Real-Time QoS Optimization for Vehicular Edge Computing With Off-Grid Roadside Units

Yu-Jen Ku, *Student Member, IEEE*, Po-Han Chiang, *Student Member, IEEE*, and Sujit Dey, *Fellow, IEEE*

***Abstract*—To sustainably provide low-latency communication and edge computing for connected vehicles, a promising solution is using Solar-powered Roadside Units (SRSUs), which consist of small cell base stations and Mobile Edge Computing servers. However, due to the intermittent nature of solar power, SRSUs may suffer from a high risk of power deficiency, which will lead to severe disruption of vehicular edge computing applications. In this paper, we aim to address this challenge of Quality of Service (QoS) loss (i.e., edge computing service outage for vehicle users (VUs)). We formulate a QoS optimization problem for VUs and solve it in two phases: an offline solar energy scheduling phase, and an online user association and SRSU resource allocation phase. We simulate our proposed technique in a dense SRSU network environment with real-world urban vehicular traffic data and solar generation profile. The simulation results show that our proposed approach can significantly reduce QoS loss of vehicular edge computing applications using SRSUs, compared to existing techniques. Further, the results are beneficial to service providers and city planners to identify adequate SRSU configurations for expected solar energy generation and edge computing service demands.**

***Index Terms*—Solar energy, Multiuser channels, Mobile edge computing, Roadside unit.**

I. Introduction

Roadside Units (RSUs) equipped with small cell base stations (SBSs) are evolving as a key infrastructure to support connected vehicles. Due to the low latency and high throughput, communications provided by SBSs to connected vehicles, RSUs can enable or extend various vehicular applications, such as autonomous driving, road safety, infotainment, and collaboration services [2]. Further, when augmented with Mobile Edge Computing (MEC) servers, the RSUs can fulfill the computation-intensive needs of vehicular applications, while maintaining low latency, through offloading vehicle users' (VUs') computing tasks to RSUs. The scenario has been defined in literature as Vehicular Edge Computing (VEC) [3], [4].

In 2020, SBSs are projected to consume 4.4 TWh of energy and emit 2.3 million tons of carbon dioxide equivalent (CO_{2e}) [5], [6]. Furthermore, dense deployments of RSUs are expected in order to support the massive growth of emerging connected vehicles and their high throughput requirements [7], leading to further power consumption and carbon emissions. One promising solution is the use of renewable energy (RE) in wireless communications [8]. In order to enhance the sustainability of RSUs by easing their grid power consumption, we proposed the idea of Solar-powered Roadside Units (SRSUs) in [1], which consist of SBS, MEC, and a self-sustained solar system.

The main challenge of adopting RE in an SRSU network is the intermittent and fluctuating nature of RE (i.e., solar energy) generation [9]. RE-powered VEC must consider the SRSU's communication and computing resources as opportunistic due to the intermittent harvested RE. Further, RE-powered VEC must also consider the VU's high mobility and low application latency requirement.

In this work, we consider that VUs offload their applications (e.g., object recognition and collision prediction using camera or lidar data) to the MEC server of the associated SRSU. For these time-sensitive and computation-intensive applications, VUs will send the raw data to SRSU and receive the processed results with ultra-low latency. Such applications will inevitably suffer from service degradation when the communication and/or computing capacity of SRSU is limited. In this work, we aim to minimize Quality of Service (QoS) loss in a dense SRSU network. We define QoS loss as a weighted sum of instances of (i) service outage (when no SRSU can serve the VU) and (ii) service disruption (when the VU is handed over to another SRSU), over total number of VUs.

In our preliminary work [1], we proposed an offline QoS Loss Minimization Algorithm (QLM) to heuristically minimize the weighted QoS loss using SRSUs. However, QLM assumes accurate predictions of SRSUs' solar generations and VUs' offloading demands. The impact of prediction error on the performance of QLM was not discussed. Moreover, the offline solution provided by QLM cannot adapt to dynamic solar generation and offloading demands. Finally, QLM assumes unlimited battery capacity in order to provide an analytic solution, which is not viable in real-world SRSU deployment.

In this work, given: (i) predictions of SRSUs' solar generations and power consumptions, (ii) current VUs' locations, wireless channel conditions, and offloading demands, and (iii) current SRSUs' stored energy, communication, and computing resources, we propose to jointly solve solar energy scheduling, VU-SRSU association, and SRSU resource allocation problems. We propose to solve this problem in two phases: (i) solar energy scheduling phase, which determines battery

Manuscript received December 26, 2019; revised April 16, 2020; accepted June 8, 2020. Date of publication July 8, 2020; date of current version October 22, 2020. This work was supported by the National Science Foundation under Grant CNS-1619184. This work was presented in part at the 27th International Conference on Computer Communication and Networks (ICCCN) [1]. The review of this article is coordinated by Dr. Kaigui Bian. *(Corresponding author: Yu-Jen Ku.)*

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92037 USA (e-mail: yuku@ucsd.edu; pochiang@ucsd.edu; dey@ece.ucsd.edu).

Digital Object Identifier 10.1109/TVT.2020.3007640

charging/discharging for SRSUs in advance in order to schedule the available solar energy in each time slot, and (ii) user association and resource allocation phase, which decides VU-SRSU association and SRSU resource allocation in real-time to minimize the weighted QoS loss, based on the available energy determined from the first phase. Compared to QLM, the proposed solution adapts the solar generations and offloading demands dynamically in real-time. Our simulation results show that this approach produces up to a 54% reduction in the weighted QoS loss compared to our preliminary work in [1].

The contributions of this paper are summarized as follows:

1) To the best of our knowledge, this is the first work to address the problem of using SRSUs in vehicular edge computing. Specifically, the paper considers the problem of SRSU edge computing and small cell communication resource allocation problems given the real-time offloading demands of the fast moving VUs as well as the limited solar energy availabilities of SRSUs.
2) For the first time, service outage incurred when no SRSU can serve a VU and service disruption caused by VU handover between SRSUs are considered in defining QoS. We propose a weighted QoS objective function to incorporate preference between these two factors.
3) To optimize the weighted QoS, we propose a two-phase approach consisting of an offline solar energy scheduling (battery charging/discharging scheduling) phase and an online user association and SRSU resource allocation phase. The proposed approach is real-time adaptive to offloading demands, locations, and channel conditions of VUs, as well as SRSU resource availabilities.
4) To demonstrate the feasibility and effectiveness of the proposed technique, we develop a simulation framework consisting of real-world solar generation [10], urban traffic profiles [11], and offloading demands. The simulation results show that the proposed approach significantly reduces the weighted QoS loss compared to existing techniques.

The rest of the paper is organized as follows. We review the related work in Section II. In Section III, the overview of our system model and problem formulation is presented. In Section IV we introduce the proposed two-phase approach. The simulation results are presented in Section V and we conclude in Section VI.

II. Related Work

There have been various studies addressing either RE-powered wireless communication system [12]– [14] or RE-powered edge and cloud server network [15], [16]. However, they do not jointly consider both wireless communication and edge computing resources. For RE-powered MEC system, to jointly consider these resources while using RE as the only power supply, Mao *et al*. [17] address the fluctuating RE challenges for computation task offloading between a single BS-user link. Xu *et al*. [18], [19] characterize multiple aspects of RE-powered MEC system by Markov Decision Process (MDP) states and propose an online learning-based algorithm to minimize system delay, battery depreciation, and backup power supply cost. The above techniques only consider single-BS scenario, while our work considers load-balancing and intercell interference in the multi-BS scenario.

[20]–[22] address the challenges of RE-powered multi-BS system, where each BS is equipped with a MEC server. [20] and [21] provide online solutions to control MEC capacity based on Lyapunov optimization [23]. In [20], Chen *et al*. aim at minimizing system delay through workload balancing among BSs under their long-term energy availability constraint, which does not consider the real-time availability of RE. In [21], Wu *et al*. minimize the drop rate of computation task and downlink data traffic due to excessive delay or lack of RE. The authors propose a workload balancing and data traffic admission control solution. However, they model the computation task and the downlink data traffic separately. In VEC, delay constraint of vehicular applications usually jointly constrains both task execution and data transmission delay. Therefore, in this work, we consider a joint delay constraint consisting of execution and transmission delay. In [22], Gou *et al*. maximize the number of offloading users by an algorithm that iteratively decides SBS coverage, channel allocation, and MEC computing allocation. However, compared to our proposed technique, the iterative nature of the solution is not real-time adaptive to the current RE availability, VU traffic, and offloading demand.

The above studies do not consider challenges specific to characteristics of VUs, such as high mobility, fast-changing channel condition, and ultra-low delay constraint. On the contrary, RE-powered Vehicle-to-Everything (V2X) studies [24]–[26] take these VU characteristics into consideration. Yang *et al*. [24] and Atoui *et al*. [25], [26] both consider a straight stretch of road with RE-powered RSU deployed along it. Based on vehicles' locations and velocities, they schedule the uplink [24] and downlink [25], [26] data transmission between BSs and vehicles to maximize both network throughput [24] and the number of served vehicles [25], [26]. These studies focus on data transmission and do not consider the challenges for computation task offloading in VEC. Also, these studies require vehicle to buffer the data and transmit at the scheduled time slot, which is not feasible for time-sensitive vehicular applications that our research considers.

Without the use of RE, there are a few papers integrating both MEC and V2X with in-grid RSUs [3], [4]. In [4], Zhang *et al*. leverage vehicle-to-vehicle (V2V) technology and propose a predictive task offloading scheme to address the communication overhead when a vehicle is moving between different RSUs. In [3], Dai *et al*. balance the offloading tasks from vehicles by jointly considering vehicle mobility, transmission rate, and MEC computing capacity to minimize task completion delay. These two studies do not consider RE and how to utilize the opportunistic MEC computing and V2X communication resources given limited RE power supply is not discussed.

III. System Model and Problem Formulation

In this section, we will first introduce our system model. Then we define the weighted QoS loss and formulate a QoS loss

TABLE I
SUMMARY OF KEY NOTATIONS AND ABBREVIATION

Notation	Description	Notation	Description
$\mathcal{B}$	Index set of SRSU in the network	x_{bi}^t	Association indicator of VU i and SRSU b
$\boldsymbol{I}^t$	Index set of VU in the network	a_i^t	Location of VU
$K_{D,b}$	Available downlink subcarriers of SRSU b	P_b^t	Power consumption of SRSU b
$K_{U,b}$	Available uplink subcarriers of SRSU b	L_b^t	Scheduled solar energy for SRSU b
U_b	Maximum computing speed of MEC b	E_b^t	Battery level of SRSU b
γ	SINR threshold for user association	S_b^t	Generated solar energy of SRSU b
ω_i^t	Data generation rate of the on-board sensor of VU i	u_{bi}^t	Computing speed of MEC b allocated to VU i
c_i^t	Computing resource required for processing the uploaded data of VU i	$k_{U,bi}^t$	Number of uplink subcarriers of SBS b allocated to VU i
d_i^t	Maximum delay of *delay sensitive data*	$k_{DS,bi}^t$	Number of subcarriers of SBS b allocated to VU i for *delay sensitive downlink data*
ϵ_i^t	Data rate of *delay tolerant downlink data*	$k_{DT,bi}^t$	Number of subcarriers of SBS b allocated to VU i for *delay tolerant downlink data*
δ_i^t	Size of data processing result	E^{max}	Maximum battery capacity
θ_i^t	Maximum delay of *delay tolerant downlink data*		

t(superscript): at the t^{th} time slot

minimization problem. For ease of reference, we list the key notations of our system model in Table I.

A. Network and Channel Model

We consider an SRSU network with a set of SRSUs $\mathcal{B}$. Each SRSU consists of a communication module SBS and a computation module MEC server. For the sake of notation brevity, we will use SBS b and MEC b to represent the SBS and MEC server in SRSU $b \in \mathcal{B}$, respectively. The total operation time is equally divided into T time slots. The duration of each time slot is τ. At the t^{th} time slot, there is a set of VUs $\boldsymbol{I}^t = \{1, 2, \ldots, \ell^t\}$ in the network, where $\ell^t = |\boldsymbol{I}^t|$ is the number of VUs in $\boldsymbol{I}^t$. We denote the location of VU $i \in \boldsymbol{I}^t$ as a_i^t.

At the t^{th} time slot, let $\eta_{D,bi}^t$ be the signal-to-interference-noise ratio (SINR) of downlink transmission from SBS b to VU i. $\eta_{D,bi}^t$ is given by,

$$\eta_{D,bi}^t = \frac{p_b g_{bi}^t}{N_0 + \sum_{b' \neq b} p_b g_{b'i}^t} \tag{1}$$

where g_{bi}^t denotes the downlink channel gain, p_b is the transmit power of SBS b and N_0 is the noise level. b' is the interfering SBS, which operates the same frequency bands as SBS b.

Let $r_{D,bi}^t$ be the achievable downlink transmission rate from SBS b to VU i per subcarrier,

$$r_{D,bi}^t = W \log_2 \left(1 + \eta_{D,bi}^t\right), \tag{2}$$

where W is the bandwidth per subcarrier. Similarly, we denote p_i as the transmit power of VU i and h_{ib}^t as the uplink channel gain. The uplink transmission rate from VU i to SBS b per subcarrier can thus be represented as,

$$r_{U,bi}^t = W \log_2 \left(1 + \frac{p_i h_{ib}^t}{N_0}\right), \tag{3}$$

where the interference from other VUs is negligible with frequency reuse and bandwidth allocation techniques [27].

Note that in vehicular communication, the channel condition between SRSU and VU changes rapidly due to mobility of VU. Therefore, we assume the duration of time slot τ to be small enough so that the channel condition is unchanged within the time slot.

