ensure the safety and quality of experience (QoE), many novel vehicular Internet of Things (IoT) services are designed to control vehicle behavior by collecting the data from on-board sensors, for example, surrounding vehicle perception and collision risk [3]. In order to cope with great demands for these services, roadside units (RSUs) that are generally located in different geographic regions can be deployed on an edge server to act as a service provider (SP) [4]. However, the RSUs are always overloaded with the rapidly increasing number of vehicles and various service requests [5]. As such, more available resources are on demand [6]. It has been observed that the idle resources in parked vehicles (PVs) can be explored to efficiently provide services to other vehicles [7]. Therefore, a new computing paradigm, called parked vehicle edge computing (PVEC), has emerged to improve the vehicular IoT performance [8].

Generally speaking, private cars usually spend most of the time in parking lots with their resources in idle states, where the average parking duration is approximately seven hours [9]. PVs may consume energy for providing services to other vehicles, a discount in parking fees can be an incentive to a PV [10]. PVEC can take the parking lots as SPs by managing and integrating the rich resources such as storage resources and computing resources of the PVs [11]. The access point of the parking lot determines which PV is suitable to provide services to other vehicles. Through the PVEC, a vehicle can request for service from the parking lot to improve QoE. As such, the load of RSUs will be reduced, which means RSUs can provide higher QoE for vehicles [12]. However, it is a complicated process to manage the vehicle services with involvement of the parking [13,14]. For example, when the PV providing the service to other vehicles leaves the parking lot or refuses to continue providing the service, the ongoing service will be interrupted and delayed [15]. It is the case that the PVs with longer parking duration will help the stability of QoE. Moreover, different available resources in PVs, e.g., energy, also effect the network performance. When the resources can not support the service requests, or the PVs enter and leave the parking lots, the service needs to be migrated into the optimal PV so that the service continuity can be maintained [16]. With the migration, the service resumes exactly where it stopped at the target serving node. Furthermore, the service migration is also triggered by the vehicle mobility, which requires that the service is migrated to the nearest SP, for instance, the next immediate vehicle [17].

A proper service migration strategy should make a trade-off between the service latency and energy consumption of PVs, which encounters two challenges [18,19]. One challenge is that the parking duration of PVs greatly affects the overall performance of the service migration strategy [4]. As the leaving time of a PV cannot be known in advance, the service which is migrated to the leaving PV will result in an invalid migration or frequent migrations. Another challenge is that different long-term energy constraints in PVs are coupled with the service migration decision over the time, where the decisions have to be made without future information [20]. If a myopic migration decision achieves an optimal performance in terms of latency with huge energy consumption of a PV, the PV would suffer from energy shortage in the near future [21].

To address these challenges, this paper investigates the dynamic service migration in PVEC. A novel two-stage service migration (SEA) algorithm is proposed to solve the formulated problem. The main contributions of this paper are summarized as follows:

- We investigate the service migration problem in PVEC, and formulate the service migration process as a mixed integer nonlinear programming (MINP) to minimize the average service latency, where the QoE constraints and the available energy of PVs are taken into account. To our best knowledge, this is the first work that studies service migration in PVFC networks.
- We propose an efficient SEA algorithm to solve the formulated problem, where the original problem is divided into service migration between SPs and the selection of PVs for migration in parking lots. Utilizing Lyapunov optimization, we perform service migration decision between SPs in an online fashion without requiring future information. Meanwhile, a modified Hungarian algorithm is proposed to effectively tackle the difficult in selection of PVs with different available energy.
- The effectiveness of the proposed SEA algorithm is verified by simulation based on the real-world vehicle traces. Compared with three benchmarks, the simulation results demonstrate SEA algorithm has the superior performance in terms of average latency, utilization of PVs and energy consumption, especially with large total number of vehicles.

The rest of this paper is organized as follows. Section 2 reviews the relevant existing literature. The system model considered in the paper is presented in Section 3. Section 4 proposes SEA algorithm and conducts the complexity analysis. Section 5 discusses the performance of SEA algorithm. Section 6 concludes the paper.

2. Related Works

PVEC is an extension of edge network to relieve the high load pressure of RSUs in IoT. According to [22,23], the total number of PVs tend to remain stable with an average parking duration of around 6 h in North America. Many existing works design a incentive mechanism to encourage PVs to share its idle resources [18]. Malandrino et al. [24] leverages optimization model, where PVs serve as relay nodes, to maximize both freshness of the content that download retrieve and the efficiency in the utilization of radio resources. Balen et al. [25] utilize PVs that are near to passing vehicles to distribute messages. The objective of these methods is minimizing the download delay by choosing optimal PVs. Tonguz et al. [26] design a local rule to cache services in PVs, which are selected by biologically inspired self-organizing network approach, and thus achieve low latency of download updates. Li et al. [4] classify PVs into different types according to their parking duration. They design a contract-based incentive mechanism to motivate PVs with long parking duration to contribute their idle resources, and thus maximize the utility of the SP. However, these methods did not fully utilize the resources of PVs.

Recently, the majority of the literature on PVEC have focused on allocating the resources of PVs [7,27]. Gu et al. [13] propose a two-tier data center architecture that leverages the excessive storage resources in parking lots. They design three management policies for data center, i.e., non-replication, simple replication and network coding based replication, which aim to minimize the total communication cost in a closed form. Zhang et al. [28] study task assignment problem in PVs based on Stackelberg game. Through the game between PVs and tasks, it decide the serving PV for each task, and thus minimize the overall cost of a task publisher and subjective dissatisfaction caused by un-offloaded workloads. Huang et al. [29] minimize the service latency by introducing a machine learning framework, which allocates the heterogeneous resources of PVs based on the requirements of the services. Ouyang et al. [30] introduce a scheme based on Lyapunov optimization to minimize the system costs over time in compliance with the time-varying resources of SPs. Zhu et al. [31] make a trade-off between the average service latency and the overall quality loss by designing an event-triggered dynamic allocation scheme based on particle swarm optimization. However, these methods did not take the mobility of vehicles into account. As the vehicle moves, the SP of the vehicle changes constantly, therefore the latency performance obtained by these resource allocation schemes can not be maintained.

To tackle the mobility of vehicles, service migration has been extensively studied in edge networks to reduce the service latency [32,33]. Ksentini et al. [34] assume the user makes the rectilinear translation, and build a one-dimension (1D) MDP model to predict the trajectory. According to the predicted trajectory, the users determine the target serving node of the service migration. Nadembeg et al. [35] select optimal serving node by predicting the throughput in the target node. The service request is assumed to be split into several parts, and the service can be served by many servers at the same time. To be closer to the real scenes, Wang et al. [33] build a 2D Markov Decision Process (MDP) model to predict the user movement behavior in cellular networks, and further make the service migration to achieve the lowest system cost. Yu et al. [36] study a single-user service migration algorithm to select a fraction of the services according to the priority queue, and only migrates these services to reduce the negative effect of trajectory prediction error. Although these service migration strategies have a good performance in edge networks, it can not be directly applied to PVEC networks due to the uncertain parking duration and resources heterogeneity of PVs.

