Joint Task Scheduling and Energy Management for Heterogeneous Mobile Edge Computing With Hybrid Energy Supply

Ying Chen, *Member, IEEE*, Yongchao Zhang, Yuan Wu, *Senior Member, IEEE*, Lianyong Qi, *Member, IEEE*, Xin Chen, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Mobile edge computing (MEC) has recently become a promising paradigm to meet the increasing computing requirement of mobile devices, and hybrid energy supply has been considered as an effective approach for saving the energy consumption of the MEC system and making it environmentally friendly. In particular, the joint task scheduling and energy management (TSEM) scheme plays a crucial role in reaping the benefits of MEC with hybrid energy supply. In this article, we focus on jointly optimizing the TSEM decisions to maximize the utility of the MEC system which accounts for both the computation throughput and the fairness among different cells, by formulating a stochastic optimization problem subject to the constraints of queue stability and energy budget. We transform the formulated problem into a deterministic problem and then decouple it into four independent subproblems, which can be solved in a distributed manner without future system statistical information. An online TSEM algorithm is developed to derive the optimal solutions to these subproblems. Mathematical analysis shows that TSEM can achieve a close-to-optimal system utility and realize the *utility–queue* tradeoff. The experimental results validate the advantages of TSEM in improving the system utility and stabilizing the queue length.**

***Index Terms*—Energy management, hybrid energy supply, mobile edge computing (MEC), task scheduling.**

I. Introduction

With the proliferation of mobile devices and fast development of Internet of Things (IoT), more and more novel and complicated applications, which require heavy computation and high energy consumption, have been emerging in the past decades [1], [2]. However, due to the size and hardware constraints, the computing capacity and battery lifetime of mobile devices are usually limited, which makes them almost impractical to process these computation-intensive applications locally. To address this issue, mobile edge computing (MEC) is proposed as a promising paradigm by enabling computation-intensive tasks to be processed at network edges in close proximity to mobile devices [3]–[5]. In addition, to cope with the dramatic increase in data traffic and enhance the network coverage, a two-tier heterogeneous network called a small-cell network is proposed in recent years, where multiple low-power and low-cost small-cell base stations (SBSs) are deployed within one macrocell BS (MBS) [3]. Thus, a heterogeneous MEC, where MEC servers are deployed in both SBSs and MBSs [4], would play an important role in the next-generation wireless networks. SBSs could offload computation tasks that they cannot handle to the MBS for computing, which helps to improve the quality of experience.

At the same time, due to the continuing growth in the number of BSs, the energy consumption has witnessed a dramatic increase over the past years. In order to reduce the cost for purchasing energy as well as greenhouse gas emissions, energy harvesting (EH) technologies have been gaining more and more attention. Integrating with EH modules, BSs are able to collect renewable energies (such as wind and solar energies) from external environments. It is reported that EH can help to reduce 20% greenhouse gas emissions of cellular networks [5]. However, as the EH ability is largely influenced by many uncertain factors (e.g., weather and other climate conditions), it is extremely difficult to ensure steady and sufficient energy sources only with EH. Therefore, a hybrid energy supply, which includes both the renewable energy and steady energy from the power grid, is used to power BSs. If the harvested renewable energy cannot meet the energy demand, extra energy needs to be purchased from the power grid.

In the heterogeneous MEC with hybrid energy supply, a series of important decisions need to be made carefully for performance optimization. The first decision is how many computation tasks should be admitted in each SBS. As both the available energy and computing ability are limited, if too many computation tasks are admitted, it would cause

Manuscript received February 4, 2020; revised April 6, 2020; accepted April 21, 2020. Date of publication May 6, 2020; date of current version September 15, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61902029, Grant 61872044, and Grant 61872219; in part by the Excellent Talents Projects of Beijing under Grant 9111923401; in part by the Science and Technology Development Fund of Macau under Grant 0162/2019/A3; and in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF001. *(Corresponding author: Lianyong Qi.)*

Ying Chen, Yongchao Zhang, and Xin Chen are with the School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: chenying@bistu.edu.cn; zhangyongchao@mail.bistu.edu.cn; chenxin@bistu.edu.cn).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China (e-mail: yuanwu@um.edu.mo).

Lianyong Qi is with the School of Information Science and Engineering, Qufu Normal University, Qufu 276826, China (e-mail: lianyongqi@gmail.com).