B. Workload Model

In this work, we consider the case that VU has no spare computing capacity, which is the case for current vehicles and will be so for a vast majority of vehicles in the near future. Therefore, each VU will offload all the computation tasks of its vehicular applications. We refer to these tasks as workloads. At the t^{th} time slot, each VU will generate a workload to be offloaded, which is modeled by the following parameters. First, ω_i^t is the data generation rate of the on-board sensor (e.g., camera or Lidar) on VU i, which will be uplink transmitted to the MEC server. Second, c_i^t is the computing resource required for processing the uploaded data, which is quantized as number of machine instructions. Third, δ_i^t is the processing result (e.g., an alert/guidance message), which will be downloaded by VU i. Fourth, d_i^t is the delay requirement from MEC server receives the data to VU i receives the result. Finally, VU may request to download extra information from the MEC server or the Internet, which has data size ϵ_{i}^t and delay constraint θ_i^t. Note that the MEC processing result is critical to driving safety and needs low latency, therefore, d_i^t is much smaller than θ_{i}^t. We refer to the MEC processed data as *delay sensitive downlink data*, and the extra information as *delay tolerant downlink data*.

C. SRSU Association and Resource Utilization

Let $x_{bi}^t = \{0, 1\}$ be the user association indicator at the t^{th} time slot. $x_{bi}^t = 1$ if VU i is associate with SRSU b (its data processing tasks are thus offloaded to SRSU b), and $x_{bi}^t = 0$ otherwise. At each time slot, we assume each VU can only associate with one SRSU. A MEC server, on the other hand, can serve workloads from different VUs by using techniques like Virtual Machine (VM) [28]. Also note that workload cannot be offloaded between different SRSUs.

To satisfy the workload demand, SRSU needs to allocate adequate amounts of computing and communication resources to each associated VU. In our case, the connection between VU and SBS will create two bearers, one default bearer and one

Guaranteed Bit Rate (GBR) bearer (i.e., dedicated bearer) [29]. Note that the *delay tolerant downlink data* is transmitted through the default bearer, we let $k^t_{DT,bi}$ be the number of downlink subcarriers allocated to VU i by SBS b for this bearer at the t^{th} time slot. On the other hand, the offloaded data and the *delay sensitive downlink data* are transmitted through the GBR bearer. We denote $k^t_{U,bi}$ and $k^t_{DS,bi}$ as the number of uplink and downlink subcarriers, respectively, used for the GBR bearer between VU i by SBS b. We also denote u^t_{bi} as the computing speed, which is quantized as machine instructions per second, of the VM server created for VU i by MEC b.

To ensure that the data generated by the on-board sensor will not be dropped due to VU's memory buffer overflowing, the average uplink transmission rate of VU i should be greater than (or equal to) the data generation rate ω^t_i of the on-board sensor. The uplink subcarriers allocated to VU i, henceforth, should satisfy the following constraint,

$$\sum_{b\in\mathcal{B}} x^t_{bi} r^t_{U,bi} k^t_{U,bi} \geq \sum_{b\in\mathcal{B}} x^t_{bi}\omega^t_i. \tag{4}$$

To satisfy the downlink delay constraint, the number of subcarriers allocated to VU i for the *delay tolerant downlink data* should satisfy,

$$\sum_{b\in\mathcal{B}} x^t_{bi} r^t_{D,bi} k^t_{DT,bi} \geq \sum_{b\in\mathcal{B}} x^t_{bi}\frac{\epsilon^t_i}{\theta^t_i}. \tag{5}$$

Note that the *delay sensitive downlink data* need to be processed and transmitted in low latency. Hence, the computing speed of VM server and downlink subcarriers allocated to VU i by SRSU b should satisfy the following,

$$\sum_{b\in\mathcal{B}} x^t_{bi}\left(\frac{c^t_i}{u^t_{bi}} + \frac{\delta^t_i}{r^t_{D,bi}k^t_{DS,bi}}\right) \leq \sum_{b\in\mathcal{B}} x^t_{bi} d^t_i. \tag{6}$$

On the other hand, the computing and communication resources of each SRSU are limited, which is constrained by the following three equations,

$$\sum_{i\epsilon\boldsymbol{I}^t} x^t_{bi}u^t_{bi} \ \leq\ U_b, \tag{7}$$

$$\sum_{i\epsilon\boldsymbol{I}^t} x^t_{bi}k^t_{U,bi} \ \leq\ K_{U,b}, \tag{8}$$

$$\sum_{i\epsilon\boldsymbol{I}^t} x^t_{bi}\left(k^t_{DS,bi} + k^t_{DT,bi}\right) \ \leq K_{D,b}, \tag{9}$$

where U_b is the maximum number of machine instructions the processor of MEC b can execute per second [30]. $K_{U,b}$ and $K_{D,b}$ are SBS b's maximum number of available sub-carriers for uplink and downlink transmission, respectively.

D. Power Consumption Model

Power consumption of each SRSU is modeled by the power consumption of MEC plus the power consumption of SBS. At the t^{th} time slot, we denote $P^t_{S,b}$ as the power consumption of MEC b, which linearly increases with the overall processor's computing speed [28]. Let $p_{M,b}$ be the idle power of MEC b and $p_{C,b}$ be the power consumption for each unit utilization of the processor's speed of MEC b. $P^t_{S,b}$ can then be represented by the following equation,

$$P^t_{S,b} = \ \tau p_{M,b} \ + \tau p_{C,b}\sum_{i\epsilon\boldsymbol{I}^t} x^t_{bi}u^t_{bi}. \tag{10}$$

Besides, power consumption of SRSU also includes energy consumed by the SBS. The energy consumption of SBS is the energy consumed by operating uplink and downlink transmissions. Power consumption of uplink transmission is the circuit power for demodulation and baseband processing. It increases linearly with the number of active subcarriers [31]. Secondly, operating downlink transmission consumes circuit and RF related power; both are linearly increasing with the number of active downlink subcarriers [32]. Hence, the power consumption of SBS at the t^{th} time slot can be expressed as:

$$\begin{aligned} P^t_{X,b} = \tau \sum_{i\epsilon\boldsymbol{I}^t} x^t_{bi}\left(p_{D,b}\left(\frac{\delta^t_i}{r^t_{D,bi}} + k^t_{DT,bi}\right) + p_{U,b}k^t_{U,bi}\right) \\ + \tau p_{N,b}, \end{aligned} \tag{11}$$

where $p_{N,b}$ is the idle power of SBS b, $p_{U,b}$ is the circuit power consumption per active uplink subcarrier, and $p_{D,b}$ is the joint circuit and transmission power consumption per active downlink subcarrier. The overall power consumption of SRSU b at the t^{th} time slot can, therefore, be represented as, $P^t_b = P^t_{S,b} + P^t_{X,b}$.

E. Solar Generation and Battery Model

At the t^{th} time slot, let S^t_b be the amount of energy harvested from the solar panel of SRSU b. We assume S^t_b is available at the beginning of the t^{th} time slot and will be immediately stored without any loss of energy. The battery level of SRSU b is denoted as E^t_b, which is constrained by energy causality and battery capacity. We assume battery is lossless and let $E^{max} \in (0, \infty)$ denote the battery capacity. Therefore, the battery level E^t_b should satisfy,

$$0 \leq \ E^t_b = E^{t-1}_b + S^t_b - P^t_b \leq E^{max}. \tag{12}$$

F. QoS Model

The evaluation of QoS in this paper is defined according to the instance of service outage and service disruption on workloads.

1) Service Outage: Because the energy, computing, and communication resources are limited, SRSUs may not be able to serve a VU while satisfying this VU's workload requirements (4)-(6). Because there is no computing capacity in a VU, service outage happens when its workload cannot be offloaded to any SRSU in the network. We denote the number of VUs experiencing service outage at the t^{th} time slot as C^t_{drop}, which can be calculated as,

$$C^t_{drop} \ = \sum_{i\epsilon\boldsymbol{I}^t}\left(1 - \sum_{b\in\mathcal{B}} x^t_{bi}\right), \tag{13}$$

and the *service outage rate* is $\frac{C^t_{drop}}{\ell^t}$, where $\ell^t = |\boldsymbol{I}^t|$ is the total number of VUs in the network at the t^{th} time slot.

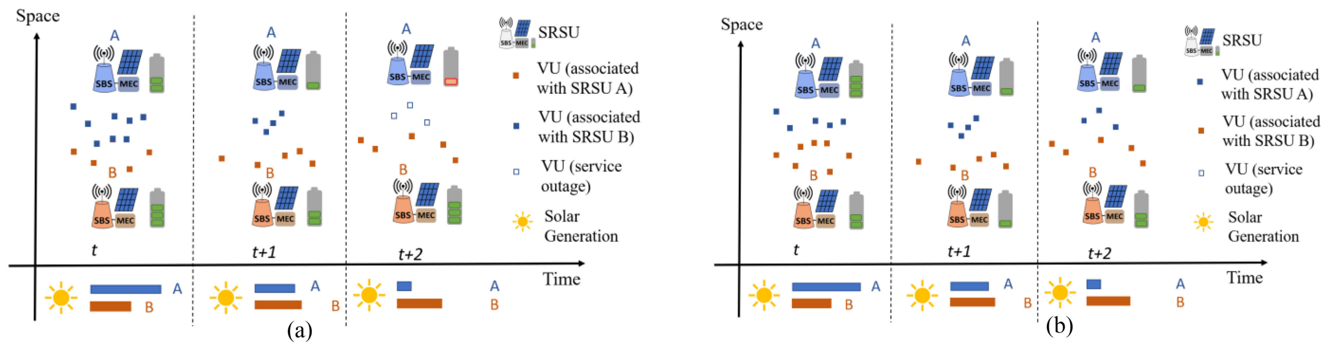

Fig. 1. Two dimensions that are involved in solving P1: offline solar energy scheduling (temporal dimension), and online user association and resource allocation (spatial dimension); also showing two scenarios describing the impact of energy scheduling (a) left, the condition with the absence of not performing energy scheduling at SRSU A and (b) right, the condition when performing energy scheduling at SRSU A.

2) Service Disruption: Service disruption happens to a VU when an SRSU hands it to another SRSU. The handover can take place when a VU is leaving an SRSU's coverage or when we actively change its associated SRSU. During the handover, the VU's workload cannot be offloaded, leading to service disruption. We denote the number of VUs experiencing service disruption at the t^{th} time slot as $C^t_{handover}$, which can be calculated as

$$C^t_{handover} = \sum_{i\epsilon \boldsymbol{I}^t} \left(\sum_{b\in\mathcal{B}} x^t_{bi} \right) \left(1 - \sum_{b\in\mathcal{B}} x^t_{bi} x^{t-1}_{bi} \right). \quad (14)$$

and the *service disruption rate* is $\frac{C^t_{handover}}{\ell^t}$.

The level of impact of the above two cases, service outage and service disruption, on driving experience is different. In the first case, the VU will be left unserved during the whole time slot. However, in the second case, the duration of handover disruption may be small. Once the VU is successfully associated with the next SRSU, it can then be served by the MEC server during the remaining period of the current time slot.

Therefore, we introduce a weighted factor $\kappa < 1$ on the *service disruption rate* to capture the different impacts on VUs between these cases. We then define the weighted QoS loss of the t^{th} time slot as $\mathcal{L}_t = (C^t_{drop} + \kappa C^t_{handover})/\ell^t$, and the weighted QoS loss of the total operation time as,

$$\mathcal{L} = \frac{\sum_{t=1}^{T} \left(C^t_{drop} + \kappa C^t_{handover} \right)}{\sum_{t=1}^{T} \ell^t}. \quad (15)$$

By properly adjusting κ, solving **P1** can effectively optimize QoS for VUs, depending on the network policy.

G. Problem Formulation

Our objective is to determine the user association x^t_{bi}, and the resource allocation u^t_{bi}, $k^t_{U,bi}$, $k^t_{DS,bi}$, and $k^t_{DT,bi}$ for VU i to minimize the weighted QoS loss of the total operation time. The decision is made at the beginning of each time slot based on the current SRSUs' available energy, computing, and computation resources, as well as VUs' locations, workload demands, and wireless channel conditions.

The optimization problem is formulated as,

$$\mathbf{P1}: \min_{x^t_{bi}, k^t_{U,bi}, k^t_{DT,bi}, k^t_{DS,bi}, u^t_{bi}\ \forall i\in \boldsymbol{I}^t,\ \forall t} \quad \mathcal{L}$$

$$\text{s.t. } (4)\text{–}(9), (12)$$

$$\sum_{b\in\mathcal{B}} x^t_{bi} \le 1, \quad \forall i \in \boldsymbol{I}^t, \quad t \in [1, T], \quad (16)$$

$$x^t_{bi} = \{0, 1\}, \quad \forall i \in \boldsymbol{I}^t, \quad t \in [1, T], \quad (17)$$

$$\sum_{b\in\mathcal{B}} x^t_{bi} \eta^t_{D,bi} \ge \sum_{b\in\mathcal{B}} x^t_{bi}\gamma, \ \forall i \in \boldsymbol{I}^t, t \in [1, T]. \quad (18)$$

Constraint (16), together with (17), state that the workload is not separable and cannot be offloaded to multiple SRSUs simultaneously. Moreover, constraint (18) limits a VU to only offload its workload to the SRSU that provides enough downlink SINR, with the threshold being set by γ.

Furthermore, we assume to have the knowledge of the predicted profiles of SRSU's solar energy generation and power consumption in advance. These data will help us plan the utilization of solar energy (i.e., the battery charging/discharging scheduling strategy) for each SRSU. SRSU power consumption and solar generation profiles are shown to be predictable in [10], [33]. We will list the prediction performance in Section V-B and further discuss the effect of prediction error on the optimization problem.