3. System Model

3.1. System Overview

In this paper, we consider a PVEC networks consisting V vehicles and N SPs, as shown in Figure 1. As vehicle A moves along the main road, it connects to SP 1, SP 2 and SP 3 sequentially. Initially, a service is deployed on SP 1 which can be accessed by vehicle A directly. As vehicle A approaches SP 2 and SP 3, a proper service migration decision will be made to decide whether to migrate the service by considering both the migration cost and response delay. As shown in the Figure 1, the service is migrated to SP 2 when vehicle A is near SP 2, and is not migrated to SP 3 when vehicle A is near SP 3. In the latter case, vehicle A gets access to the service by request routing. We divide the total time into T time slots, where each time slot lasts for π seconds. In time slot

$t \in [0, T-1]$, the vehicles move among the coverage regions. In general, a vehicle connects to the SP with the largest indicator of received signal strength. As the vehicle moves, each vehicle $v \in [1, V]$ generates its service request randomly to SP $n \in [1, N]$. In each time slot, the operator decides whether and where to migrate the services for the vehicles.

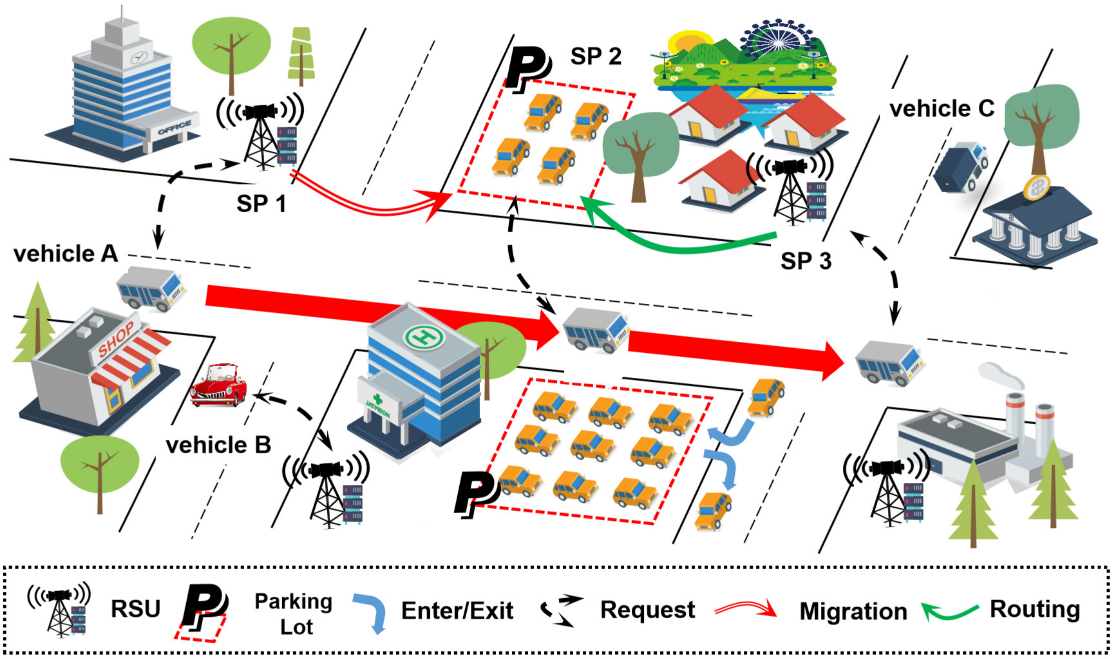

Figure 1. System overview.

By deploying the vehicular services, SPs can provide services to the vehicles within their radio ranges. When the corresponding service is deployed by the SP which the vehicle is connected with, the service request can be served locally with and minimized latency. These SPs are connected to each other via Ethernet so that any vehicle is able to access all the services along the request routing once it is in the coverage region of a SP.

Assume the SPs contain I RSUs and J parking lots (I + J = N), where SPs $i \in [1, I]$ are RSUs with an edge server and other SPs $j \in [I+1, I+J]$ are the parking lots equipped with a access point which is used to communicate with the PVs. The parking lots use the idle resources (e.g., storage capacity) of PVs to serve the vehicles. Let $\mathbb{K}_t^j$ denote the set of PVs in parking lot j, which is updated by

$$\mathbb{K}_{t+1}^j = \mathbb{K}_t^j \cup \mathbb{K}_{t,in}^j - \mathbb{K}_{t,out}^j, \tag{1}$$

where $\mathbb{K}_{t,in}^j$ and $\mathbb{K}_{t,out}^j$ are the set of the PVs which enter the parking lot j in time slot t, and the set of the PVs which leave the parking lot j in time slot t, respectively. Thus, we have

$$\mathbb{K}_{t,in}^j \cap \mathbb{K}_t^j = \varnothing, \quad \mathbb{K}_{t,out}^j \subseteq \mathbb{K}_t^j. \tag{2}$$

Each PV $k \in \mathbb{K}_t^j$ provides its idle resources to get a discount for its parking fees.

There are M different vehicular services that the vehicles may request for. We use a 5-tuple $< \lambda_m, \gamma_m, D_m, f_{m,v}, \theta_{m,v} >$ to describe service m, where the five coordinates denote the request data size, the computation intensity (i.e., CPU cycles/bit), the response deadline, computing capacity and storage requirement of service m for vehicle v, respectively. What a vehicle requests for multiple services is the same as that multiple vehicles request for a service. We assume each vehicle v continuously requests for one service $m(v) \in [1, M]$ in this paper. Table 1 summarizes the key parameter notations in our paper.

Table 1. Definitions of notations.

Notation	Definition
$\mathbb{K}_{t+1}^{j}$	the set of PVs in parking lot j
$c_t v$	the connected SP of the vehicle v in time slot t
$\boldsymbol{b}_t(v)$	the SP of the request of vehicles v in time slot t
$\boldsymbol{b}_t'(v)$	the migration decision of SP for vehicle v in time slot t
$\boldsymbol{p}_t(v)$	the serving PV of the request of vehicles v in time slot t
$\boldsymbol{p}_t'(v)$	the migration decision of PV for vehicle v in time slot t
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 SP i and SP j in time slot t
$L(v,t)$	the total latency of vehicle v in time slot t
P	the communication power
$E(k,j,t)$	the total energy consumption of PV k in parking lot j at time slot t
$\tau(k), \bar{\tau}(k)$	the enter time and leaving time of PV k

3.2. Service Placement

The vehicle states are composed of its connected SP information, its serving node information in each time slot t, and the migration decision made by the networks operator.