Xuemin Shen is with the Department of Electronic and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2020.2992522

serious delay and performance degradation. The second one is how each SBS allocates the admitted computation tasks between processing locally and offloading to MBS. Compared with transmitting tasks to MBS for computing, processing tasks at local SBSs would help to save the overall energy, since there is no additional transmission energy consumption. While the computing capacity of SBSs is relatively restricted, only part of admitted computation tasks could be allocated to local SBSs; otherwise, there would be severe task congestion. Meanwhile, another decision to be made is the energy management, i.e., how much energy needs to be bought from the power grid to maintain BSs. The final decision is that owing to its limited computing capacity, MBS needs to decide how many computation tasks from each SBS would be computed.

Many factors, such as the task arrival, energy price, and wireless channel are all time varying. In such a context, optimizing the network performance in a long time scale is of practical importance. Therefore, we here focus on the long-term performance optimization. However, it is remarkably challenging in the face of the highly stochastic environment. This is because apart from the current network state, the future information of the above stochastic factors is also required to make the optimal decisions, while these factors are highly dynamic and random in reality. It is extremely difficult to acquire the statistical knowledge of these stochastic factors exactly. As a result, it is a challenging work to optimize the long-term task scheduling and energy management (TSEM) decisions without the system statistical information.

To address this challenge, we investigate the joint TSEM for MEC with hybrid energy supply. To model the dynamics in the task arrival, energy price, and wireless channel condition, a stochastic optimization problem is formulated. Considering the throughput and fairness metrics, we maximize the time-average system utility subject to the constraints of queue stability and time-average energy budget. By leveraging stochastic optimization techniques, we transform the original stochastic problem into a deterministic problem and decompose it into four independent subproblems. A TSEM algorithm without the future system statistical information is devised to obtain the optimal solutions to these subproblems in a distributed way. Mathematical analysis is conducted to validate the asymptotic optimality of TSEM. It shows that TSEM can realize the arbitrary *utility–queue* tradeoff by adjusting the value of the tradeoff parameter. The experimental results demonstrate the effectiveness of TSEM on improving the system utility and stabilizing queue.

The remainder of this article is organized as follows. Section II reviews the related works. In Section III, we give the system model, and a stochastic optimization problem is formulated to maximize the system utility. In Section IV, problem transformation is conducted, and an online algorithm, TSEM, is proposed to solve the optimization problem. Section V conducts the mathematical analysis for TSEM. Section VI presents the experimental results. In Section VII, this article is concluded, and the future direction of this article is discussed.

II. Related Work

As an important component in the next-generation wireless network, MEC has gained lots of research interests. This section elaborates the recent efforts spent on MEC and hybrid energy supply.

Sun *et al.* [6] investigated the mobility management strategy for the MEC-powered ultradense networks and proposed an optimization model to minimize the delay. You *et al.* [7] investigated the resource allocation in the MEC system with multiusers and tried to optimize the users' energy consumption with the constraint of delay. Two different radio access methods were considered in their work, which included time-division and orthogonal frequency-division multiple accesses. Wang *et al.* [8] focused on the task offloading and resource allocation in the MEC with wireless energy transfer. An energy consumption optimization problem was considered, and a semi-closed form solution to this problem was given. Lu *et al.* [9] applied deep reinforcement learning to make task offloading decision in large-scale heterogeneous MEC, where long short-term memory and candidate networks were explored to improve the traditional deep Q network algorithm. Neto *et al.* [10] investigated the computation offloading problem to optimize the computation time, energy consumption, and profit of edge computing facilities, whereas the above works all focused on the MEC networks whose energies only came from the power grid, they did not take into consideration of the renewable energy harvested from the environment, such as winds and sun.

There have been several studies investigating the MEC with EH technologies. Mao *et al.* [11] assumed that the mobile device was powered by an EH module and studied joint communication and computing resources management to optimize system energy consumption while satisfying queue stability. Min *et al.* [12] studied an IoT device selected the connected edge server and controlled offloading rate, and then a reinforcement-learning-based offloading strategy was devised to make these decisions. Different from the two works considering the mobile devices with EH, Guo *et al.* [13] focused on the content caching and base station association approach in a small-cell network where each base station was powered by the EH equipment. Wu *et al.* [14] jointly investigated the admission control and load balancing problem for MEC where renewable energy was the sole power supply. Chen *et al.* [15] focused on resource allocation for vehicular offloading in MEC, and the goal was to minimize the execution cost while guaranteeing task latency.