IV. Solution Methodology

The solution of **P1** involves decisions in two dimensions, as shown in Fig. 1. In the spatial dimension, feasible solutions of user association and resource allocation at each time slot should be decided to minimize the weighted QoS loss. However, the decision at each time slot is coupled with the temporal solar energy availability. As an example, if SRSU A in Fig. 1(a) uses most of its solar energy (shown in the blue bar) in the t^{th} time slot to serve as many VU as possible, 3 VUs at the $t + 2^{th}$ time slot will experience service outage due to the lack of solar energy. But if SRSU A reserves some energy and lets SRSU B serve more VUs than it served in Fig. 1(a), SRSU A will have enough energy to serve all its VUs at the $t + 2^{th}$ time slot, as Fig. 1(b) shows. Based on this observation, we follow the logic of [14], [34], and [35] to schedule the utilization of renewable energy for each time

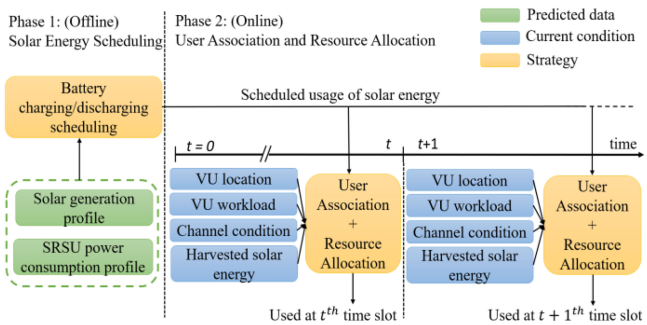

Fig. 2. The proposed two-phase approach, TQMA, to solve P1.

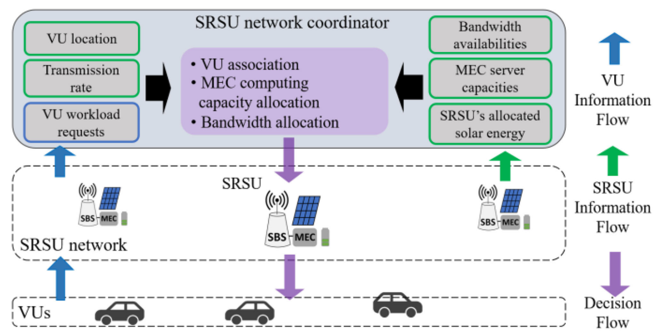

Fig. 3. Overview of the SRSU-assisted vehicular edge computing system, including request and decision flows.

slot in advance so that multiple BSs will not run out of renewable energy simultaneously. We therefore propose a two-phase QoS loss Minimization Algorithm (TQMA). TQMA solves P1 in two phases corresponding to the two dimensions: (i) solar energy scheduling phase (temporal dimension), and (ii) user association and resource allocation phase (spatial dimension). The process flow of TQMA is depicted in Fig. 2. Note that Phase 1 is executed offline based on the predicted profiles of SRSUs' solar generations and power consumptions, and Phase 2 is executed online based on current (i) VUs' workloads, locations, and transmission rates, and (ii) SRSUs' available communication, computing, and scheduled solar energy resources.

Fig. 3 shows the overview of the SRSU-assisted vehicular edge computing network and the information flows for Phase 2 of TQMA. At the beginning of each time slot, each VU will send the workload offloading request (blue arrows), including all the workload parameters, to the SRSU it associated with. Each SRSU will then send all the required information for Phase 2 decision to the SRSU network coordinator (green arrow). The SRSU network coordinator will make the Phase 2 decision and forward the resulting user association and SRSU resource allocation decisions back to SRSUs (purple arrows). Note that while the offloaded tasks are executed on the MECs associated with the SRSUs, the network coordinator and hence the proposed TQMA algorithm will be run in a separate server.

A. Phase 1 and Solar Energy Scheduling Algorithm (SESA)

We denote L_b^t as the scheduled solar energy of SRSU b at the t^{th} time slot, which will be regarded as the maximum allowable amount of energy for SRSU b to utilize at the t^{th} time slot. We also define $\pi_b^t = L_b^t/\hat{P}_b^t$ as SRSU b's Solar Utilization Ratio (SUR) for the t^{th} time slot, where $\hat{P}_b^t$ is the predicted SRSU power consumption. For SRSU b, the objective of Phase 1 is to maximize the minimum value of SUR within the whole operation time by optimally arranging the value of $L_b^t, t \in [1, T]$. Note that L_b^t needs to follow the energy causality constraint, $0 \leq \sum_{t'=1}^{t} \hat{S}_b^t - \sum_{t'=1}^{t} L_b^t \leq E^{max}, t \in [1, T]$, where $\hat{S}_b^t$ is the predicted solar generation profile for SRSU b.

The rationale is to distribute the solar energy at each time slot proportional to the SRSU's expected power consumption. This will prevent all SRSUs from having energy surplus and deficit at the same time. Therefore, neighboring SRSUs can better balance their power consumption based on their energy availability in Phase 2. Moreover, this can also prevent SRSUs from fully depleting their batteries during the hours when solar energy is not being generated.

It is inevitable that imperfect predictions will lead to a non-optimal $L_b^t, t \in [1, T]$ when applied to actual solar generation and SRSU power consumption. We will discuss the effect of prediction error on performance in Section V-B.

To arrange L_b^t, we propose the algorithm, SESA, which is shown in Algorithm 1. To begin with, we initialize L_b^t as $\hat{S}_b^t$ for each time slot $t \in [1, T]$. Let β_b^t be the expected battery level of SRSU b at the t^{th} time slot, which is initialized as zero. Let t_f be the last time slot that we can schedule the solar energy to. t_f is initialized as T in line 2 of SESA. To satisfy the energy causality constraint, we will start to schedule the solar energy iteratively from the last time slot to the beginning. At each iteration, we execute Procedure *DistributeEnergy* in SESA for the current time slot t. In Procedure *DistributeEnergy,* we will decide how much energy to be scheduled to each future time slot of t. We will first calculate the SUR π_b^t for t and the average SUR $\bar{\pi}$ for the time slots between t and t_f. If $\pi_b^t > \bar{\pi}$, we will decrease the value of L_b^t until the new π_b^t equals $\bar{\pi}$. The remaining energy will be distributed to time slots $t' \in (t, t_f]$. Each time slot t' will receive $\varepsilon^{t'}$ amount of energy that will be added to $L_b^{t'}$. We assume $\varepsilon^{t'}$ is proportional to the required energy for $\pi_b^{t'}$ to reach $\bar{\pi}$ for t'. The above steps are listed in lines 1-6 of *DistributeEnergy.*

However, during the scheduling process, the expected battery level may achieve the maximum capacity at any time slot between t and t_f. Assume the maximum capacity is achieved at t'', no more energy can be stored and scheduled from t to any time slot after t''. Let $t^* \in (t, t_f]$ be the earliest time slot that achieves the maximum battery capacity after $\varepsilon^{t'}$ is added to each time slot $t' \in (t, t_f]$. We then set its expected battery level $\beta_b^{t^*}$ to full and add the corresponding solar energy to $L_b^{t^*}$. After that, we split $(t, t_f]$ into two segments: $(t, t^*]$ and $(t^*, t_f]$, and recursively apply *DistributeEnergy* to these segments. The recursive process, which is shown in lines 13-17 of *DistributeEnergy*, ends when t^* doesn't exist within the new segment. Finally, we update the value of t_f and β^t, $t \in [1, T]$ in lines 15 and 19 of *DistributeEnergy*, then proceed to the next iteration. SESA will return $L_b^t, t \in [1, T]$, until the solar energy scheduling process is executed for all the time slots.

Therefore, at each time slot, SRSU b will drain $L_b^t - \hat{S}_b^t$ amount of energy from the battery if $L_b^t - \hat{S}_b^t \geq 0$, or store $\hat{S}_b^t - L_b^t$ amount of energy to the battery, otherwise.

The complexity of SESA is O(T^3), where T is the number of time slots. Since SESA is executed offline before the whole operation time starts, the complexity will not affect the real-time feasibility of our technique.

B. Phase 2 and the MRGAP Problem

In Phase 2, we formulate a user association and SRSU resource allocation problem to minimize the weighted QoS loss $\mathcal{L}_t$ at each time slot. At the t^{th} time slot, the above problem can be formulated as

$$\mathbf{P2}: \quad \min_{\boldsymbol{\chi}^t,\ \boldsymbol{\psi}^t} \frac{C^t_{drop}+\kappa C^t_{handover}}{\ell^t}$$

s.t. (4)–(9)

$$\sum_{b\in\mathcal{B}} x^t_{bi} \leq 1, \quad \forall i\epsilon \boldsymbol{I}^t, \tag{19}$$

$$x^t_{bi} = \{0,1\}, \quad \forall i\epsilon \boldsymbol{I}^t,\ \forall b\in\mathcal{B} \tag{20}$$

$$\sum_{b\in\mathcal{B}} x^t_{bi}\eta^t_{D,bi} \geq \sum_{b\in\mathcal{B}} x^t_{bi}\gamma,\ \forall i\epsilon \boldsymbol{I}^t, \tag{21}$$

$$P^t_b \leq \min\left(L^t_b, E^{t-1}_b + S^t_b\right)\ \forall b\in\mathcal{B} \tag{22}$$

where $\boldsymbol{\psi}^t = \{k^t_{U,bi}, k^t_{DT,bi}, k^t_{DS,bi}, u^t_{bi} | i\epsilon\boldsymbol{I}^t,\ b\in\mathcal{B}\}$ and $\boldsymbol{\chi}^t = \{x^t_{bi} | i\epsilon\boldsymbol{I}^t,\ b\in\mathcal{B}\ \}$. Constraints (19) and (20) state that the workload is not separable and can only be offloaded to one SRSU. Constraint (21) limits a VU to only associate with the SRSU which provides enough signal strength (with the SINR threshold be γ). Due to prediction error, it is possible that an SRSU's available energy is less than L^t_b. Therefore, the power consumption of SRSU should be limited by the minimum between actual available energy $S^t_b + E^{t-1}_b$ and scheduled solar energy L^t_b, in (22).

We next show that **P2** can be formulated as a variant of Multi-Resource Generalized Assignment Problem (MRGAP) [36]. MRGAP is originally proposed to minimize a total cost when assigning items to containers under multiple resource constraints. Given $\boldsymbol{N}$ is a set of items, $\boldsymbol{M}$ is a set of containers, and $\boldsymbol{K}$ is a set of multiple resources provided by containers to the items, MRGAP is formulated as

$$\mathbf{MRGAP}: \quad \min_{x_{mn}, n\in\boldsymbol{N}, m\in\boldsymbol{M}} \sum_{n\in\boldsymbol{N}}\sum_{m\in\boldsymbol{M}} z_{mn}x_{mn} \quad \text{s.t.} \quad \sum_{m\in\boldsymbol{M}} x_{mn} = 1\,, \quad \forall n\in\boldsymbol{N} \tag{23}$$

$$x_{mn} = \{0,1\}\,, \quad \forall n\in\boldsymbol{N},\ m\in\boldsymbol{M} \tag{24}$$

$$\sum_{n\in\boldsymbol{N}} v_{mnk}x_{mn}\ , \quad \forall m\in\boldsymbol{M},\ k\in\boldsymbol{K}. \tag{25}$$

where n is the index of the item, m is the index of the container, and k is the index of the resource. x_{mn} is the decision of whether to assign item n to container m. z_{mn} is the cost of assigning item n to container m, w_{mk} is the maximum capacity on resource k of container m, and v_{mnk} is the amount of resource k required to assign item n to container m. Finding the optimal solution of **MRGAP** is NP-Hard [37]. To map **P2** to **MRGAP**, we consider a special case where the assignment constraint (22) is relaxed to $\sum_{m\in\boldsymbol{M}} x_{mn} \leq 1,\ \ \forall n\in\boldsymbol{N}$, which allows items without any assignment. Different from conventional **MRGAP**, this special case always has a feasible solution.

Algorithm 1: SESA.

Inputs:
1) Predicted solar generation profile $\{\hat{S}^t_b \mid t\in[1,T]\}$
2) Predicted SRSU power consumption profile $\{\hat{P}^t_b \mid t\in[1,T]\}$
3) Battery capacity E^{max}

Output:
Scheduled solar energy $L = \{L^t_b \mid t\in[1,T]\}$

initialize $\beta \leftarrow zeros(1,T)$
$L^t_b \leftarrow\ \hat{S}^t_b,\ \forall t\in[1,T], t_f \leftarrow t^{end}$
for $t = t^{end}\ - 1:\ 1$ **do**
  update β, L, t_f using *DistributeEnergy*(β, L, t, t_f)
end for
return $L = \{L^t_b \mid t\in[1,T]\}$

Procedure *DistributeEnergy* (β, L, t_s, t_f, b):
calculate $\bar{\pi} \leftarrow \frac{\sum_{t=t_s}^{t_f} L^t_b}{\sum_{t=t_s}^{t_f} \hat{P}^t_b}$
calculate $\pi^t,\ \forall t\in[t_s, t_f]$
if $\pi^{t_s} > \bar{\pi}$ && $t_f > t_s$ **do**
  $\boldsymbol{J} \leftarrow \{\ t \mid \pi^t < \bar{\pi},\ t\in(t_s, t_f]\}$
  $\Delta\ \leftarrow L^{t_s}_b - \bar{\pi}\hat{P}^{t_s}_b,\ \beta' \leftarrow \beta,\ \varepsilon \leftarrow zeros(1,T)$
  calculate $\varepsilon^t,\ \forall\ t\in\boldsymbol{J}$
  calculate $\beta'^t \leftarrow \beta'^t + \sum_{t'=\ t+1}^{t_f} \varepsilon^{t'},\ \ \forall\ t\in[t_s, t_f)$
  $\tilde{\boldsymbol{T}} \leftarrow \{\ t \mid \beta'^t \geq\ E^{max},\ \forall\ t\in[t_1, t_f)\}$
  if $\tilde{\boldsymbol{T}} \notin \phi$ **do**
    $t^* \leftarrow \min_{t\in\tilde{\boldsymbol{T}}} t,\ \varepsilon^* \leftarrow (E^{max} - \beta'^{t^*})$
    $\beta^t \leftarrow \beta^t + \varepsilon^*,\ \forall t\in[\ t_s,\ t^*]$
    $L^{t_s}_b \leftarrow L^{t_s}_b -\ \varepsilon^*,\ L^{t^*+1}_b \leftarrow L^{t^*+1}_b + \varepsilon^*$
    update β, Γ from:
      DistributeEnergy(β, $L, t^*+1,\ t_f, b$)
    $t_f \leftarrow t^*$
    update β, Γ, t_f from:
      DistributeEnergy(β, $L, t_s,\ t_f, b$)
  else do
    $\beta \leftarrow \beta', L^{t_s}_b \leftarrow L^{t_s}_b - \Delta$
    $L^t_b \leftarrow L^t_b +\ \varepsilon_t,\ \forall\ t\in(t_s,\ t_f]$
    return β, Γ, t_f
  end if
else do
  return β, Γ, t_f
end if

Next, we show how **P2** is mapped to the relaxed case of **MRGAP**. Because **P2** has a constant denominator ℓ^t, we rewrite the numerator of its objective function,

$$C^t_{drop} + \kappa C^t_{handover} = \ell^t + \sum_{i\epsilon\boldsymbol{I}^t}\sum_{b\in\mathcal{B}}\left(-1+\kappa-\kappa\,\Omega\left(x^t_{bi}, x^{t-1}_{bi}\right)\right)x^t_{bi} \tag{26}$$

where $\Omega(x,y)$ is an indicator function, it returns 1 if $x\ =\ y$, or otherwise returns 0 (See Appendix A). Minimizing Eq. (26) is equivalent to minimizing its second term (i.e. the summation of $-1+\kappa-\kappa\ \Omega(x_{bi}^t, x_{bi}^{t-1})$), which can be mapped to z_{mn} in **MRGAP**. Let $\boldsymbol{M}$ be the SRSU set $\mathcal{B}$, $\boldsymbol{N}$ be the VU set $\boldsymbol{I}^t$, and $\boldsymbol{K}$ to contain resources of the (i) computing speed, (ii) downlink subcarriers, (iii) uplink subcarriers, and (iv) energy. Let v_{bi1}, v_{bi2}, v_{bi3}, and v_{bi4} be the amount of computing speed, the number of uplink subcarriers, the number of downlink subcarriers, and the corresponding power consumption allocated to VU i by SRSU b, respectively. Consequently, **P2** can be formulated as a special case of **MRGAP** with relaxed constraint (22) and additional constraints (21), (22), and (6).