For each vehicle v, we use $c_v(t)$ to denote the number of its connected SP. Thus, we have

$$c_t(v) \in \{1, 2, \cdots, N\}, \forall t, v. \tag{3}$$

Let $b_t(v)$ denote the serving node of vehicle v for service $m(v)$ in time slot t, where $b_t(v) = n$ means vehicle v requests for service from SP n in time slot t. In addition, we use $b_t'(v)$ to represent the migration decision, where the $b_t'(v) = n$ means the service $m(v)$ of vehicle v migrates to SP n in time slot t. Thus, the SP in the next time slot is updated by

$$b_{t+1}(v) = b_t'(v). \tag{4}$$

Similar to (3), we have

$$\begin{cases} b_t(v) \in \{1, \cdots, N\}, \\ b_t'(v) \in \{1, \cdots, N\}, \end{cases} \forall t, v. \tag{5}$$

Moreover, the service request to SP j will be further allocated to the PVs. We define a $p_t(v)$, where $p_t(v) = k$ represents PV k in parking lot j which provides the service to vehicle v in time slot t. Meanwhile, we use $p_t'(v)$ to represent the migration decision, where $p_t'(v) = k$ represents the service of vehicle v is migrated to PV k in parking lot j at time slot t. The PV for migration in the next time slot is updated by

$$p_{t+1}(v) = p_t'(v). \tag{6}$$

Thus, we have

$$\begin{cases} p_t'(v) = 0, & b_v'(t) \leq I, \\ p_t'(v) \in \mathbb{K}_t^j, & \text{otherwise}. \end{cases} \tag{7}$$

3.3. Service Latency

The service latency is composed of transmission latency, computing latency and migration latency. The transmission latency refers to the time it takes to send a request to the service node. The mobile edge network state in time slot t is described by an N-by-N matrix denoted by $\boldsymbol{R}_t$. The (a, b)th $(a \in [1, N], b \in [1, N])$ element $R_t(a, b)$ denotes the available channel resources between SP a and SP b in time slot t. When $a = b$, element $R_t(a, b)$ is the positive infinity. Otherwise, it is randomly distributed in (η_{min}, η_{max}).

Meanwhile, we use $\bar{R}_t(a, b), (a \in (I, N], b \in \mathbb{K}_t^j)$ to denote the transmission rate between the access point in parking lot a and PV b in time slot t, which is randomly distributed in $(\eta_{min}, \bar{\eta}_{max})$. As the distance from the access point and the PV is short, $\bar{\eta}_{max}$ is much higher than η_{max}.

Moreover, since the transmission latency between the vehicle and SP is hard to calculate precisely due to the movement of the vehicles, we use a constant C to approximate it. Assuming vehicle v generates a request to service $m(v)$ at the beginning of time slot t, the request transmission latency includes the transmission latency between vehicle v and SP $b_t(v)$ in time slot t

$$l_1^T(v, t) = \frac{\lambda_{m(v)}}{R_t(c_t(v), b_t(v))} + C, \tag{8}$$

and the transmission latency between SP $b_t(v)$ and PV $p_t(v)$ in time slot t

$$l_2^T(v, t) = \begin{cases} \frac{\lambda_{m(v)}}{\bar{R}_t(b_t(v), p_t(v))}, & b_t(v) > I. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The total transmission latency is

$$L_T(v, t) = l_1^T(v, t) + l_2^T(v, t). \tag{10}$$

The computation latency of vehicle v for service $m(v)$ is

$$L_C(v, t) = \frac{\lambda_{m(v)} \gamma_{m(v)}}{f_{m(v), v}}. \tag{11}$$

Note that service $m(v)$ may be migrated from $p_t(v)$ in SP $b_t(v)$ to $p'_t(v)$ in SP $b'_t(v)$ in time slot t. The migration latency includes the migration latency between SP $b_t(v)$ and SP $b'_t(v)$

$$l_1^M(v, t) = \frac{\theta_{m(v), v}}{R_t(b_t(v), b'_t(v))}, \tag{12}$$

the migration latency between SP $b_t(v)$ and PV $p_t(v)$

$$l_2^M(v, t) = \begin{cases} \frac{\theta_{m(v), v}}{\bar{R}_t(b_t(v), p_t(v))}, & b_t(v) > I \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

and the migration latency between SP $b'_t(v)$ and PV $p'_t(v)$

$$l_3^M(v, t) = \begin{cases} \frac{\theta_{m(v), v}}{\bar{R}_t(b'_t(v), p'_t(v))}, & b'_t(v) > I \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

The total migration latency is

$$L_M(v, t) = \begin{cases} 0, & b_t(v) = b'_t(v), p_t(v) = p'_t(v) \\ l_1^M(v, t) + l_2^M(v, t) + l_3^M(v, t), \text{otherwise.} \end{cases} \tag{15}$$

Thus, the service latency for vehicle v in time slot t is

$$L(v, t) = L_T(v, t) + L_C(v, t) + L_M(v, t). \tag{16}$$

3.4. Energy Consumption

When entering the parking lot, PV k messages its available energy $\hat{e}(k, j)$ to the access point in parking lot j. The energy consumption in PV must satisfy this available energy constraint. The transmission energy consumption of PV k for vehicle v in parking lot j at time slot t is

$$e_T(v, k, j, t) = P\pi(b_t(v), j)\pi(p_t(v), k)l_2^T(v, t), \quad k \in \mathbb{K}_t^j, \tag{17}$$

where

$$\pi(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

where P is the communication power. The computing energy consumption of PV k for vehicle v in parking lot at time slot t is

$$e_C(v, k, j, t) = \pi(b_t(v), j)\pi(p_t(v), k)\kappa f_{m(v),v}^3 L_C(v, t) \quad k \in \mathbb{K}_t^j, \tag{19}$$

where κ is the unit energy consumption per CPU cycle. The migration energy consumption of PV k for vehicle v in parking lot j at time slot t is

$$e_M(v, k, j, t) = P\Big[\pi(b_t(v), j)\pi(p_t(v), k)l_2^M(v, t) + \pi(b_t'(v), j)\pi(p_t'(v), k)l_3^M(v, t)\Big], \quad k \in \mathbb{K}_t^j. \tag{20}$$

The total energy consumption in PV k for vehicle v in parking lot j at time slot t is

$$e(v, k, j, t) = e_T(v, k, j, t) + e_C(v, k, j, t) + e_M(v, k, j, t), \quad k \in \mathbb{K}_t^j. \tag{21}$$

Thus, the available energy of PV k in parking lot j at the beginning of time slot t is

$$E(k, j, t) = \begin{cases} \hat{e}(k, j), & t = \tau(k),\ k \in \mathbb{K}_t^j \\ E(k, j, t-1) - \sum\limits_{v=1}^{V} e(v, k, j, t-1), & t \in (\tau(k),\ \bar{\tau}(k)],\ k \in \mathbb{K}_t^j \end{cases}, \tag{22}$$

where $\tau(k)$ and $\bar{\tau}(k)$ are the enter time of PV k and the corresponding leaving time, respectively.

4. Two-Stage Service Migration Algorithm

4.1. Problem Formulation

Our goal is to minimize the long-term average latency with the energy consumption constraint in PVs by selecting the optimal migration strategy $b_v'(t)$ and $p_v'(t)$, which can be expressed as

$$\begin{aligned} \mathbf{P1}: \quad & \min_{b_v', p_v'} \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{v=1}^{V} L(v, t) \\ s.t. \quad & L(v, t) \leq D_{m(v)},\ \forall v, t, && (23) \\ & \sum_{v=1}^{V} e(v, k, j, t) \leq E(k, j, t),\ \forall k, j, t, && (24) \\ & (3), (5), (7). \end{aligned}$$

Constraint (23) ensures the service request can be completed before its response deadline. Constraint (24) denotes the energy of PV k in parking lot j can support the service request. The lack of future information is the difficulty in deriving the optimal solution to the above problem as solving **P1** optimally requires the complete information. Moreover, **P1** is an MINP problem that is proven to be NP-hard.