However, all the above works focused on the sole energy source, i.e., either the renewable energy or grid energy. As an important approach to save the on-grid energy consumption and make the system more sustainable, the hybrid energy supply needs to be considered. Thus, the MEC with hybrid energy supply is of practical importance and needs to be carefully investigated. To fill this gap, we focus on the heterogeneous MEC with hybrid energy supply and investigate the TSEM problem. Considering the throughput and fairness metrics, we devise an online algorithm to maximize the system utility, which is defined as a logarithmic function of the time-average amount of admitted data.

III. System Model

Consider a two-tier heterogeneous MEC system consisting of an MBS and N SBSs. These N SBSs are covered by the MBS and access the MBS via wireless communication channels [16]. In the MBS and each of SBS, a MEC server is deployed to process computation tasks offloaded from edge devices. Without loss of generality, the computing capacity of the MEC server in the MBS is usually much more powerful than those in the SBSs. Therefore, SBSs can transfer the computation tasks to the MBS, which helps to relieve the computing loads in SBSs. Inspired by [17], each SBS has two main energy sources, which are renewable energy (i.e., harvested from the environment through solar panels or wind turbines) and conventional grid energy. The energy in MBS is supplied by the conventional power grid. We consider a widely used time-slotted model, where $t \in \{0, 1, 2, \ldots\}$ [21]–[23]. The time slot length, referring to the duration between two adjacent decision moments, is denoted by τ.

A. Task and Computing Models

During slot t, each SBS will receive the computation tasks offloaded from network edge devices. For each SBS $i \in \{1, 2, \ldots, N\}$, let $A_i(t)$ (in bits) denote its amount of arrived computation tasks during slot t, and ρ_i denote the needed number of CPU cycles to process 1-bit input data. Without loss of generality, $A_i(t)$ has a peak value $A_i^{\max}$. Owing to the limited computing ability in each SBS, there would be a severe task backlog if all the computation tasks are allowed into the system. Therefore, only some of these computation tasks should be admitted and processed. Let $a_i(t)$ represent the amount of computation tasks which are admitted in SBS i. Since the admitted computation tasks cannot exceed the arrived ones, the following constraint must be satisfied:

$$0 \le a_i(t) \le A_i(t). \tag{1}$$

Let L_i denote the computing ability (i.e., the maximum available number of the CPU cycles) of the MEC server in SBS i. In each slot, for the computation tasks which have been admitted, some of them would be directly computed by the MEC servers in SBSs. In this article, the computation tasks can be arbitrarily divided, which corresponds to the applications like speech recognition and video compression [7], [18]. Let $s_i(t)$ represent the amount of computation tasks to be processed in SBS i. Naturally, SBS i should have enough computing capacity to process these computation tasks, which leads to the following constraint:

$$\rho_i s_i(t) \le L_i. \tag{2}$$

However, the MEC servers in the SBSs usually cannot handle all the admitted computation tasks because of their limited computing abilities. Therefore, in addition to processing tasks on its own MEC server, each SBS also transfers some computation tasks to the MBS. MBS then helps SBSs to compute these computation tasks. For each SBS i, let $m_i(t)$ denote the amount of computation tasks to be transmitted to MBS. Similar to [19], an orthogonal frequency-division multiple access system is adopted, in which the wireless channel allocated to each SBS is orthogonal to others. Let P_i represent the transmit power of SBS i, and $h_i(t)$ denote the channel power gain from SBS i to MBS. Then, the maximum achievable transmission rate of SBS i can be obtained by [20]

$$R_i(t) = B \log_2\left(1 + \frac{P_i h_i(t)}{B\sigma}\right) \tag{3}$$

where σ is the noise power spectrum density and B is the wireless channel bandwidth. Since the amount of transmitted tasks cannot exceed the wireless channel capacity, the following constraint must be satisfied:

$$m_i(t) \le R_i(t)\tau. \tag{4}$$

In each SBS, the admitted but not yet processed tasks will be stored in a task queue. At the beginning of slot t, let $Q_i^D(t)$ represent the length (i.e., the amount of unprocessed tasks) of the task queue in SBS i. During every slot, some computation tasks would be taken from the task queue and then processed by the MEC servers. Meanwhile, some new computation tasks would also arrive. Thus, the queue length of each SBS evolves as the following equality:

$$Q_i^D(t+1) = \max\left\{Q_i^D(t) - s_i(t) - m_i(t), 0\right\} + a_i(t). \tag{5}$$