Next, we develop a real-time heuristic algorithm H-URA for **P2**. To begin with, let v_{bik}^t denote the value of v_{bik} in the corresponding **MRGAP** problem of **P2** at the t^{th} time slot. We first show how many subcarriers for uplink and *delay tolerant downlink data* transmission are needed to serve VU i. The allocation of $k_{U,bi}^t$ and $k_{DT,bi}^t$ from SRSU b should follow constraints (4) and (5), respectively. Once these constraints are satisfied, there is no need to increase the value of $k_{U,bi}^t$ and $k_{DT,bi}^t$. The constraints in (4), (5) thus, can be reduced to deterministic allocation decision,

$$k_{U,bi}^t = \frac{\omega_i^t}{r_{U,bi}^t},\ k_{DT,bi}^t = \frac{\epsilon_i^t}{\theta_{\mathrm{i}}^{\mathrm{t}} r_{D,bi}^t}\,. \tag{27}$$

The value of v_{bi2}^t can, therefore, be set as $\omega_i^t / r_{U,bi}^t$ for VU i. On the other hand, the allocation of computing speed and downlink subcarriers for the *delay sensitive downlink data* should satisfy the joint delay constraint (6). Therefore, deterministic allocation decision does not exist. A reasonable way is to define v_{bi1}^t (required computing speed) and v_{bi3}^t (required downlink subcarriers) based on the availability of these two resources,

$$\begin{aligned} v_{bi1}^t &= \frac{K_{D,b}+U_b}{K_{D,b}}\left(\frac{c_i^t}{d_i^t}\right), \\ v_{bi3}^t &= \frac{K_{D,b}+U_b}{U_{b,b}}\left(\frac{\delta_i^t}{r_{D,bi}^t d_i^t}\right)+\frac{\epsilon_i^t}{\theta_i^t r_{D,bi}^t}\,. \end{aligned} \tag{28}$$

Meanwhile, v_{bi4}^t is set to be the power consumption for SRSU b when utilizing v_{bi1}^t, v_{bi2}^t, and v_{bi3}^t amount of resources.

With the value of v_{bi1}^t, v_{bi2}^t, v_{bi3}^t, and v_{bi4}^t, we propose to solve **P2** by heuristically solving the Lagrangian dual problem of its MRGAP form [38]. The Lagrangian dual of **P2** can be formulated as,

$$\begin{aligned} \mathbf{P2_{LD}}:& \max_{\lambda_b^t,\mu_b^t,\rho_b^t,\sigma_b^t\in\mathbb{R}_+, b\in\mathcal{B}}\ \min_{x_{bi}^t, b\in\mathcal{B}, i\epsilon\boldsymbol{I}^t} \sum_{i\epsilon\boldsymbol{I}^t}\sum_{b\in\mathcal{B}} z_{bi}^t x_{bi}^t \\ &+\sum_{b\in\mathcal{B}}\lambda_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t} x_{bi}^t v_{bi1}^t - U_b\right)+\sum_{b\in\mathcal{B}}\mu_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t} x_{bi}^t v_{bi2}^t - K_{U,b}\right) \\ &+\sum_{b\in\mathcal{B}}\rho_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t} x_{bi}^t v_{bi3}^t - K_{D,b}\right)+\sum_{b\in\mathcal{B}}\sigma_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t} x_{bi}^t v_{bi4}^t - L_b'^t\right) \\ &\text{s.t. (19)–(21),} \end{aligned}$$

where $L_b'^t = \min(\ L_b^t, E_b^{t-1}+S_b^t)$. λ_b^t, μ_b^t, ρ_b^t, and σ_b^t are the Lagrangian multipliers for dualizing constraints (7)-(9) and (22). The optimality of $\mathrm{P2_{LD}}$ for **P2** depends on the values of λ_b^t, μ_b^t, ρ_b^t and σ_b^t. However, since the workload demands will change in different time slots, the optimal values of these Lagrangian multipliers will also change. Consequently, traditional searching-based methods [36], [38] to find the optimal Lagrangian multipliers are time-consuming since the solution is only applicable to the current time slot. Therefore, we propose to define the Lagrangian multipliers as follows,

$$\begin{aligned} \lambda_b^t &= \gamma\frac{\sum_{i\epsilon\boldsymbol{I}^{t-1}} x_{bi}^{t-1}u_{bi}^{t-1}}{U_b},\ \mu_b^t = \gamma\frac{\sum_{i\epsilon\boldsymbol{I}^{t-1}} x_{bi}^{t-1}k_{U,bi}^{t-1}}{K_{U,b}}, \\ \rho_b^t &= \gamma\frac{\sum_{i\epsilon\boldsymbol{I}^{t-1}} x_{bi}^{t-1}k_{D,bi}^{t-1}}{K_{D,b}},\ \sigma_b^t = \gamma\frac{P_b^{t-1}}{L_b'^{t-1}} \end{aligned} \tag{29}$$

where γ is a constant scaling factor. The rationale is as follows. Consider two SRSUs which have the same z_{bi}^t to VU i, we tend not to assign this VU to the SRSU whose resources are more likely to be fully utilized. The likelihood relies on the resource utilization condition at the previous time slot.

lemma 1: With fixed λ_b^t, μ_b^t, ρ_b^t, and σ_b^t, solving $\mathrm{P2_{LD}}$ is equivalent to finding the SRSU which minimizes $q_{bi}^t = z_{bi}^t + \lambda_b^t v_{bi1}^t + \mu_b^t v_{bi2}^t + \rho_b^t v_{bi3}^t + \sigma_b^t v_{bi4}^t$ for each VU.

Proof: See Appendix B.

To further minimize the service disruption, we tend to assign VU to the SRSU that locates on its future path. We propose to use a Maximum Likelihood Markov Chain [39] to predict the probability of a VU's future location. First, we divide the network neighborhood into A non-overlapping areas. Each area is represented by a state in the Markov Chain. Second, we create an $|A|$ x $|A|$ transition matrix $\hat{A}^t$ for this Markov Chain at the t^{th} time slot, where $|A|$ is the size of A. We define $N_{s_1 s_2}^t$ as the total instances of VUs moving from area s_1 to area s_2 during any consecutive time slots before t. The state transition probability $\hat{A}_{s_1,s_2}^t$ can then be represented as $\hat{A}_{s_1,s_2}^t = N_{s_1 s_2}^t / \sum_{s\in A} N_{s_1 s}^t$. Let b_{s_2} be the SRSU which provides the best signal strength to the geological center of area s_2. If a VU is in area s_1, the probability that b_{s_2} is the next SRSU for this VU to associate in the next time slot is predicted as $\hat{A}_{s_1,s_2}^t$. This probability is then multiplied by κ and added to q_{bi}^t for each VU-SRSU pair. For each $s \in A$, the complexity of calculating $\sum_{s\in A} N_{s_1 s}^t$ is O($|A|$) and hence the complexity of updating $\hat{A}_{s_1,s_2}^t$, $s_1 \in A, s_2 \in A$ is O($|A|^2$). Note that in an SRSU network, the number of VU is usually larger than $|A|$. Therefore, O($|A|^2$) $<$ O(ℓ^2).

Based on lemma 1 and $\hat{A}^t$, we assign each VU to the SRSU which corresponds to the VU's minimal q_{bi}^t. However, this assignment may not be valid since we relax constraints (7)-(9), and (22) in **P2**. Therefore, we propose to make association decisions for VUs one by one while checking if the decision satisfies the relaxed constraints. We will pick the VU which has the largest difference between its best and second-best q_{bi}^t, $b \in \mathcal{B}$, as the highest priority VU to make the association decision for. We then assign the VU to the SRSU that corresponds to the best q_{bi}^t if the constraints (7)-(9), (21), (22) of **P2** can be satisfied, and proceed to the next VU.

Algorithm 2: H-URA.

Inputs:

1) The scheduled solar energy, battery level and solar generation $L_b^t,\ E_b^{t-1}, S_b^t,\ \forall\, b \in \mathcal{B}$
2) VU location $\{a_i^t\}$, and workload $\{\omega_i^t, c_i^t, \delta_i^t, d_i^t, \epsilon_i^t, \theta_{\mathrm{i}}^{\mathrm{t}}\},\ \forall i \in \boldsymbol{I}^t,$
3) Channel conditions $\{g_{bi}^t |\ i \in \boldsymbol{I}^t, b \in \mathcal{B}\}$
4) System Parameters γ, E^{max}, $K_{D,b}$, $K_{U,b}$, and $U_b,\ \forall b \in \mathcal{B}$
5) Previous association indicators $x_{bi}^{t-1},\ \forall i \in \boldsymbol{I}^t,\ \forall b \in \mathcal{B}$
6) Next SRSU probability prediction $\hat{A}^t$
7) Lagrangian multipliers λ_b^t, μ_b^t, ρ_b^t, and $\sigma_b^t, \forall b \in \mathcal{B}$

Output:

1) User association $\boldsymbol{\chi}^t$, and Resource allocation $\boldsymbol{\psi}^t$

1: Initialization: $L_b'^t \leftarrow \min(\ L_b^t, E_b^{t-1} + S_b^t), \forall b \in \mathcal{B}$
2: *visit_UE* $\leftarrow 0$
3: $\boldsymbol{Q}^t \leftarrow \{q_{bi}^t\}_{b\in\mathcal{B},\ i\in\boldsymbol{I}^t}$
4: **while** *visit_UE* $\leq \ell$ && $\exists q_{bi}^t \neq \infty$ **do**
5: **for** $\forall\ i \in \boldsymbol{I}^t$ **do**
6: $b_i^1 \leftarrow argmin_{b\in\mathcal{B}}\, \boldsymbol{Q}_{bi}^t$
7: $b_i^2 \leftarrow argmin_{b\in\mathcal{B}\setminus\{b_i^1\}}\, \boldsymbol{Q}_{bi}^t$
8: **end for**
9: $i^* \leftarrow \max_i \boldsymbol{Q}_{b_i^2 i}^t - \boldsymbol{Q}_{b_i^1 i}^t,\ b^* \leftarrow b_i^1,$ $\boldsymbol{\zeta'} \leftarrow \{i |\ x_{b^*i}^t =\ 1\} + \{i^*\}$
10: $\{\tilde{u}_{b^*i}^t,\ \tilde{k}_{DS,b^*i}^t,\ \tilde{k}_{U,b^*i}^t,\ \tilde{k}_{DT,b^*i}^t | i \in \boldsymbol{\zeta'}\}, \leftarrow$ *MCPA(*$\boldsymbol{\zeta'}, b^*$*)*
11: **calculate** $P_{b^*}^t$ **using** (11)
12: **if** *MCPA(*$\boldsymbol{\zeta'}, b^*$*)*$\neq 0$ && $P_{b^*}^t \leq L_{b^*}'^t$ **do**
13: $x_{b^*i^*}^t \leftarrow 1,$
14: **for** $\forall\ i \in \boldsymbol{\zeta'}$ **do**
15: $k_{U,b^*i}^t \leftarrow \tilde{k}_{U,b^*i}^t, k_{DT,b^*i}^t \leftarrow \tilde{k}_{DT,b^*i}^t,$ $u_{b^*i}^t \leftarrow \tilde{u}_{b^*i}^t, k_{DS,b^*i}^t \leftarrow \tilde{k}_{DS,b^*i}^t$
16: **end for**
17: $\boldsymbol{Q}_{bi^*}^t \leftarrow \infty, \forall b \in \mathcal{B}$, *visit_UE* $\leftarrow$ *visit_UE*+1
18: **else do**
19: $\boldsymbol{Q}_{b^*i^*}^t \leftarrow \infty, \boldsymbol{\zeta'} \leftarrow \boldsymbol{\zeta'}\setminus\{i^*\}$
20: **end if**
21: **end while**
22: **return** $\boldsymbol{\chi}^t,\ \boldsymbol{\psi}^t$

Procedure *MCPA(*$\boldsymbol{\zeta}, b$*)*:

1: **for** $\forall\ i \in \boldsymbol{\zeta}$ **do**
2: **calculate** $\tilde{u}_{bi}^t,\ \tilde{k}_{DS,bi}^t,\tilde{k}_{U,bi}^t,\tilde{k}_{DT,bi}^t$ using (27) and (31)
3: **end for**
4: **if** constraints (7)-(9), and (22) are satisfied for SRSU b and every $i \in \boldsymbol{\zeta}$ satisfies (21) and $H_b^t{>}0$
5: **return** $\{\tilde{u}_{bi}^t,\ \tilde{k}_{DS,bi}^t,\tilde{k}_{U,bi}^t,\tilde{k}_{DT,bi}^t | i \in \boldsymbol{\zeta}\}$
6: **else**
7: **return** 0
8: **end if**