To solve **P1**, first we decouple **P1** into two sub-problems, service migration between SPs problem and serving PV selection problem. Then we propose a two-stage algorithm, SEA, to solve these two sub-problems. In the first stage, we solve the service migration between SPs problem, i.e., deciding the serving SP for each vehicle. We utilize the Lyapunov optimization to convert this problem to per-slot optimization problem, which can be solved in an online style requiring only the current slot information. In the second stage, we solve the serving PV selection problem, i.e., deciding the serving PVs for vehicles whose serving SP is parking lot. We propose a modified Hungarian algorithm to choose the suitable PVs for vehicles, which fully utilizes the resources of all the PVs.

4.2. Problem Decomposition

The difference between RSUs and parking lots is that the parking lot needs to allocate its received service request from the vehicles to PVs with the energy constraint. Thus, we first consider parking lot j as a special RSU, which produces a extra latency during the request process. Replacing the transmission rate $\bar{R}(b_t(v), p_t(v))$ in (9) and (13), and the transmission rate $\bar{R}(b'_t(v), p'_t(v))$ in (14) with a constant $\frac{\bar{\eta}_{max}}{2}$, we can obtain a new total latency $\bar{L}_1(v,t)$. Meanwhile, the available energy in parking lot j at the beginning of time slot t is

$$\bar{E}(j,t) = \sum_{k \in \mathcal{K}_t^j} E(k,j,t). \tag{25}$$

Note that this energy only can be used before the PVs leave the parking lots. The first stage can be described by

$$\begin{aligned} \mathbf{P2}: \quad & \min_{\boldsymbol{b}'_v} \sum_{t=0}^{T-1} \sum_{v=1}^{V} \bar{L}_1(v,t) \\ s.t. \quad & \bar{L}_1(v,t) \le D_{m(v)}, \forall v, && (26) \\ & \sum_{k \in \mathcal{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) \le \bar{E}(j,t), && (27) \\ & (3), (5). \end{aligned}$$

After obtaining the solution of **P2**, solving the original problem **P1** only needs to select the serving PV for the received migrated service in each parking lot j at time slot t. The total service latency in parking lot j is

$$\bar{L}_2^j(v,t) = \pi(b_t(v), j)\left[l_2^T(v,t) + l_C(v,t) + l_2^M(v,t)\right] + \pi(b'_t(v), j) l_3^M(v,t). \tag{28}$$

Let $\mathcal{V}_t^j$ denote the set of the vehicles, which migrate the service to parking lot j at time slot t, i.e., $b'_t(v) = j$. In this way, the second stage problem for parking lot j in time slot t can be represented by

$$\begin{aligned} \mathbf{P3}: \quad & \min_{\boldsymbol{p}'_v,\, v \in \mathcal{V}_t^j} \sum_{v \in \mathcal{V}_t^j} \bar{L}_2^j(v,t) \\ s.t. \quad & (5), (24). \end{aligned}$$

The detail of the SEA algorithm is shown in Algorithm 1. In each time slot t, we obtain the migration decision b'_v, i.e., the SP of each vehicle by solving the first stage problem (Line 2). Based on this, we obtain the migration decision $p'_v(t)$, i.e., actual serving PV of the vehicle which migrates the service to the PV in parking lot j by solving the second stage problem (Lines 3–5).

Algorithm 1 Two-stage service migration (SEA) algorithm.

```
Input: : $\boldsymbol{c}_v(t), \boldsymbol{b}_v, \boldsymbol{b}'_v$;
Output: : Service migration decision $\boldsymbol{b}'_v, \boldsymbol{p}'_v$;
for $t = 0$ to $T - 1$ do
    Obtain $b'_v(t)$ by solving P2 according to Algorithm 2;
    for $j = I + 1$ to $N$ do
        Obtain $p'_v(t),\ v \in \mathcal{V}_t^j$ by solving P3 according to Algorithm 3;
    end for
end for
```

4.3. Service Migration between SPs

In this subsection, we transform the **P2** to per-slot optimization problems based on Lyapunov optimization.

4.3.1. Expected Parking Duration

The energy consumption under a certain migration decision greatly influences the available energy in the future, e.g., the parking lot needs to store the energy to attend frequent requests. We predict the average parking duration to ensure the energy consumption in each time slot is reasonable, in order to avoid rapidly running out of energy and lack of energy in the future, and also avoid using too little energy and compromising, the current latency performance.

Firstly, we identify the type of the PVs based on the parking duration. The parking duration is sorted in an ascending order and classified into $\bar{N}$ types $\{\delta_1, \delta_2, \cdots, \delta_{\bar{N}}\}$, given by

$$\delta_1 < \cdots < \delta_{\bar{n}} < \cdots < \delta_{\bar{N}},\ \bar{n} \in \{1, 2, \cdot, \bar{N}\}. \tag{29}$$

However, the information asymmetry between the SP and the PVs causes the SP can not realize the type of PVs. We generate the probability distribution of PV types through the historical data statistics. The probability that a PV belongs to type-$\bar{n}$ in parking lot j is denoted by $\Pr(\delta_{\bar{n}})$, and

$$\sum_{\bar{n}=1}^{\bar{N}} \Pr(j, \delta_{\bar{n}}) = 1. \tag{30}$$

The expected parking duration of PVs in parking lot j is

$$\mathbb{E}\{\tau(j)\} = \sum_{\bar{n}=1}^{\bar{N}} \Pr(j, \delta_{\bar{n}})\delta_{\bar{n}}. \tag{31}$$

4.3.2. Algorithm for the Service Migration between SPs

As discussed in Section 4.1, to solve **P2** online, we need to convert **P2** to per-slot optimization problems requiring only current slot information. Thus, according to the expected parking duration of PVs, we leverage the Lyapunov optimization technique and build a virtual queue to guide the migration decisions with the energy constraint.

$$Q_j(t+1) = \left[Q_j(t) + \sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v, k, j, t) - \frac{\bar{E}(j, t)}{\mathbb{E}\{\tau(j)\}} \right]^+, \tag{32}$$

where $Q_j(t)$ is the queue backlog for parking lot j in time slot t, indicating the deviation of the current energy consumption from the energy constraint. The Lyapunov function is defined as $\mathbb{L}(\xi(t))$, representing the "congestion level" in energy deficit queue, which is

$$\mathbb{L}(\xi(t)) \triangleq \frac{1}{2} \sum_{j=I+1}^{N} \bar{Q}_j^2(t). \tag{33}$$

A small value of $\mathbb{L}(\xi(t))$ implies that the queue backlog is small, i.e., the virtual queue has strong stability. To keep the energy deficit queue stable, i.e., to enforce the energy consumption constraints by persistently pushing the Lyapunov function towards a lower value, let $\triangle(t)$ denote the one-slot Lyapunov drift, which is $\triangle(t) = L(\xi(t+1)) - L(\xi(t))$, and we have

$$\begin{aligned}
\triangle(t) &= \frac{1}{2} \sum_{j=I+1}^{N} \left[Q_j^2(t+1) - Q_j^2(t) \right] \\
&\le \frac{1}{2} \sum_{j=I+1}^{N} \Big[\sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) - \frac{\bar{E}(j,t)}{\mathbb{E}\{\tau(j)\}} \Big]^2 + \sum_{j=I+1}^{N} Q_j(t) \Big[\sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) - \frac{\bar{E}(j,t)}{\mathbb{E}\{\tau(j)\}} \Big] \\
&\le JU + \sum_{j=I+1}^{N} Q_j(t) \Big[\sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) - \frac{\bar{E}(j,t)}{\mathbb{E}\{\tau(j)\}} \Big],
\end{aligned}$$

where

$$U \triangleq \sup \frac{1}{2} \sum_{j=I+1}^{N} \Big[\sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) - \frac{\bar{E}(j,t)}{\mathbb{E}\{\tau(j)\}} \Big]^2. \tag{34}$$