For the MBS, it maintains N task queues to store the computation tasks transmitted from each SBS but not yet processed. In each slot, some computation tasks in these N queues would be fetched and computed by the MEC server in MBS. Let $x_i(t)$ denote the amount of computation tasks to be processed in the i-th queue.[1] Let $Q_i^M(t)$ represent the queue length of the i-th queue in MBS

$$Q_i^M(t+1) = \max\left\{Q_i^M(t) - x_i(t), 0\right\} + m_i(t). \tag{6}$$

The total computation tasks to be processed should be within the computing capacity of the MBS. Let L_0 denote the maximum computing ability of MBS. We thus have

$$\sum_{i=1}^{N} \rho_i x_i(t) \le L_0. \tag{7}$$

In addition, we use $\bar{Q}_i^D(t)$ to denote the average queue length of SBS i in a long time, which can be expressed by

$$\bar{Q}_i^D = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{Q_i^D(t)\right\}. \tag{8}$$

Similarly, let $\bar{Q}_i^M(t)$ denote the average length of the i-th queue in MBS. We have

$$\bar{Q}_i^M = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{Q_i^M(t)\right\}. \tag{9}$$

[1] In the MBS, the i-th queue refers to the queue storing computation tasks transmitted from SBS i.

B. Energy Harvesting and Consumption Models

For each SBS, its energy source includes two parts: 1) the harvested renewable energy and 2) the grid energy. Let $e_i(t)$ denote the amount of harvested energy in SBS i during slot t. When too many computation tasks are admitted or the EH ability becomes very poor, the harvested energy would be unable to afford the operations of the SBSs. In this case, SBSs will purchase the grid energy. Let $w_i(t)$ denote the amount of purchased grid energy. Without loss of generality, in each slot t, there exists a maximum value on $w_i(t)$ [5], which is

$$w_i(t) \leq w_i^{\max}. \tag{10}$$

Let $g(t)$ denote the price of unit grid energy, which may fluctuate across the time [21]. Then, the payment in SBS i for purchasing grid energy is $C_i(t) = g(t)w_i(t)$.

For each SBS, we consider that its energy consumption consists of two main parts: 1) computing energy consumption and 2) transmission energy consumption. Then, let $E_i^l(t)$ represent the energy consumption incurred by MEC server processing computation tasks, which is as follows [8]:

$$E_i^l(t) = \alpha_i \cdot s_i(t) \tag{11}$$

where α_i denotes the energy consumption coefficient of the MEC server in SBS i, and the value of α_i depends on the chip architecture and other factors. In addition, let $E_i^M(t)$ represent the energy consumed by data transmission from SBS i to MBS, which can be obtained by

$$E_i^m(t) = P_i \cdot \frac{m_i(t)}{R_i(t)}. \tag{12}$$

Each SBS has a sufficiently large battery [5], which is used to store and supply the energy. At the beginning of slot t, let $Q_i^E(t)$ represent the available energy in the battery of SBS i. In this article, similar to [7] and [8], we assume that the energy consumption used for sending the computation results back to the mobile devices is negligible. Thus, we do not take this part of energy consumption into account. For each SBS, to ensure that its energy consumption does not exceed the available energy, the following constraint must be satisfied:

$$\alpha_i \cdot s_i(t) + P_i \cdot \frac{m_i(t)}{R_i(t)} \leq Q_i^E(t) + w_i(t). \tag{13}$$

During each slot, the harvested energy and purchased energy from the power grid are stored in the batteries. Meanwhile, a part of stored energy is consumed to process and transmit tasks. Then, for the adjacent two slots, $Q_i^E(t)$ evolves as the following equality:

$$Q_i^E(t+1) = Q_i^E(t) + e_i(t) + w_i(t) - E_i^m(t) - E_i^l(t). \tag{14}$$

For the MBS, it will process the computation tasks transmitted from all the SBSs. Let $E_0(t)$ denote the energy consumption of the MBS, which is

$$E_0(t) = \alpha_0 \cdot \sum_{i=1}^{N} x_i(t) \tag{15}$$

where α_0 is the energy consumption coefficient of the MEC server in MBS. Then, according to (15), the payment in MBS for purchasing grid energy $C_0(t)$ is

$$C_0(t) = g(t)\alpha_0 \sum_{i=1}^{N} x_i(t). \tag{16}$$

C. Optimization Problem

This article studies the TSEM problem for MEC with hybrid energy supply. Rather than considering the system performance in a single moment, we pay attention to the long-term system performance considering dynamic task arrival, wireless channel, and energy price. Motivated by [22], there exists an energy budget on the long-term time-average payment for purchasing energy from the power grid. Let $\bar{J}$ denote the energy budget. Then, the following inequality should be satisfied:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\{g(t)(w_i(t) + \alpha_0 x_i(t))\} \leq \bar{J}. \tag{17}$$