To check if a VU association satisfies the constraints (7)-(9), (21), (22) of **P2** and determine the optimal resource allocation decision, we adopt the procedure Minimize SRSU Power Consumption Algorithm (*MPCA*), which is proposed in our previous work [1]. Given a VU set $\boldsymbol{\zeta}$ of an SRSU, *MPCA* will first check if the SRSU can serve all the workloads from $\boldsymbol{\zeta}$. If possible, then *MPCA* will allocate computing and communication resources to the VUs in $\boldsymbol{\zeta}$ while minimizing the power consumption of the SRSU (with the rationale to save solar energy). *MPCA* determines the optimal resource allocation as follows. We have argued the optimal value of $k_{U,bi}^t$ and $k_{DT,bi}^t$. To show the optimal allocation of $k_{DS,bi}^t$ and u_{bi}^t in Eq. (31) for a given VU set $\boldsymbol{\zeta}$ of SRSU b, we define the following terms for all the VUs in $\boldsymbol{\zeta}$,

$$l_i^t = \frac{\delta_i^t}{r_{D,bi}^t d_i^t}\ ,\ \ \varphi_i^t = \frac{l_i^t c_i^t}{d_i^t}\ ,\ \ \varpi_b^t = \sum_{i\epsilon\boldsymbol{\zeta}} k_{DT,bi}^t\ ,$$
$$H_b^t = \frac{\sum_{i\epsilon\boldsymbol{\zeta}} (\varphi_i^t)^{1/2}}{K_{D,b} - \varpi_b^t - \sum_{i\epsilon\boldsymbol{\zeta}} l_i^t}\ ,\ y_i^t = \frac{(c_i^t)^{1/2}}{d_i^t} + \frac{(l_i^t)^{1/2}}{(d_i^t)^{1/2}} H_b^t. \tag{30}$$

Then, the optimal resource allocation for u_{bi}^t and $k_{DS,bi}^t$ will be,

$$u_{bi}^t = \left\lceil y_i^t \left(c_i^t\right)^{1/2} \right\rceil,\ \ k_{DS,bi}^t = \left\lceil \frac{y_i^t (l_i^t d_i^t)^{1/2}}{H_b^t} \right\rceil, i\epsilon\, \boldsymbol{\zeta}. \tag{31}$$

The above resource allocation solution to minimize power consumption of the SRSU can be solved by analyzing the problem's Karush–Kuhn–Tucker (KKT) conditions [40] or using convex optimization programming tools [41]. We omit the proof here for the sake of brevity.

MPCA returns 0 if the KKT conditions are violated or constraints (7)–(9), (21), or (22) are not satisfied. Otherwise, *MPCA* returns the optimal resource allocation decisions u_{bi}^t, $k_{U,bi}^t$, $k_{DT,bi}^t$, and $k_{DS,bi}^t$ for each VU in $\boldsymbol{\zeta}$.

Based on the above discussion, we propose H-URA for real-time user association and SRSU resource allocation, which is shown in Algorithm 2. The pseudocode of *MPCA* is also included in Algorithm 2. H-URA takes real-time VUs' locations workload demands, and channel conditions, as well as SRSUs' resource availabilities and Lagrangian multipliers as input. To begin with, $\boldsymbol{Q}^t$ in line 3 of H-URA records the value of q_{bi}^t for all the VU-SRSU pairs. The user association procedure is determined by the *while* loop in lines 4-21. H-URA will decide the highest priority VU to make the association decision for in lines 5-9. If H-URA determines VU i^* as the highest priority VU and b^* is the SRSU corresponds to its minimal q_{bi}^t, then H-URA will consider associating i^* with b^*. H-URA will check if this association satisfies all the constraints of **P2** in lines 10 and 12 by using *MPCA*. If constraints are satisfied, H-URA will confirm the association, update the association indicator and resource allocation decisions in lines 13-16. Note that $\boldsymbol{\zeta'}$ in line 9 is the set of VUs that have been associated with SRSU b^* by H-URA. The elements in $\boldsymbol{Q}_{bi}^t$ related to VU i^* will then be set as ∞ in line 17 so that VU i^* will not be considered again in the next iteration. If the constraints of **P2** cannot be satisfied, H-URA will set the value of $\boldsymbol{Q}_{b^*i^*}^t$ as ∞ in line 19 and proceed to the next iteration. The iteration ends when all the VUs are associated with an SRSU or when all the elements in $\boldsymbol{Q}^t$ are ∞.

Note that in the worst case, the *while* loop will iterate ℓB times, which is the size of $\boldsymbol{Q}^t$. For each iteration, in the worst case, the time complexity of lines 5-8 is ℓB while the complexity

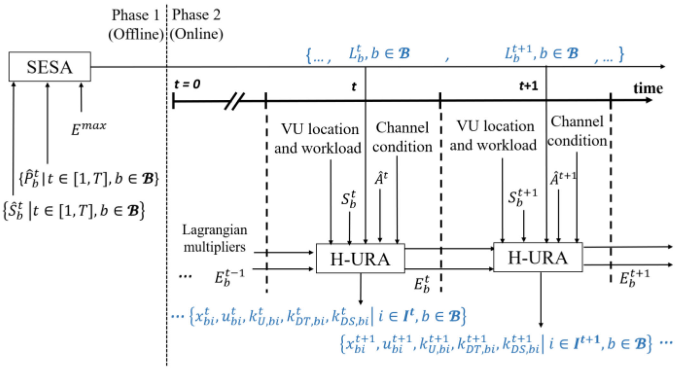

Fig. 4. Breakdown of TQMA algorithm.

Fig. 5. A neighborhood in Brooklyn, NY and SRSU deployment studied in this paper [42].

of other steps is less than or equal to ℓ. On the other hand, the complexity of updating $\hat{A}^t$ is less than O(ℓ^2). Therefore, the time complexity of H-URA is O($\ell^2 B^2$) for time slot t. Hence, H-URA is possible to be executed in real-time for reasonable sizes of the current VU set $\boldsymbol{I^t}$ and SRSU set $\mathcal{B}$. This is validated with experimental results reported in the next section.

By combining the proposed SESA and H-URA algorithms, we present our proposed heuristic method to solve **P1**, TQMA, as shown in Fig. 4. In Phase 1, SESA will schedule the solar energy for each time slot. Then, H-URA will be executed at each time slot to make user association the resource allocation decisions real-time in Phase 2.

V. Experimental Result

A. Simulation Framework

The objective of our simulation framework is to observe the weighted QoS loss performance of different solar energy scheduling, user association, and SRSU resource allocation strategies. In the simulation results below, we assume that VUs offload object detection tasks to SRSUs. In the meantime, some VUs will request to download videos as the *delay tolerant downlink data.* To simulate realistic VU movement and topology, we take a 1000∗800 (meters) rectangular area in Brooklyn, New York City, as shown in Fig. 5. We use historical vehicular traffic data in this area collected by New York State Department of Transportation [11]. Fig. 5 also shows the placement of 20 SRSUs used in our simulation environment.

We list the related simulation parameters in Table II. The duration of each time slot τ is 1 second. Because the duration of

TABLE II
Key Parameters in Simulation Framework

Parameter	Value	Parameter	Value
$K_{D,b}$	710	$p_{U,b}$	0.0067 W/subcarrier
$K_{U,b}$	710	$p_{D,b}$	0.0266 W/subcarrier
U_b	4744 MIPS	$p_{N,b}$	10 W
$p_{M,b}$	4.8 W	γ	0 dB
$p_{C,b}$	6.25 W	N_0	-174 dBm/Hz
p_b	30 dBm	E_b^0	0 Wh
p_i	23 dBm	E^{max}	600 Wh
Parameter	Value		
g_{bi}^t, h_{ib}^t	Pathloss and slow fading: Manhattan grid layout (B1) in [46] Fast fading: Nakagami-m distribution [47]		

the handover process in LTE-A can be less than 100 ms [43], we set $\kappa = 0.1$. Total simulation time is 24 hours, starting from 9 AM to include both day and night. Therefore, T is 86400.

At the beginning of each time slot, VUs enter the area from both ends of each street following a Poisson distribution with rate Θ. Each VU travels with predetermined route and speed. The travel route decision, speed, and Θ are set in a manner that the average traffic volume of each street satisfies the historical data in [11]. Furthermore, the channel model and the transmit power of SRSUs and VUs are listed in Table II [44], [45]. We set $A = 40$ for the next SRSU prediction.

To model the workload, we assume that each VU will upload an H.264 encoded video file with the data rate ω_i^t be uniformly distributed between 11 and 13.5 MB/s. It requires 10 million instructions per second (MIPS) as c_i^t for video processing, including decoding and object detection [1] at the MEC. We assume the size of the *delay sensitive downlink data* δ_i^t is uniformly distributed between 0.1 and 0.3 MB and the delay constraint d_i^t is 0.1s. In the meantime, VUs will have 0.25 probability to download a video file with size uniformly distributed between 7 and 9 MB as the *delay tolerant downlink data,* which has delay constraint $\theta_i^t = 1$s.

We model the downlink and uplink channel gains, g_{bi}^t and h_{ib}^t, by using Manhattan grid layout (B1) in [46] as the pathloss and slow fading, and the Nakagami-m distribution [47] as the fast fading, which have been widely used by the industry [44], [48] and are shown to be sufficient to model the vehicular communication channel [47].

The subcarriers are allocated to VUs in groups, and each group has 12 subcarriers (i.e., W = 180kHz/group) [49]. Multiple groups of subcarriers can be allocated to the same VU simultaneously. We assume each SRSU can utilize 710 subcarrier groups concurrently for each direction of transmission. To improve the inter-cell interference, we adopt the frequency reuse mapping technique [50] with reuse factor 3.

We model the MEC server of an SRSU by a Raspberry Pi 2 Model B [51], which is used to serve the offloaded workloads. Its corresponding computing resource and power consumption profiles are specified in Table II.

For the solar generation profile, we use the data collected at multiple sites in UC San Diego [10]. We normalize the solar energy data and assume the solar panel size is 1 m^2 for each SRSU. We use the proposed algorithm in [10] to predict solar generation profiles 24 hours in advance.

To compare against SESA, we use a best-effort technique, denoted as the Best effort Solar Energy scheduling Algorithm (BSEA). BSEA consists of a best-effort solar energy scheduling strategy and the same user association and SRSU resource allocation technique (H-URA) as TQMA. BSEA allows each SRSU to serve the associated VUs without constrained by the scheduled solar energy.

Another comparison is the Green energy and delay Aware User association and Resource Allocation (GAURA) algorithm proposed by [14]. GAURA is a combination of battery charging/discharging scheduling, SBS transmit power control, and user association algorithms, which is the closest approach to TQMA compared to other works. We assume GAURA follows the same way of H-URA to allocate subcarriers for uplink and the *delay tolerant downlink data* transmission. On the other hand, to fulfill the delay constraint in (6), we assume that GAURA will allocate $k^t_{DS,bi}$ downlink subcarriers and u^t_{bi} computing speed to VU i by the ratio: $u^t_{bi} = 4k^t_{DS,bi}$.

We also compare TQMA with our previous approach, QLM [1]. We assume that QLM has accurate predictions of VU's location and workload.

In the following sub-section, we will first present a performance comparison of our proposed TQMA with BSEA, GAURA, and QLM. Second, to show the efficiency of the Phase 2 algorithm, H-URA, a dynamic programming based Optimal User association and Resource allocation Algorithm (OPTA) [52] is implemented. Since [52] does not solve phase 1, we use the proposed SESA as the Phase 1 algorithm. We will compare the performance of TQMA and OPTA to show the efficiency of our proposed Phase 2 algorithm, H-URA. We introduce and analyze the complexity of OPTA in Appendix C. Third, to show the gap between the optimal solution and the proposed TQMA algorithm, we implement the exhaustive search method for **P1**. The complexity analysis of the exhaustive search method is listed in Appendix D. Finally, we will show the effect of solar energy prediction error on the performance of TQMA.

B. Simulation Results

We have implemented the proposed TQMA algorithm using MATLAB on a computer with a 3.8 GHz CPU, which is used to perform the offline battery scheduling and online user association and resource allocation for all the SRSUs in a neighborhood, like shown in Fig. 5. Note that a TQMA instance will be responsible for the SRSUs and the VUs of each such neighborhood. Since the battery scheduling algorithm SESA is run offline, we focus here on the run-time performance of H-URA. From our simulation-based experiments, the worst-case run-time of H-URA algorithm for a time slot is less than 180 ms. This is well below the time interval of 1s H-URA is executed (each time slot). Note that the input information (e.g., VU locations, workloads, and harvested solar) will not change dramatically during the 180 ms run-time of H-URA. Hence, we can conclude that H-URA is real-time, validating our time complexity based assertion in Section IV-B.

1) Performance Comparison of TQMA With Other Techniques: The weighted QoS loss performance of TQMA, BSEA, QLM, and GAURA are 0.125, 0.145, 0274, and 0.453, respectively. The performance of TQMA is the best compared to other techniques. To further discuss the effect of the above algorithms on individual VUs, we define *service outage time ratio* and *service disruption time ratio* for each VU as the following:

$$\textit{service outage time ratio} = \frac{\text{service outage time}}{\text{service request time}} \tag{32}$$

$$\textit{service disruption time ratio} = \frac{\text{service disruption time}}{\text{service request time}} \tag{33}$$

where the service outage time is the duration that this VU is experiencing the service outage, the service disruption time is the duration that this VU is experiencing the service disruption. The service request time is the duration that this VU is in the neighborhood and sending offloading demands.