Under the Lyapunov optimization framework, the underlying objective of our optimal control decision is to minimize a supremum bound on the following drift-plus-cost expression in each time slot.

$$\triangle(t) + \mu \sum_{v=1}^{V} \bar{L}(v,t) \le JU + \sum_{j=I+1}^{N} Q_j(t) \sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) + \mu \sum_{v=1}^{V} \bar{L}(v,t),$$

where the parameter $\mu > 0$ can make the trade-off between the service latency and the energy consumption of SPs.

Under the Lyapunov optimization framework, the underlying migration decision of **P2** is made by solving the optimization problem **P4**

$$\begin{aligned}
\mathbf{P4}: \quad & \min_{\boldsymbol{b}'_v} \mu \sum_{v=1}^{V} \bar{L}_1(v,t) - \sum_{j=I+1}^{N} Q_j(t) \sum_{k \in \mathbb{K}_t^j} e(k,j,t) \\
s.t. \quad & (26), (3), (5).
\end{aligned}$$

Note that **P4** only requires the current available information. By considering the additional term $\sum_{j=I+1}^{N} Q_j(t) \sum_{k \in \mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t)$, the energy deficit of SPs during the current slot is taken into account. A high value of $\bar{E}(j,t)$ means it is critical to minimize the energy deficit. The detail of solving the service migration between SPs is shown in Algorithm 2. We first initialize the PV set as a null set (Line 1). At the beginning of the time slot t, the parking lots update its PVs set (Line 3). After this step, we can obtain $\boldsymbol{b}'_v$ by solving **P4** in an iterative manner (Line 4). Next, we calculate the available energy in the next time slot, and update the energy queue (Lines 5, 6). Finally, the solution in the time slot t of **P4** becomes the initial state in the next time slot $t+1$ (Line 7).

Algorithm 2 Service migration between the service providers (SPs) algorithm.

```
Input: :$Q_j(t) \leftarrow 0, \bar{E}(j,0) \leftarrow 0, \boldsymbol{c}_v(t), \boldsymbol{b}_v, \boldsymbol{b}'_v$;
Output: : Service migration decision $\boldsymbol{b}'_v$;
initialize the PV set $\mathbb{K}_0^j = \varnothing$;
for $t = 0$ to $T - 1$ do
    Update the PV set $\mathbb{K}_t^j$ according to (1);
    Obtain $\boldsymbol{b}'_t$ by solving P4;
    Update $Q_j(t+1)$ according to (32);
    Update $\bar{E}(j,t+1)$ according to (25),
    Update $b_v(t+1)$ according to (5);
end for
```

According to [21], we find that the average latency satisfies

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{v=1}^{V} \bar{L}_1(v,t) \leq \frac{JU}{\mu} + O^*. \tag{35}$$

where O^* is the optimal value of **P2**. Meanwhile, the energy consumption in parking lot j satisfies

$$\sum_{k\in\mathbb{K}_t^j} \sum_{v=1}^{V} e(v,k,j,t) \leq \frac{\bar{E}(j,t)}{\mathbb{E}\{\tau(j)\}} + \frac{U}{\epsilon^*} + \frac{\mu}{J\epsilon^*}(O^{max} - O^*), \tag{36}$$

where O^{max} represents the long-term latency achieved by no migration strategy. It demonstrates an $[O(1/\mu), O(\mu)]$ latency-energy trade-off. The latency performance of **P4** improves with $\mu \to \infty$. However, it also results in the large energy deficit queue. That is, the energy consumption in PVs grows linearly when μ increases.

4.4. Serving PV Selection

After solving **P2**, we can know the vehicles which request service from parking lots. In each time slot t, we need to select the serving PVs for these vehicles according to the service latency and available energy in PVs. A possible solution is to randomly choose a PV for each vehicle. However, the latency performance may not be optimal. Thus, we propose a modified Hungarian algorithm, which can fully utilize the resources of PVs in each parking lot, to solve **P3**.

First of all, we build a cost function $F(v,k,j,t)$ to calculate the fitness of a migration decision $p'_t(v)$ (i.e., $k = p'_t(v)$). A low value of $F(v,k,j,t)$ means the vehicle gains a big benefit when it is served by PV k.

$$F(v,k,j,t) = \alpha \bar{L}_2^j(v,t) - \left[\frac{E(k,j,t)}{\Delta_k} - e(v,k,j,t)\right], \tag{37}$$

where Δ_k is the number of service requests that PV k offers. $\alpha > 0$ is the weight parameter. The modified Hungarian algorithm is shown in Algorithm 3. It works as follows.

1. Compare the the number of service requests of vehicles $\mathcal{S}(\mathcal{V}_t^j)$ with the number of available PVs $\mathcal{S}(\mathcal{K}_t^j)$, and let $Z = \max\{\mathcal{S}(\mathcal{V}_t^j), \mathcal{S}(\mathcal{K}_t^j)\}$ (Line 1).
2. $Z > \mathcal{S}(\mathcal{K}_t^j)$ means the PVs are not enough to achieve one-to-one service. We generate a expanded set $\mathcal{K}_t^{j*}$, which size is equal to Z. When $Z = \mathcal{S}(\mathcal{K}_t^j)$, $\mathcal{K}_t^{j*} = \mathcal{K}_t^j$ (Lines 2–10).
3. If $Z > \mathcal{S}(\mathcal{V}_t^j)$, we generate a expanded vehicle set $\mathcal{V}_t^{j*}$, $\mathcal{V}_t^{j*} = \mathcal{V}_t^j$ otherwise (Lines 11–14).
4. Next, we initialize a $\mathcal{S}(\mathcal{V}_t^j)$-by-$\mathcal{S}(\mathcal{K}_t^j)$ cost matrix $\mathcal{F}$, where each element $\mathcal{F}(v,k), v \in \mathcal{V}_t^{j*}, k \in \mathcal{K}_t^{j*}$ is the cost value $F(v,k,j,t)$ (Line 15). In this way, we can obtain the migration decision $p'_v(t)$ by traditional Hungarian algorithm (Line 16).