Besides, we take into consideration of the time-average queueing delay for the computation tasks. According to [23], the time-average queueing delay is positively proportional to time-average queue length. Thus, we upper bound the queue length to ensure that the queueing delay is bounded, which is

$$\bar{Q}_i^D \leq \varepsilon, \quad \exists\ \varepsilon > 0 \tag{18}$$

$$\bar{Q}_i^M \leq \delta, \quad \exists\ \delta > 0. \tag{19}$$

In this article, the objective function accounts for two factors, namely, improving the network throughput and keeping the fairness among the SBSs. Specifically, the network throughput is important yet conventional performance metric of the considered multicell systems. However, due to the dynamics and diversity in the task arrival, EH ability, and wireless channel condition among different SBSs, the throughput of some SBSs may be much higher than others, which means that the performance of some SBSs would be largely degraded. Therefore, in order to guarantee the network throughput of each SBS, we also take into consideration the fairness among SBSs. In such a context, we define the system utility function as a logarithmic function of the time-average admitted data, which is [24]

$$\psi(\bar{\mathbf{a}}) = \sum_{i=1}^{N} \log(1 + \bar{a}_i). \tag{20}$$

In (20), $\bar{\mathbf{a}} = \{\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_N\}$ denotes the collection of the time-average admitted data of all the N SBSs, and $\bar{a}_i = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1} a_i(t)$. It is noted that $\psi(\bar{\mathbf{a}})$ is concave and strictly nondecreasing.

We here aim to maximize the system utility while guaranteeing the payment for purchasing energy from power grid does not exceed its budget. Therefore, we formulate the following problem:

$$\mathbf{P:} \max_{\mathbf{a}(t),\mathbf{s}(t),\mathbf{m}(t),\mathbf{w}(t)} \ \psi(\bar{\mathbf{a}}), \quad \text{s.t.} \quad (1),\ (2),\ (4),\ (7),\ (10),\ (13),\ (17)\text{–}(19). \tag{21}$$

The formulated problem **P** is a stochastic problem, since the optimization goal and constraints (17) and (18) depend on not only the current system states but also the future ones, such as the task arrival, wireless channel, and energy price. In addition, these variables are typically stochastic and time varying.

IV. Algorithm Design

For the stochastic optimization problem **P**, we can find that deriving its offline solution is an extremely difficult job, because this process requires the future system information (such as task arrival and wireless channel). However, this information varies over time and is hard to predict precisely. Therefore, it is of great challenge to optimize the TSEM decisions to adapt to a dynamic environment. In such a context, we take advantage of the Lyapunov optimization techniques to address this challenge. Based on the Lyapunov function, we transform the long-term stochastic optimization problem into a deterministic problem under each time slot. Then, we propose an online algorithm which makes the decisions based on only the current information. One of the key advantages of using the Lyapunov function is that it requires no statistical knowledge about the future system information and only relies on the current information for decision making.

A. Problem Transformation

The function in **P**, which is a logarithmic function of the time-average variables, is difficult to be maximized directly. To tackle this difficulty, we introduce an *auxiliary variable* $u_i(t)$ and equally simplify the original optimization function. Thus, **P** can be transformed to the following problem:

$$\begin{aligned}\mathbf{P1}: \max_{\mathbf{a}(t),\mathbf{s}(t),\mathbf{m}(t),\mathbf{w}(t),\mathbf{u}(t)} \quad & \sum_{i=1}^{N}\bar{\psi}(u_i(t)),\\ \text{s.t.} \quad & (1),\ (2),\ (4),\ (7),\ (10),\ (13),\ (17)-(19)\\ & \bar{u}_i \le \bar{a}_i \quad \forall i \in \{1,2,\ldots,N\} \qquad (22)\\ & 0 \le u_i(t) \le A_i^{\max} \quad \forall i \in \{1,2,\ldots,N\}\end{aligned} \tag{23}$$

where $\bar{\psi}(u_i(t)) = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}\log(1+u_i(t))$ and $\bar{u}_i = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}u_i(t)$. Since the system utility function is concave and nondecreasing, the original problem **P** and the transformed problem **P1** can be proved to be equivalent by using Jensen's inequality [25]. The detailed process is omitted here for simplicity.