In Fig. 6, we show the empirical cumulative distribution function (CDF) of the *service outage time ratio* and *service disruption time ratio* for the VUs. In Fig. 6(a), 86.2% of the VUs are served by the SRSUs for at least 80% of the service request time (*service outage time ratio* < 0.2) by using TQMA. On the contrary, 85.8%, 47%, and 40% of the VUs are served by SRSUs for at least 80% of their service request time by using BSEA, QLM, and GAURA algorithms, respectively. The performance of BSEA is close to TQMA because they share the same H-URA algorithm.

On the other hand, in Fig. 6(b), we can see that 85.7% of the VUs have less than 50% of their service request time experiencing the service disruption (the *service disruption time ratio* $<$ 0.5) by using TQMA. Compared to TQMA, 9.6%, 59.6%, and 90.1% of the VUs have the *service disruption time ratio* < 0.5 by using QLM, BSEA, and GAURA, respectively. QLM performs the worst because it will first consider associating a VU to the SRSU which provides the best signal strength, regardless of the VU's location, future movement, and the current associated SRSU. Compared to other algorithms, TQMA enables more VUs being served by SRSUs for longer duration while reducing their chances of experiencing service disruption.

Fig. 7 shows the weighted QoS loss performance comparison of the above algorithms under various system parameters (i.e., solar panel size, available computing speed, subcarrier groups, and battery capacity of SRSU). Fig. 7(a) shows the weighted QoS loss performance of these four algorithms under different solar energy availabilities, which are controlled by changing the solar panel size. TQMA has the best performance in terms of the weighted QoS loss among all the listed algorithms for different solar panel sizes. For instance, when the solar panel size equals 1 m^2, the performance of TQMA is 13.8% better than BSEA, 54.4% better than QLM, and 72.5% better than GAURA. The QoS loss of TQMA decreases while the solar panel size increases. However, the decrease starts to slow down and stops after the solar panel size exceeds 1.1 m^2. It is because the bottleneck of the performance becomes other limited resources after SRSU has enough solar energy.

From Fig. 7(b), we can observe that the weighted QoS loss decreases when the available number of subcarrier groups of each SRSU increases. Again, TQMA outperforms other algorithms.

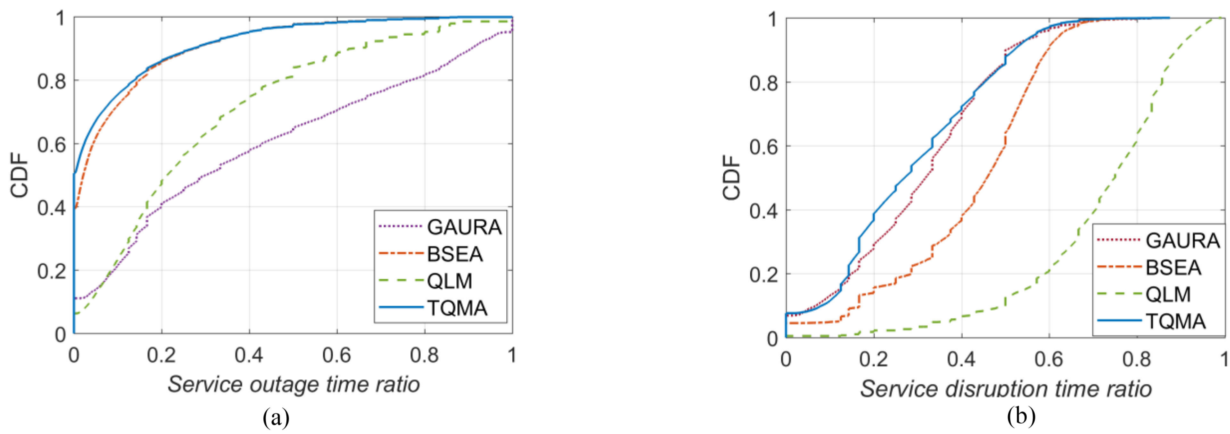

Fig. 6. The empirical cumulative distribution function of (a) left, the *service outage time ratio* and (b) right, the *service disruption time ratio* for individual VUs.

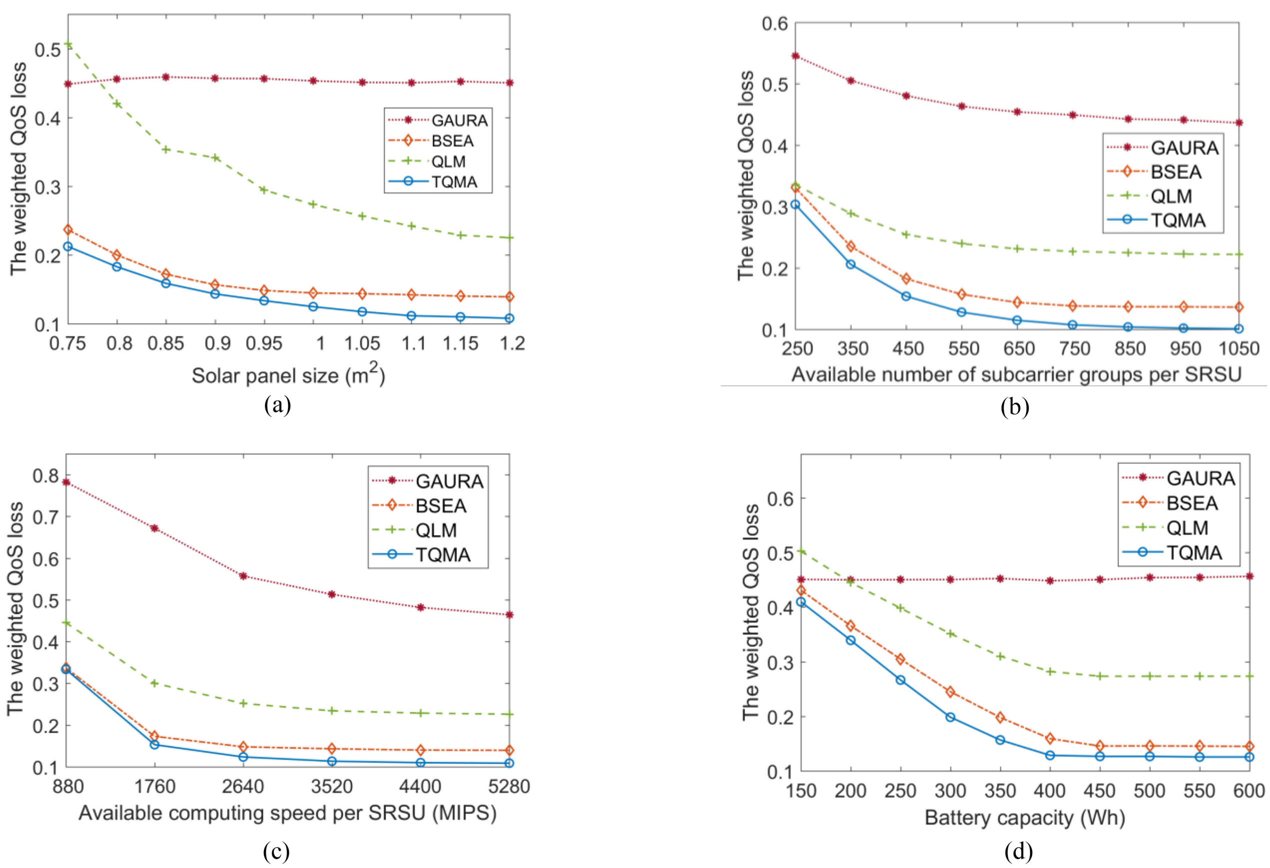

Fig. 7. The weighted QoS loss performance of various algorithms on (a) upper left, different solar panel sizes, (b) upper right, different available subcarrier groups of SRSU, (c) lower left, different available computing speeds of SRSU, and (d) lower right, different battery capacities of SRSU.

The performance gap between TQMA and the second-best algorithm, BSEA, grows with the number of subcarrier groups. The gap grows from 0.0273 to 0.0353 when the number of subcarrier groups increases from 250 to 1050, which shows that TQMA can more efficiently utilize these increased subcarrier resource.

In Fig. 7(c), the weighted QoS loss decreases when the available computing speed of each SRSU increases. Again, TQMA outperforms the other three algorithms under all conditions. Notice that the performance of TQMA improves slowly after the available computing speed exceeds 3520 MIPS. The weighted QoS loss only improves 0.0048 (i.e., 4%) from 3520 MIPS to 5280 MIPS. The performance of GAURA rises vastly in low available computing speed conditions, as its resource allocation mechanism (i.e., $u_{bi}^t = 4k_{DS,bi}^t$) will put a heavier burden on utilizing the computing speed than downlink subcarrier groups, especially in low available computing speed conditions.

In Fig. 7(d), the weighted QoS loss increases rapidly after the battery capacity decreases to a certain level. For TQMA, QLM,

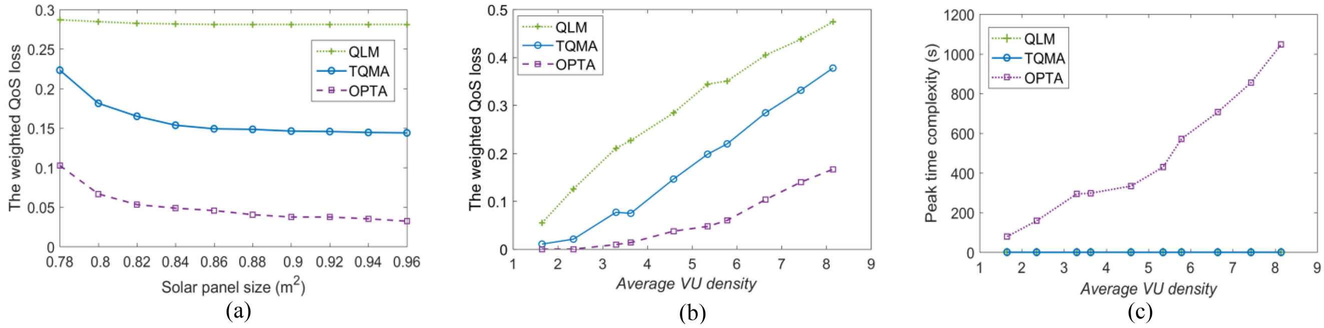

Fig. 8. (a) left, the weighted QoS loss performance of various algorithms on different solar panel sizes, (b) center, the weighted QoS loss performance of various algorithms on different *Average VU densities*, and (c) right, the peak time complexity of various algorithms on different *Average VU densities*.

and BSEA, we can observe that the critical point is 400 Wh. The weighted QoS loss starts to increase below this capacity because the capacity cannot fulfill the SRSU's power demand at night when there is no solar energy generated.

The results in Fig. 7 demonstrate the tradeoff between QoS and different resource availabilities, including solar panel sizes, battery capacities, MEC specifications, and configurations of SBS (subcarriers). This enables the service providers to identify what might be the best configurations of SRSU for expected solar generations and offloading demand profiles.

2) Performance Comparison With OPTA: In this comparison, we investigate the efficiency of our proposed Phase 2 algorithm, H-URA, by comparing TQMA to QLM and OPTA. To lower the complexity, we consider a smaller neighborhood surrounded by the dashed rectangle in Fig. 5. There are 2 SRSUs in this neighborhood and less than 14 VUs during peak hours. We equally divide the available computing speed into 5 levels and allocate them to each VU by levels. Subcarriers are divided into 5 groups. Fig. 8(a) shows the weighted QoS loss performance of QLM, TQMA, and OPTA when the solar panel size varies from 0.76 m^2 to 0.98 m^2. When the solar panel size is 0.9 m^2, the performance gap is 0.109 between TQMA and OPTA, while the gap between QLM and OPTA is 0.244. In terms of the peak time complexity (i.e., the recorded longest computation time for a time slot), TQMA takes 0.0938s while OPTA requires 333.5s when running on a 3.8 GHz CPU.

In Fig. 8(b), we present the weighted QoS loss performance of these 3 algorithms on different *average VU density* scenarios. The *average VU density* is calculated as $\sum_t^T |\boldsymbol{I}^t|/T$, where $\boldsymbol{I}^t$ is the VU set at the t^{th} time slot and T is the total number of time slots. We control the value of the *average VU density* by changing the vehicle generating rate Θ. In the meantime, Fig. 8(c) shows the corresponding peak time complexity. The gap between TQMA and OPTA increases linearly from 0.01 to 0.211 when the *average VU density* increases from 1.6 to 8.1. However, the corresponding peak time complexity of OPTA increases exponentially from 78.7s to 1047s. Although OPTA's dynamic programming Phase 2 algorithm provides promising QoS performance under different solar energy availability and *average VU density* conditions, it is prohibitively expensive in terms of time complexity. On the contrary, our proposed Phase 2 algorithm H-URA can keep the peak time complexity low for real-time decision making while compromising somewhat on optimal QoS performance though significantly better than QLM.

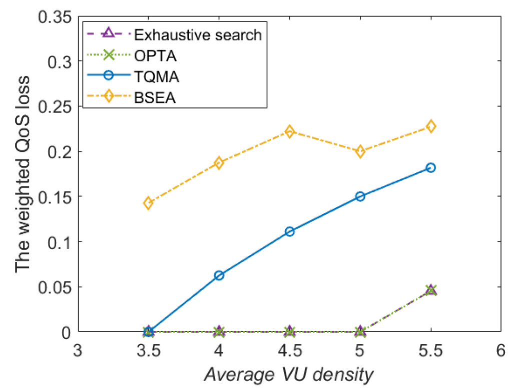

Fig. 9. The weighted QoS loss performance of TQMA, OPTA, and BSEA, compared with the optimal solution using exhaustive search.