Algorithm 3 Modified Hungarian algorithm.

```
Input: :$\mathcal{V}_t^j, \mathcal{K}_t^j$;
Output: : Service migration decision $\boldsymbol{p}'_v$;
Initialize Z $= \max\{\mathcal{S}(\mathcal{V}_t^j),\ \mathcal{S}(\mathcal{K}_t^j)\}$;
Let $\mathcal{K}_t^{j*} = \mathcal{K}_t^j$;
while $Z > \sum_{k \in \mathcal{K}_t^j} \Delta_{k^*}$ do
    Find the maximum $\frac{E(k^*,j,t)}{\Delta_{k^*}}$;
    $\Delta_{k^*} = \Delta_{k^*} + 1$;
end while
for $k \in \mathcal{K}_t^j$ do
    Make $\Delta_k$ copies of the the PV $k$;
    Add the copies of PV $k$ into $\mathcal{K}_t^{j*}$;
end for
Let $\mathcal{S}(\mathcal{V}_t^{j*}) = \mathcal{S}(\mathcal{V}_t^j)$
while $Z > \mathcal{S}(\mathcal{V}_t^j)$ do
    Add $Z - \mathcal{S}(\mathcal{V}_t^j)$ virtual vehicles $v^*$ into $\mathcal{S}(\mathcal{V}_t^{j*})$, which the cost function $F(v^*, k, j, t) = 0$;
end while
Initialize cost matrix $\mathcal{F}$ according to (37) based on $\mathcal{V}_t^{j*}, \mathcal{K}_t^{j*}$;
Obtain $p'_t(v)$ according to traditional Hungarian algorithm;
```

The time-complexity of the modified Hungarian algorithm relies on the difference between cost function of PVs for the vehicles. When the cost function values are same, the time-complexity becomes higher, which is at most $O(\mathcal{S}^3(\mathcal{V}_t^{j*}))$.

5. Performance Evaluation

In order to evaluate the performance of SEA algorithm, we conduct simulation experiments. We build a PVEC system with 10 RSUs and 5 parking lots, deployed in the map. PVs behavior is formulated based on a real dataset which are collected by SmartParking App [37]. Figure 2 shows the cumulative distribution function of the parking duration for PVs. The vehicle trajectories are collected from the real-world Rome taxi traces obtained in 2014 [38]. Four typical vehicular services, i.e., emergency stop, collision risk, accident report and parking, are used as the service that the vehicles may request for. The parameters of the services are listed in Table 2. The other major parameters are listed in Table 3.

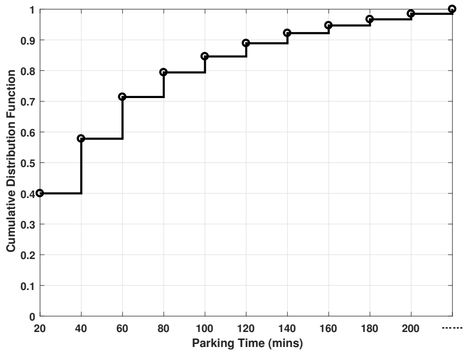

Figure 2. The cumulative distribution function of the parking duration for parked vehicles (PVs).

Table 2. Service parameters.

Type	D_m	λ_m	γ_m (KB)	$f_{m,v}$ (GHz)	$\theta_{m,v}$ (MB)
emergency stop	0.1	3200	36	[1.6, 3.2]	[1, 3]
collision risk	0.1	4800	40	[2.4, 4.8]	[1, 3]
accident report	0.5	4800	28	[2.4, 4.8]	[2, 3]
parking	0.1	1200	80	[0.6, 1.2]	[3, 5]

Table 3. Experimental parameters.

Notation	Value
η_{min}	5 Mbps
η_{max}	10 Mbps
$\bar{\eta}_{max}$	20 Mbps
κ	10^{-26}
P	0.5 W

We compare SEA with three benchmarks.

- Never-migrate scheme (NM): The service is deployed in the original SP and never migrates to other SPs. Note that all the service in PVs randomly migrate to other PVs when the PV leaves the parking lot.
- Always Migration (AM): the service is always migrated to the nearest SP, while the service in the parking lot is assigned to the PVs that can provide best latency performance regardless of the energy consumption constraint in PVs.
- Dynamic Markov Decision Process (DMDP) [33]: This is a single-user service migration algorithm, where the serving RSU or PV is predicted using the MDP model. The optimal service migration decision is made to minimize the energy consumption. DMDP were applied to each of the users independently in our simulations.

5.1. Impact of Different Total Number of Vehicles

Figure 3 shows the average latency with different total number of vehicles. It was observed that the average latency with NM is the highest, which is owing to a significant amount of latency for request transmission. On the other hand, the lowest average latency is achieved by AM, as AM always migrate service to the nearest SPs and further migrate service to PV that can provide best latency performance. Since the objective of DMDP is to minimize the energy cost, the average latency of DMDP is higher than that of AM and of SEA. Meanwhile, with the total number of vehicles increases from 100 to 1000, the average latency greatly increases to reduce the migration energy consumption. The average latency of SEA is quite close to that of AM when the total number of vehicles is smaller than 500. When the total number of vehicles exceeds 500, SEA slowly grows with the limited energy of PVs. However, it is still very close to AM.

Figure 4 shows the average utilization rate of PVs with different total number of vehicles. As can be seen from the figure, the utilization rate of PVs for NM is lowest. This is expected as the services were only migrated when the serving PV was leaving the parking lot. Due to the service requests concentrating on PVs that can provide best latency performance, the utilization rate of PVs of AM is near-minimal. Meanwhile, the utilization rate of PVs of DMDP maintain a second highest performance. It was observed that the highest average utilization rate is achieved by SEA as the modified Hungarian algorithm takes the advantage of available energy of PVs. The average utilization rate of PVs with SEA improves by up to 22.2%, as compared with the state-of-the-art solution.

Figure 5 shows the average additional energy consumption, which is the sum of the energy consumption exceeding the available energy in PVs divided by total number of PVs, with different total number of vehicles. The highest average additional energy consumption is achieved by AM, where increases exponentially with the total number of vehicles increases from 100 to 1000. This is because too many service requests are concentrated on a PV, resulting in the rapid depletion of energy in this PV. Meanwhile, the average additional energy consumption of NM is near to highest, which is caused by a significant amount of energy being required for requesting transmissions. The average additional energy of DMDP is near to optimal with the MDP model can not predict the parking duration of PV. The average additional energy consumption of SEA is near to zero when the vehicle number below 600. Even with the total number of vehicles increases from 600 to 1000, the average additional energy consumption grow slowly.

Figure 6 shows the average surplus energy in PVs, which is the sum of the residual energy in PVs divided by total number of PVs, with different total number of vehicles. Note that if the energy consumption in PV exceeds the available energy, the surplus energy is zero. The performance of NM, AM and DMDP are consistent with those in Figure 5. The highest utilization rate of PVs of SEA lead the energy in PVs is fully exploited. Thus, the average surplus energy of SEA reduce rapidly to 36.4 mJ with the total number of vehicles increase from 100 to 1000. Such observations above indicate that SEA algorithm is suitable for PVEC networks.

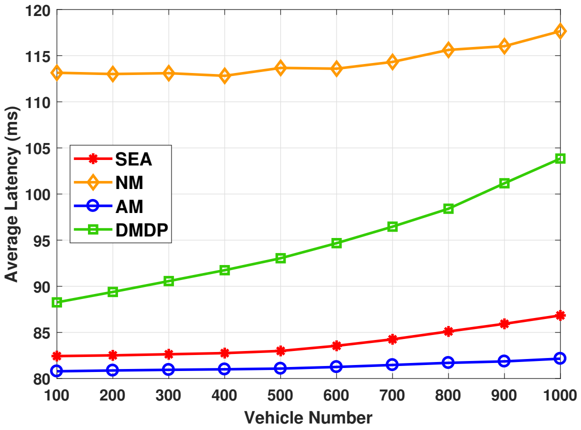

Figure 3. The average latency with different total number of vehicles.

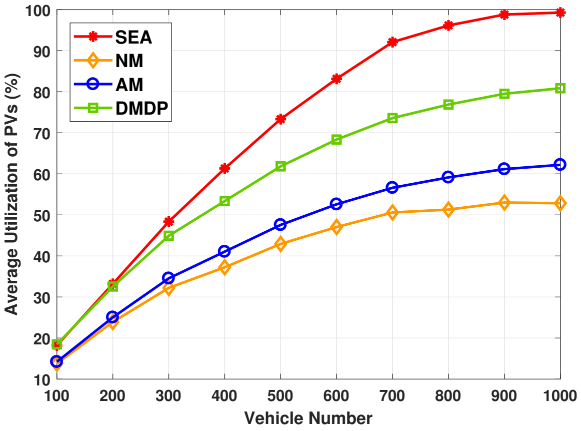

Figure 4. The average utilization rate of PVs with different total number of vehicles.

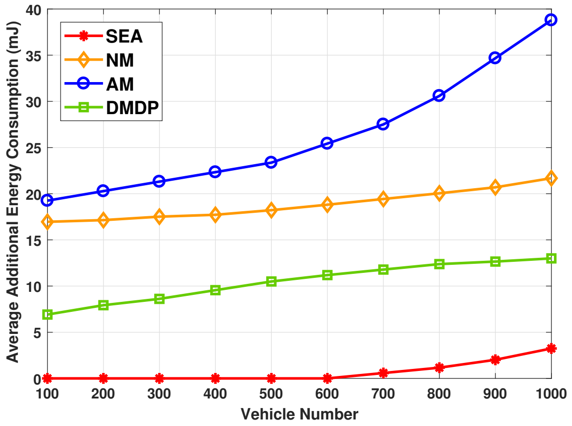

Figure 5. The average additional energy with different total number of vehicles.

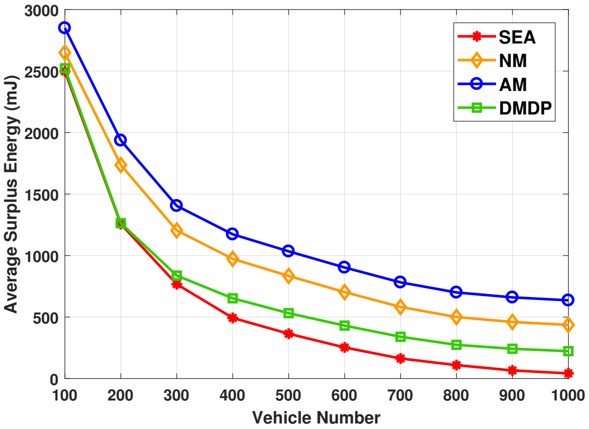

Figure 6. The average surplus energy with different total number of vehicles.

5.2. Impact of the Weight Parameter in the Cost Function

Figure 7 shows the average latency and average additional energy consumption with different α, while the total number of vehicle is 1000. By increasing α from 0.01 to 100, SEA algorithm puts more emphasis on the average latency, i.e., the average latency decreases and the average additional energy consumption increases. By adjusting the value of α, we can achieve a balance between these two. In our simulation experiments, we set $\alpha = 10$.

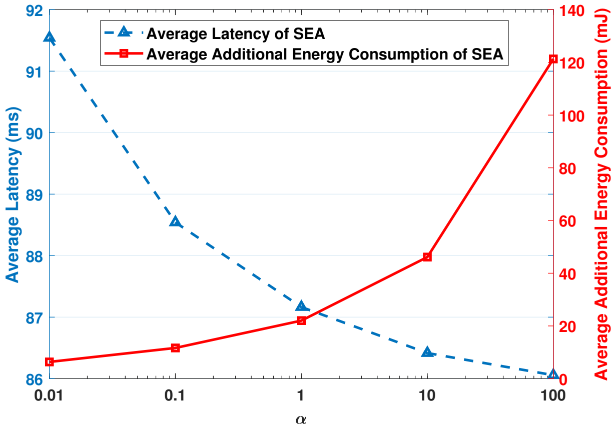

Figure 7. The average latency and average additional energy consumption with different α, while the total number of vehicles is 1000.

6. Conclusions

In this paper, we investigated a service migration problem in PVEC networks. We formulated the service migration process as an MINP optimization problem to minimize the average service latency while satisfying the energy constraints of PVs. A two-stage algorithm, called the SEA algorithm, was proposed to solve the MINP problem, which divides the original problem into the service migration between SPs and the selection of PVs for migration in parking lots sub-problems. Utilizing Lyapunov optimization, we perform service migration decision between SPs in an online fashion without requiring future information. Meanwhile, a modified Hungarian algorithm is proposed to effectively tackle the difficult in selection of PVs with different available energy and services. A series of simulation experiments based on the real-world traces were conducted to verify the superior performance in reducing the average latency with the energy constraint of PVs by our proposed SEA algorithm. It demonstrated that the proposed solution can reduce average latency by up to 17.2% and improve the average utilization rate of PVs by up to 22.2%, as compared with the state-of-the-art solution DMDP. It indicated that SEA algorithm is suitable for PVEC networks. Besides the available energy, storage and computing resources also affect the migration decision, which will be taken into account in future studies.

Author Contributions: Investigation, S.G.; X.H. and X.Z.; methodology, S.G.; Project administration, M.C. and X.Z.; Resources, M.C. and X.H.; Visualization, S.G.; Writing-original draft, S.G. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Natural Science Foundation of Tianjin under Grant No. 17JCQNJC00700, 18ZXZNGX00040 and 18ZXJMTG00290.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, R.; Zhang, J.; Song, S.H.; Letaief, K.B. Mobility-Aware Caching in D2D Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 5001–5015. [CrossRef]
2. Qiu, T.; Li, B.; Qu, W.; Ahmed, E.; Wang, X. TOSG: A Topology Optimization Scheme With Global Small World for Industrial Heterogeneous Internet of Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3174–3184. [CrossRef]
3. Zhang, Y.; Li, C.; Luan, T.H.; Fu, Y.; Shi, W.; Zhu, L. A Mobility-Aware Vehicular Caching Scheme in Content Centric Networks: Model and Optimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3100–3112. [CrossRef]
4. Li, C.; Wang, S.; Huang, X.; Li, X.; Yu, R.; Zhao, F. Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach. *IEEE Internet Things J.* **2019**, *6*, 6079–6088. [CrossRef]
5. Shah, S.A.A.; Ahmed, E.; Imran, M.; Zeadally, S. 5G for Vehicular Communications. *IEEE Commun. Mag.* **2018**, *56*, 111–117. [CrossRef]
6. Wei, H.; Luo, H.; Sun, Y. Mobility-Aware Service Caching in Mobile Edge Computing for Internet of Things. *Sensors* **2020**, *20*, 610. [CrossRef]
7. Zhang, Y.; Wang, C.; Wei, H. Parked Vehicle Assisted VFC System with Smart Parking: An Auction Approach. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–7.
8. Sun, G.; Yu, M.; Liao, D.; Chang, V. Analytical Exploration of Energy Savings for Parked Vehicles to Enhance VANET Connectivity. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1749–1761. [CrossRef]
9. Morency, C.; Trépanier, M. *Characterizing Parking Spaces Using Travel Survey Data*; Tech. Rep. CIRRELT-2008-15; Cirrelt: Montréal, QC, Canada, 2008.
10. Elsayed, S.A.; Abdelhamid, S.; Hassanein, H.S. Proactive Caching at Parked Vehicles for Social Networking. In Proceedings of the IEEE International Conference on Communications, Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
11. Zhang, Y.; Wang, C.; Wei, H. Parking Reservation Auction for Parked Vehicle Assistance in Vehicular Fog Computing. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3126–3139. [CrossRef]