3) Performance Comparison With Exhaustive Search: In this experiment, we investigate the efficiency of our proposed TQMA algorithm for solving **P1** by comparing with an exhaustive search method, which finds the optimal solution for **P1**. The exhaustive search method searches all the solar energy scheduling possibilities and uses dynamic programming algorithm (i.e. OPTA's Phase 2 algorithm) for user association and resource allocation for each solar energy scheduling possibility. Fig. 9 shows the performance comparison of BSEA, TQMA, OPTA, and the exhaustive search method. As shown in Appendix D, the complexity of the exhaustive search method is O$(TU^B {K_U}^B {K_D}^{B+1} \ell_{max}^2 B^2 T!^{\hat{S}})$, where ! is the factorial function, $\hat{S}$ is the maximum harvested solar energy of a time slot, and ℓ_{max} is the maximum number of VUs for a time slot. Due to the extremely high complexity, in this experiment we simulate only 4 time slots to represent a day (i.e., the gap between each slot is 6 hours). The granularity of the solar energy scheduling decision is 10 W. We consider the same neighborhood as in the previous subsection. Similar to the previous subsection, we control the value of the *average VU density* by changing the value of the vehicle generating rate Θ. We equally divide the available computing speed into 5 levels and allocate them to each VU by levels. The subcarriers are divided into 5 groups. Compared to BSEA, where no solar energy scheduling algorithm is implemented, TQMA's performance is closer to the optimal

TABLE III
PERFORMANCE WITH PREDICTION ERROR

	Solar Prediction Error			Performance		
Day #1	*MAE*	*MAPE(%)*	*RMSE*	*QoS loss*	*SO*[1]	*SD*[2]
Prediction Error	3.31	6.61%	5.73	12.5	8.74	37.5
No error	-	-	-	12.0	8.23	37.6
Day #2	*MAE*	*MAPE*	*RMSE*	*QoS loss*	*SD*[1]	*SO*[2]
Prediction error	8.63	49.8%	18.29	37.8	35.3	24.2
No error	-	-	-	37.2	34.8	25.1

[1]*SD: Service disruption rate (%)*, [2]*SO: Service outage rate (%)*

solution. The performance gap between TQMA and the optimal solution is 0.15 under regular traffic conditions (i.e. *average VU density* $= 5.0$). However, the peak time complexity of TQMA is 19.2ms, while the exhaustive search method requires 192,038s when running on a 3.8 GHz CPU. Therefore, finding the optimal solution is prohibitively expensive in terms of peak time complexity. To show their performances for high VU density scenarios, we increase Θ and create a 5.5 *average VU density* scenario. The weighted QoS loss gap between TQMA and the optimal solution is 0.14, which is almost the same as the gap when the *average VU density* is 5.0. But the peak time complexity of the exhaustive search method increases to 228,220s while TQMA only requires 20.3ms. Therefore, our proposed TQMA is more efficient in terms of both the peak time complexity and the weighted QoS loss.

To further investigate the cause of the performance gap between TQMA and the exhaustive search method, we include the performance of OPTA in Fig. 9. OPTA achieves the same weighted QoS loss as the optimal value. Because TQMA and OPTA share the same solar energy scheduling algorithm, the performance of OPTA shows that the gap between TQMA and the optimal value is due to the heuristic user association and resource allocation. Moreover, OPTA also demonstrates an approach for narrowing the performance gap without sacrificing largely on the time complexity. Its peak time complexity is 144.7s under regular traffic conditions, which is between TQMA (i.e. 19.2 ms) and the exhausted search method (i.e. 192038 s). Note that the performance of OPTA converges to the optimal value in Fig. 9 because this experiment is conducted under a limited-scale scenario. In fact, OPTA is not an optimal approach for **P1** as it considers only one solar energy scheduling possibility.

4) Effect of Prediction Error on TQMA: Finally, in this subsection, we present the effect of the prediction error of solar generation on the performance of TQMA. For each experiment, we run TQMA two times with the same simulation settings. For the first time, we use the predicted solar generation profile for SESA. The second time, we use the exact solar generation profile (no prediction error) for SESA.

The simulation results of two different days are shown in Table III, where *SD* is the *service disruption rate* and *SO* is the *service outage rate*. For day number 1, we observe prosperous and less intermittent solar generation since the weather is mostly sunny. Therefore, the prediction error is very small. We observe that its weighted QoS loss, *SD*, and *SO* are very similar with and without solar prediction error (compared to no prediction error). The weighted QoS loss of using solar prediction increases by 0.5(4.2%) compared to no prediction case. On the other hand, for day number 2, we observe poor and highly intermittent solar generation since the weather is partly sunny and partly cloudy. Consequently, the prediction error is worse than day number 1. The weighted QoS loss of using solar prediction increases by 0.6(1.6%) compared to no prediction error case. Its *SO* increases by 0.5% and *SD* drops by 0.9%. In this case, *SD* drops because *SO* increases. If a VU is experiencing service outage, it will not be counted as service disruption. Although the prediction error increases, the performance drop of TQMA in terms of the increased weighted QoS loss is still under 5%.

VI. CONCLUSION

In this paper, we propose a real-time QoS loss minimization algorithm to support the offloading of delay sensitive vehicular applications in a Solar-powered RSU network. The algorithm involves a two-phase approach: (i) the solar energy scheduling phase and (ii) the user association and resource allocation phase. SESA and H-URA respectively are developed for these two phases. A complete algorithm, TQMA, is proposed by integrating the above two algorithms which our simulation shows to significantly reduce the weighted QoS loss for the total operation time compared to existing techniques under various resource availabilities. The results help service providers and city planners to identify adequate SRSU configurations for expected solar energy generation and offloading demands.

Since solar power can be low due to weather conditions, our proposed approach cannot mitigate all risks of VUs experiencing high QoS loss alone. In future work, we plan to investigate the addition of other RE sources (e.g., wind energy) to ensure energy diversity and thus reduce risks to QoS loss in adverse weather conditions. Further, we plan to implement TQMA in a RE-powered road infrastructure prototype that will show the feasibility of the proposed algorithm for a sustainable SRSU network in a real-world scenario.

APPENDIX

A. Proof of Eq. 26

$$
\begin{aligned}
&C^t_{drop} + \ \kappa C^t_{handover}\\
&= \sum_{i \epsilon \boldsymbol{I}^t} \left(1 - \sum_{b \in \mathcal{B}} x^t_{bi}\right) + \kappa \sum_{i \epsilon \boldsymbol{I}^t} \left(\sum_{b \in \mathcal{B}} x^t_{bi}\right)\left(1 - \sum_{b \in \mathcal{B}} x^{t-1}_{bi} x^t_{bi}\right)\\
&= \ell^t - \sum_{i \epsilon \boldsymbol{I}^t} \sum_{b \in \mathcal{B}} x^t_{bi} + \kappa \sum_{i \epsilon \boldsymbol{I}^t} \left(\sum_{b \in \mathcal{B}} x^t_{bi} - \sum_{b \in \mathcal{B}} x^t_{bi} \Omega\left(x^t_{bi}, x^{t-1}_{bi}\right)\right)\\
&= \ell^t - \sum_{i \epsilon \boldsymbol{I}^t} \sum_{b \in \mathcal{B}} x^t_{bi} + \kappa \sum_{i \epsilon \boldsymbol{I}^t} \sum_{b \in \mathcal{B}} x^t_{bi} - \kappa \sum_{i \epsilon \boldsymbol{I}^t} \sum_{b \in \mathcal{B}} x^t_{bi} \Omega\left(x^t_{bi}, x^{t-1}_{bi}\right)\\
&= \ell^t + \sum_{i \epsilon \boldsymbol{I}^t} \sum_{b \in \mathcal{B}} \left(-1 + \kappa - \kappa \Omega\left(x^t_{bi}, x^{t-1}_{bi}\right)\right) x^t_{bi}.
\end{aligned}
$$

B. Proof of Lemma 1

With fixed Lagrangian multipliers $\lambda_b^t, \mu_b^t, \rho_b^t,$ and σ_b^t, $\mathbf{P2_{LD}}$ is reduced to:

$$\mathbf{P2'_{LD}}: \min_{x_{bi}^t, b\in\mathcal{B}, i\epsilon \boldsymbol{I}^t} \sum_{i\epsilon\boldsymbol{I}^t}\sum_{b\in\mathcal{B}} z_{bi}^t x_{bi}^t$$

$$+\sum_{b\in\mathcal{B}}\lambda_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t}x_{bi}^t v_{bi1}^t - U_b\right)+\sum_{b\in\mathcal{B}}\mu_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t}x_{bi}^t v_{bi2}^t - K_{U,b}\right)$$

$$+\sum_{b\in\mathcal{B}}\rho_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t}x_{bi}^t v_{bi3}^t - K_{D,b}\right)+\sum_{b\in\mathcal{B}}\sigma_b^t\left(\sum_{i\epsilon\boldsymbol{I}^t}x_{bi}^t v_{bi4}^t - L_b'^t\right)$$

s.t. (19)–(21),

The objective function of $\mathbf{P2'_{LD}}$ can then be rewritten as

$$\sum_{i\epsilon\boldsymbol{I}^t}\sum_{b\in\mathcal{B}} x_{bi}^t\left(z_{bi}^t+\lambda_b^t v_{bi1}^t+\mu_b^t v_{bi2}^t+\rho_b^t v_{bi3}^t+\sigma_b^t v_{bi4}^t\right)$$
$$-\sum_{b\in\mathcal{B}}\left(\lambda_b^t U_b+\mu_b^t K_{U,b}+\rho_b^t K_{D,b}+\sigma_b^t L_b'^t\right), \quad (34)$$

where the second term is a constant. Therefore, $\mathbf{P2'_{LD}}$ is equal to,

$$\mathbf{P2''_{LD}}: \min_{x_{bi}^t, b\in\mathcal{B}, i\epsilon\boldsymbol{I}^t}\sum_{i\epsilon\boldsymbol{I}^t}\sum_{b\in\mathcal{B}} x_{bi}^t q_{bi}^t$$

s.t. (19)–(21),

with $q_{bi}^t = z_{bi}^t + \lambda_b^t v_{bi1}^t + \mu_b^t v_{bi2}^t + \rho_b^t v_{bi3}^t + \sigma_b^t v_{bi4}^t$.

Note that q_{bi}^t and constraints (19)-(21) are separate for different VUs. Therefore, the optimal solution of $\mathbf{P2''_{LD}}$ (which is also the optimal solution of $\mathbf{P2_{LD}}$) will be finding the SRSU which minimizes q_{bi}^t under constraints (19)-(21) for each VU.

C. OPTA Algorithm

Since we have introduced SESA in Section IV-A, in this appendix, we analysis the complexity of OPTA's Phase 2 algorithm, which is based on dynamic programming. For a given instance of Phase 2, integers $i,\ n,\ \alpha_1,\ \ldots,\ \alpha_{3B}$, we use $f(i,n,\alpha_1,\ldots,\alpha_{3B})$ to represent the optimal value of **P2** with B SRSUs, which considers the VU set $\{1,2,\ldots,i\}\subseteq\boldsymbol{I}^t$ and allows at most n dropped VUs. Furthermore, each SRSU b utilizes exactly α_{3b-2} amount of computing speed, α_{3b-1} uplink subcarriers, and α_{3b} downlink subcarriers. To track the optimal user association and resource allocation decisions, we let $X(i,n,\alpha_1,\ldots,\alpha_{3B})$ and $\Psi(i,n,\alpha_1,\ldots,\alpha_{3B})$ be the corresponding user association and computing speed allocation of VU i for the instances $i,n,\alpha_1,\ldots,\alpha_{3B}$. We only track the allocation of computing speed because once we get x_{bi}^t from X, the optimal $k_{U,bi}^t, k_{DT,bi}^t$ can be derived by choosing the smallest possible values which satisfy workload constraints (4), (5). With the recorded u_{bi}^t in Ψ, we can calculate the optimal $k_{DS,bi}^t$ by delay constraint (6).

The core formula of OPTA is,

$$f\left(i,n,\alpha_1,\ldots,\alpha_{3B}\right)=$$
$$\begin{cases}\infty & if\ n<0\\ \infty & if\ \exists b\in\mathcal{B},\ \alpha_{3b-2}<0\ or\ \alpha_{3b-1}<0\ or\ \alpha_{3b}<0\\ 0 & if\ i=0,\ n\geq 0,\ \alpha_{3b-2}\geq 0,\ \alpha_{3b-1}\geq 0,\ \alpha_{3b}\geq 0,\\ & \quad \forall b\in\mathcal{B}\\ \infty & if\ \exists b\in\mathcal{B},\ P_b^t\left(\alpha_{3b-2},\alpha_{3b-1},\alpha_{3b}\right)<\ L_b'^t\\ min\left(A_1,A_2\right) & \text{otherwise}\end{cases} \quad (35)$$

where $P_b^t(\alpha_{3b-2},\alpha_{3b-1},\alpha_{3b})$ returns the corresponding power consumption of SRSU b for utilizing α_{3b-2} amount of computing speed, α_{3b-1} uplink subcarriers, and α_{3b} downlink subcarriers. $L_b'^t=\min(\ L_b^t, E_b^{t-1}+S_b^t)$ is for SRSU b to follow constraint (22). $A_1=1+f(i-1,n-1,\alpha_1,\ldots,\alpha_{3B})$ is the optimal value when choosing not to serve VU i. Finally, A_2 is the optimal value considering all possible values of $x_{bi}^t,\ u_{bi}^t\ b\in\mathcal{B}$ for VU i, and can be defined as,

$$A_2=\min_{b,x_{bi}^t,\ u_{bi}^t} z_{bi}^t+f\Big(i-1,n,\alpha_1,\ldots,\alpha_{3b-2}-u_{bi}^t,$$
$$\alpha_{3b-1}-k_{U,bi}^t,\alpha_{3b}-k_{DT,bi}^t-k_{DS,bi}^t,\ldots,\alpha_{3B}\Big) \quad (36)$$

with $k_{U,bi}^t$, $k_{DT,bi}^t$, and $k_{DS,bi}^t$ be the optimal numbers of uplink and downlink subcarriers correspond to x_{bi}^t and u_{bi}^t. Note that in (36), if $\eta_{D,bi}^t>\gamma$, $z_{bi}^t=-1+\kappa-\kappa\,\Omega(x_{bi}^t,x_{bi}^{t-1})$, otherwise $z_{bi}^t=\infty$.

f is initialized by an arbitrarily large value. X and Ψ are initialized as zero matrices. We recursively calculate the elements in f for i from 1 to ℓ, n from 1 to ℓ, α_{3b-2} from 1 to U_b, α_{3b-1} from 1 to $K_{U,b}$, α_{3b} from 1 to $K_{D,b}$, $\forall b\in\mathcal{B}$, until all the elements in f are updated. We record the corresponding optimal values of $x_{bi}^t$and u_{bi}^t in$X(i,n,\alpha_1,\ldots,\alpha_{3B})$ and $\Psi(i,n,\alpha_1,\ldots,\alpha_{3B})$, respectively. The optimal value of **P2** is then the smallest element in matrix $f(\ell,\ell,:,\ldots,:)$ (i.e., f with the specific indices, $i=\ell$, $n=\ell$, $1\leq\alpha_{3b-2}\leq U_b$, $1\leq\alpha_{3b-1}\leq K_{U,b}$, and $1\leq\alpha_{3b}\leq K_{D,b}\ \forall b\in\mathcal{B}$). We then calculate the optimal x_{bi}^t, u_{bi}^t, $k_{U,bi}^t$, $k_{DT,bi}^t$ and $k_{DS,bi}^t$ for VU i iteratively from $i=\ell$ to $i=1$, by using X,Ψ, and the indices correspond to the minimum element.