12. Yang, L.; Yao, H.; Wang, J.; Jiang, C.; Benslimane, A.; Liu, Y. Multi-UAV Enabled Load-Balance Mobile Edge Computing for IoT Networks. *IEEE Internet Things J.* **2020**, to appear, [CrossRef]
13. Gu, L.; Zeng, D.; Guo, S.; Ye, B. Leverage parking cars in a two-tier data center. In Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 4665–4670.
14. Chen, N.; Qiu, T.; Zhou, X.; Li, K.; Atiquzzaman, M. An Intelligent Robust Networking Mechanism for the Internet of Things. *IEEE Commun. Mag.* **2019**, *57*, 91–95. [CrossRef]
15. Maia, A.M.; Ghamri-Doudane, Y.; Vieira, D.; de Castro, M.F. Optimized Placement of Scalable IoT Services in Edge Computing. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IM, Washington, DC, USA, 8–12 April 2019; pp. 189–197.
16. He, W.; Yan, G.; Xu, L.D. Developing Vehicular Data Cloud Services in the IoT Environment. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1587–1595. [CrossRef]
17. Taleb, T.; Ksentini, A.; Frangoudis, P.A. Follow-Me Cloud: When Cloud Services Follow Mobile Users. *IEEE Trans. Cloud Comput.* **2019**, *7*, 369–382, [CrossRef]
18. Qiu, T.; Liu, J.; Si, W.; Wu, D.O. Robustness Optimization Scheme With Multi-Population Co-Evolution for Scale-Free Wireless Sensor Networks. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1028–1042, [CrossRef]
19. Sun, Y.; Zhou, S.; Xu, J. EMM: Energy-Aware Mobility Management for Mobile Edge Computing in Ultra Dense Networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2637–2646, [CrossRef]
20. Baek, M.; Jeong, D.; Choi, D.; Lee, S. Vehicle Trajectory Prediction and Collision Warning via Fusion of Multisensors and Wireless Vehicular Communications. *Sensors* **2020**, *20*, 288. [CrossRef]
21. Xu, J.; Chen, L.; Zhou, P. Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks. In Proceedings of the IEEE Conference on Computer Communications, INFOCOM, Honolulu, HI, USA, 15–19 April 2018; pp. 207–215, [CrossRef]
22. Litman, T.A. *Parking Management: Strategies, Evaluation and Planning*; Tech. Rep. 37074029; Victoria Transp. Policy Inst.: Victoria, BC, Canada, 2006.
23. Liu, N.; Liu, M.; Lou, W.; Chen, G.; Cao, J. PVA in VANETs: Stopped cars are not silent. In Proceedings of the INFOCOM, Shanghai, China, 10–15 April 2011; pp. 431–435.
24. Malandrino, F.; Casetti, C.; Chiasserini, C.F.; Sommer, C.; Dressler, F. Content downloading in vehicular networks: Bringing parked cars into the picture. In Proceedings of the 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications—(PIMRC), Sydney, Australia, 9–12 September 2012; pp. 1534–1539.
25. Balen, J.; Martinovic, G.; Paridel, K.; Berbers, Y. PVCM: Assisting multi-hop communication in vehicular networks using parked vehicles. In Proceedings of the IV International Congress on Ultra Modern Telecommunications and Control Systems, St. Petersburg, Russia, 3–5 October 2012; pp. 119–122.
26. Tonguz, O.K.; Viriyasitavat, W. Cars as roadside units: A self-organizing network solution. *IEEE Commun. Mag.* **2013**, *51*, 112–120. [CrossRef]
27. Qiu, T.; Li, B.; Zhou, X.; Song, H.; Lee, I.; Lloret, J. A Novel Shortcut Addition Algorithm with Particle Swarm for Multisink Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3566–3577. [CrossRef]
28. Zhang, J.; Huang, X.; Yu, R. Optimal Task Assignment with Delay Constraint for Parked Vehicle Assisted Edge Computing: A Stackelberg Game Approach. *IEEE Commun. Lett.* **2020**, *24*, 598–602. [CrossRef]
29. Xiong, K.; Leng, S.; Hu, J.; Chen, X.; Yang, K. Smart Network Slicing for Vehicular Fog-RANs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3075–3085. [CrossRef]
30. Ouyang, T.; Zhou, Z.; Chen, X. Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2333–2345, [CrossRef]
31. Zhu, C.; Tao, J.; Pastor, G.; Xiao, Y.; Ji, Y.; Zhou, Q.; Li, Y.; Ylä-Jääski, A. Folo: Latency and Quality Optimized Task Allocation in Vehicular Fog Computing. *IEEE Internet Things J.* **2019**, *6*, 4150–4161. [CrossRef]
32. Chen, Y.; Tsai, Y. A Mobility Management Using Follow-Me Cloud-Cloudlet in Fog-Computing-Based RANs for Smart Cities. *Sensors* **2018**, *18*, 489. [CrossRef] [PubMed]
33. Wang, S.; Urgaonkar, R.; Zafer, M.; He, T.; Chan, K.; Leung, K.K. Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process. *IEEE Trans. Netw.* **2019**, *27*, 1272–1288. [CrossRef]
34. Ksentini, A.; Taleb, T.; Chen, M. A Markov Decision Process-based service migration procedure for follow me cloud. In Proceedings of the IEEE International Conference on Communications, ICC, Sydney, Australia, 10–14 June 2014; pp. 1350–1354, [CrossRef]

35. Nadembega, A.; Hafid, A.S.; Brisebois, R. Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE. In Proceedings of the IEEE International Conference on Communications, ICC, Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6, [CrossRef]
36. Yu, X.; Guan, M.; Liao, M.; Fan, X. Pre-Migration of Vehicle to Network Services Based on Priority in Mobile Edge Computing. *IEEE Access* **2019**, *7*, 3722–3730, [CrossRef]
37. ACT Government Open Data Portal dataACT. Available online: https://www.data.act.gov.au/Transport/SmartParkingHistory/grth-myzs (accessed on 5 March 2020).
38. Bracciale, L.; Bonola, M.; Loreti, P.; Bianchi, G.; Amici, R.; Rabuffi, A. CRAWDAD Dataset roma/taxi (v. 2014-07-17). Available online: https://crawdad.org/roma/taxi/20140717 (accessed on 18 December 2019).