The time complexity of OPTA is O($U^B {K_U}^B {K_D}^{B+1}\ell^2 B^2$) if all the SRSUs have the same computing capacity U, number of uplink subcarriers K_U, and number of downlink subcarriers K_D. The complexity grows exponentially with the number of SRSUs in the network. Since the value of U, K_U, and K_D are usually very large, OPTA will be prohibitive in terms of run-time if there are more than 2 SRSUs in the network.

D. Complexity analysis of the exhaustive search method

Here we perform a complexity analysis of the exhaustive search method for **P1**. The optimal solution of **P1** requires the solar energy to be optimally scheduled to each time slot, while the VUs are associated with the optimal SRSU and SRSU resources are optimally allocated. For the sake of simplicity of analysis, we assume each SRSU has the same value of downlink subcarriers (i.e., K_D), uplink subcarriers (i.e., K_U), and computing capacity (i.e., U). By dynamic programming analysis in Appendix C, the complexity of the Phase 2 problem is $\mathrm{O}(U^B {K_U}^B {K_D}^{B+1} \ell^2 B^2)$ for each time slot, where B is the number of SRSU and ℓ is the current number of VU. On the other hand, since energy is continuous, there are unlimited possibilities of how many portions of the generated solar energy can be used in the current time slot and how the rest of it can be scheduled in the future time slots, so as to the energy stored in the battery. For simplicity, we assume the granularity of energy is 1 W and the maximum harvested solar energy for each time slot is $\hat{S}$. For the t^{th} time slot, because every 1W of the harvested solar energy can be scheduled to any time slot $t' \in [t, T]$, there are $\mathrm{O}((T-t+1)^{\hat{S}})$ scheduling possibilities. Therefore, for the overall operation time, there are $\mathrm{O}(\prod_{t=1}^{T}(T-t+1)^{\hat{S}}) = \mathrm{O}(T!^{\hat{S}})$ possible solar energy scheduling strategies will be searched, where !. is the factorial function. Consequently, with $\ell_{max} = \max_t \ell^t$, the complexity of exhaustively searching the optimal solution of **P1** is $\mathrm{O}(TU^B {K_U}^B {K_D}^{B+1} \ell_{max}^2 B^2 T!^{\hat{S}})$.

References

[1] Y. Ku, P. Chiang, and S. Dey, "Quality of service optimization for vehicular edge computing with solar-powered road side units," in *Proc. IEEE 27th Int. Conf. Comput., Commun. Netw.*, 2018, pp. 1–10.

[2] S. Olariu, and M. Weigle, *Vehicular Networks From Theory to Practice*. London, U.K.: Chapman&Hall, 2009.

[3] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[4] 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.

[5] M. H. Alsharif, J. Kim, and J. H. Kim, "Green and sustainable cellular base stations: An overview and future research directions," *Energies*, vol. 10, no. 5, p. 587, Apr. 2017.

[6] M. Z. Shakir, K. A. Qaraqe, H. Tabassum, M. Alouini, E. Serpedin, and M. A. Imran, "Green heterogeneous small-cell networks: Toward reducing the co_2 emissions of mobile communications industry using uplink power adaptation," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 52–61, Jun. 2013.

[7] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han,"5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[8] A. M. Aris and B. Shabani, "Sustainable power supply solutions for off-grid base stations," *Energies*, vol. 8, no. 10, pp. 10904–10941, Sep. 2015.

[9] P. H. Chiang *et al.*, "Renewable energy-aware video download in cellular networks," in *Proc. IEEE 26th Personal, Indoor Mobile Radio Commun.*, 2015, pp. 1622–1627.

[10] P. Chiang *et al.*, "Forecasting of solar photovoltaic system power generation using wavelet decomposition and bias-compensated random forest," in *Proc. 9th Annu. IEEE Green Technol. Conf.*, Mar. 2017, pp. 260–266.

[11] "NYS traffic data viewer," Department of Transportation, New York, NY, USA. Accessed: Mar. 6, 2018. [Online]. Available: https://www.dot.ny.gov/tdv

[12] L. X. Cai *et al.*, "Sustainability analysis and resource management for wireless mesh networks with renewable energy supplies," *IEEE J. Sel. Areas Commun.*, vol. 32, pp. 345–355, Feb. 2014.

[13] F. Parzysz, M. D. Renzo, and C. Verikoukis, "Power-availability-aware cell association for energy-harvesting small-cell base stations," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2409–2422, Apr. 2017.

[14] V. Chamola, B. Krishnamachari, and B. Sikdar, "Green energy and delay aware downlink power control and user association for off-grid solar-powered base stations," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2622–2633, Sep. 2018.

[15] C. Li *et al.*, "ISWITCH: Coordinating and optimizing renewable energy powered server clusters," in *Proc. 39th Annu. Int. Symp. Comput. Architecture*, 2012, pp. 512–523.

[16] Í. Goiri *et al.*, "Greenslot: Scheduling energy consumption in green datacenters," in *Proc. SC'11: Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2011, pp. 1–11.

[17] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[18] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, 2016, pp. 1–6.

[19] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[20] L. Chen, J. Xu and S. Zhou, "Computation peer offloading in mobile edge computing with energy budgets," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[21] H. Wu *et al.*, "Online geographical load balancing for energy-harvesting mobile edge computing," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[22] F. Guo, L. Ma, H. Zhang, H. Ji, and X. Li, "Joint load management and resource allocation in the energy harvesting powered small cell networks with mobile edge computing," in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Comput. Commun. Workshops*, 2018, pp. 299–304.

[23] M. J. Neely, "stochastic network optimization with application 1112 to communication and queueing systems," in *Synthesis Lectures on Communication Networks*, vol. 3. San Rafael, CA, USA: Morgan & Claypool, 2010, pp. 1–211.

[24] T. Yang, Z. Zheng, H. Liang, R. Deng, N. Cheng, and X. Shen, "Green energy and content-aware data transmissions in maritime wireless communication networks," *I EEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 751–762, Apr. 2015.

[25] W. S. Atoui *et al.*, "Offline and online scheduling algorithms for energy harvesting rsus in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6370–6382, Jul. 2018.

[26] W. S. Atoui *et al.*, "Scheduling energy harvesting roadside units in vehicular ad hoc networks," in *Proc. 2016 IEEE 84th Veh. Technol. Conf.*, 2016, pp. 1–5.

[27] X. Mao, A. Maaref and K. H. Teo, "Adaptive soft frequency reuse for inter-cell interference coordination in SC-FDMA based 3GPP lte uplinks," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–6.

[28] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tut.*, vol. 19, no. 4, pp. 2322–2358, Oct./Dec. 2017.

[29] M. Alasti, B. Neekzad, J. Hui and R. Vannithamby, "Quality of service in wimax and lte networks [topics in wireless communications]," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 104–111, May 2010.

[30] T. X. Tran *et al.*, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[31] N. T. Ti and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[32] Z. Xu, C. Yang, G. Y. Li, S. Zhang, Y. Chen, and S. Xu, "Energy-efficient configuration of spatial and frequency resources in MIMO-OFDMA systems," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 564–575, Feb. 2013.

[33] Bin Wu *et al.*, "Prediction of energy consumption time series using neural networks combined with exogenous series," in *Proc. IEEE 2015 11th Int. Center Nonviolent Conflict*, 2015, pp. 37–41.

[34] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013.

[35] X. Sun and N. Ansari, "Green cloudlet network: A distributed green mobile cloud network," *IEEE Netw.*, vol. 31, no. 1, pp. 64–70, Jan./Feb. 2017.

[36] B. Gavish, and H. Pirkul, "Algorithms for the multi-resource generalized assignment problem," *Manage. Sci.*, vol. 37, no. 6, pp. 695–713, Jun. 1991.

[37] M. Yagiura *et al.*, "A very large-scale neighborhood search algorithm for the multi-resource generalized assignment problem," *Discrete Optim.*, vol. 1, no. 1, pp. 87–98, 2004.

[38] S. Haddadi and H. Ouzia, "An effective lagrangian heuristic for the generalized assignment problem," *INFOR Inf. Syst. Oper. Res.*, vol. 37, pp. 351–356, 2001.

[39] C. J. Geyer, "markov chain monte carlo maximum likelihood," in *Proc. Comput. Sci. Statist.: Proc. 23rd Symp. Interface*, pp. 156–163, 1991.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*, New York, NY, USA: Cambridge University Press, 2004.

[41] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming," ver. 2.0 beta, Sep. 2013. [Online]. Available: http://cvxr.com/cvx

[42] Google. "New utrecht ave, *Brooklyn*" Accessed Dec. 10, 2019. [Online]. Available: https://www.google.com/maps/place/New+Utrecht+Ave,+Brooklyn,+NY/@40.6259898,-73.9983422,17z

[43] K. Alexandris *et al.*, "Analyzing X2 handover in LTE/LTE-A," in *Proc. 2016 14th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw.*, 2016, pp. 1–7.

[44] The 3rd Generation Partnership Project. "Study on LTE-based V2X services," 3GPP-REF-36885, rel. 14, 2016. [Online]. Available: http://www.3gpp.org/. Accessed: Jan. 19, 2018.

[45] The 3rd Generation Partnership Project. "Small cell enhancements for e-utra and e-utran - physical layer aspects," 3GPP-REF-36.872, rel. 12, 2013, Accessed on: Jan. 7, 2018. [Online]. Available: http://www.3gpp.org/

[46] P. Kyösti*et al.*, "Winner II channel models," *Eur. Commission*, Tech. Rep. IST-4-027756, Sep. 2007.

[47] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHZ dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.

[48] C. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Intervehicle transmission rate control for cooperative active safety system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 645–658, Sep. 2011.

[49] The 3rd Generation Partnership Project. "Etura; base station (BS) radio transmission and reception," 3GPP-REF-36.104, rel. 12, 2016. Accessed on: Apr. 10, 2019. [Online]. Available: http://www.3gpp.org/

[50] Z. Wang and R. A. Stirling-Gallacher, "Frequency reuse scheme for cellular ofdm systems," *Electron. Lett.*, vol. 38, no. 8, pp. 387–388, Apr. 2002.

[51] B. Benchoff, "Benchmarking the raspberry pi 2," Feb. 5, 2015. Accessed on: Apr. 11, 2019. [Online]. Available: https://www.raspberrypi.org/blog/benchmarking-raspberry-pi-2/

[52] W. Wu *et al.*, "Generalized assignment problem," in *Handbook of Approximation Algorithms and Metaheuristics*, 2nd ed. New York, NY, USA: Chapman&Hall/CRC, 2018, ch. 7, pp. 713–736.

Yu-Jen Ku (Student Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, in 2014 and the M.S. degree in electrical and computer engineering from the University of California, San Diego. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of California, San Diego. His research interests include green communication, mobile edge computing, and time series forecasting.

Po-Han Chiang (Student Member, IEEE) received the B.S. and M.S. degrees in electrical and communication engineering from National Taiwan University, in 2011 and 2013, respectively. He is working toward the Ph.D. degree in computer engineering at University of California at San Diego, La Jolla, CA, USA. His research interests include healthcare data mining, time series forecasting, stochastic optimization and applied machine learning. His paper received Best Paper Award in IEEE International Conference on E-health Networking, Application & Services (Healthcom'18).

Sujit Dey (Fellow, IEEE) received the Ph.D. degree in computer science from Duke University, in 1991. He is a Professor in the Department of Electrical and Computer Engineering, the Director of the Center for Wireless Communications, and the Director of the Institute for the Global Entrepreneur at University of California, San Diego. He heads the Mobile Systems Design Laboratory, developing innovative and sustainable edge computing, networking and communications, multi-modal sensor fusion, and deep learning algorithms and architectures to enable predictive personalized health, immersive multimedia, and smart transportation applications. He has created inter-disciplinary programs involving multiple UCSD schools as well as community, city and industry partners; notably the Connected Health Program in 2016 and the Smart Transportation Innovation Program in 2018. In 2017, he was appointed as an Adjunct Professor, Rady School of Management, and the Jacobs Family Endowed Chair in Engineering Management Leadership. Dr. Dey served as the Faculty Director of the von Liebig Entrepreneurism Center from 2013 to 2015, and as the Chief Scientist, Mobile Networks, at Allot Communications from 2012 to 2013. In 2015, he co-founded igrenEnergi, providing intelligent battery technology and solutions for EV mobility services. He founded Ortiva Wireless in 2004, where he served as its founding CEO and later as CTO and Chief Technologist till its acquisition by Allot Communications in 2012. Prior to Ortiva, he served as the Chair of the Advisory Board of Zyray Wireless till its acquisition by Broadcom in 2004, and as an advisor to multiple companies including ST Microelectronics and NEC. Prior to joining UCSD in 1997, he was a Senior Research Staff Member at NEC C&C Research Laboratories in Princeton, NJ. Dr. Dey has co-authored more than 250 publications, and a book on low-power design. He holds 18 U.S. and 2 international patents, resulting in multiple technology licensing and commercialization. He has been a recipient of nine IEEE/ACM Best Paper Awards, and has chaired multiple IEEE conferences and workshops.