Following the framework of Lyapunov optimization, by using a virtual queue, constraint (17) can be transformed into a queue stability problem, which makes **P1** more tractable. Therefore, we define a *virtual queue* $K(t)$

$$K(t+1) = \max\left\{K(t) + \sum_{i=1}^{N} g(t)(w_i(t)+\alpha_0 x_i(t)) - \bar{J}, 0\right\}. \tag{24}$$

For queue $K(t)$, if it is stable, i.e., $\lim_{t\to\infty}\mathbb{E}\{K(t)\}/t = 0$, it can be proved that constraint (17) must hold.

Theorem 1: If $\lim_{t\to\infty}\mathbb{E}\{K(t)\}/t = 0$, constraint $\lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\{g(t)(w_i(t)+\alpha_0 x_i(t))\} \le \bar{J}$ can be satisfied.

Proof: According to (24), we have

$$\begin{aligned}K(t+1) &= \max\left\{K(t) + \sum_{i=1}^{N} g(t)(w_i(t)+\alpha_i x_i(t)) - \bar{J}, 0\right\}\\ &\ge K(t) + \sum_{i=1}^{N} g(t)(w_i(t)+\alpha_0 x_i(t)) - \bar{J}.\end{aligned} \tag{25}$$

By summing (25) over time and further dividing by t, we have

$$\frac{K(t)}{t} \ge \frac{K(0)}{t} + \frac{1}{t}\sum_{\mu=0}^{t-1}\sum_{i=1}^{N} g(\mu)(w_i(\mu)+\alpha_0 x_i(\mu)) - \bar{J}. \tag{26}$$

Rearranging (26) and taking expectations, we have

$$\begin{aligned}&\frac{\mathbb{E}\{K(t)\} - K(0)}{t}\\ &\quad\ge \frac{1}{t}\sum_{\mu=0}^{t-1}\sum_{i=1}^{N}\mathbb{E}\{g(\mu)(w_i(\mu)+\alpha_0 x_i(\mu))\} - \bar{J}.\end{aligned} \tag{27}$$

Without loss of generality, $K(0) = 0$. Letting $t \to \infty$, it yields

$$\lim_{t\to\infty}\frac{\mathbb{E}\{K(t)\}}{t} + \bar{J} \ge \lim_{t\to\infty}\frac{1}{t}\sum_{\mu=0}^{t-1}\sum_{i=1}^{N}\mathbb{E}\{g(\mu)(w_i(\mu)+\alpha_0 x_i(\mu))\}. \tag{28}$$

From (28), it can be derived that if queue $K(t)$ is stable, i.e., $\lim_{t\to\infty}[(\mathbb{E}\{K(t)\})/t] = 0$, there exists $\bar{J} \ge \lim_{t\to\infty}(1/t)\sum_{\mu=0}^{t-1}\sum_{i=1}^{N}\mathbb{E}\{g(\mu)(w_i(\mu)+\alpha_0 x_i(\mu))\}$, which means that constraint (17) is satisfied. ■

Similarly, another *virtual queue* $Z_i(t)$ is defined to reform constraint (22) as a queue stability problem. Specifically, $Z_i(t)$ updates as

$$Z_i(t+1) = \max\{Z_i(t) + u_i(t) - a_i(t), 0\}. \tag{29}$$

Then, according to the Lyapunov optimization technique, let $\mathbf{\Theta}(t) = [\mathbf{Q}^D(t), \mathbf{Q}^M(t), K(t), \mathbf{Z}(t)]$ denote the vector of all the queue lengths. We define a Lyapunov function $L(\mathbf{\Theta}(t))$

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\left\{K^2(t) + \sum_{i=1}^{N}\left[\left(Q_i^D(t)\right)^2 + \left(Q_i^M(t)\right)^2 + Z_i^2(t)\right]\right\}. \tag{30}$$

In (30), the value of $L(\mathbf{\Theta}(t))$ can reflect the queue backlog state. Based on (30), we define a *Lyapunov drift* function

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}. \tag{31}$$

Recall that the objective of **P1** is to maximize the system utility. Thus, we define a *drift-minus-utility* function $\Psi_V(\mathbf{\Theta}(t))$

$$\Psi_V(\mathbf{\Theta}(t)) = \Delta(\mathbf{\Theta}(t)) - V\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))|\mathbf{\Theta}(t)\} \tag{32}$$

in which $V > 0$ is a parameter adjusting the tradeoff between system utility and queue length. A larger V emphasizes more

improving system utility than maintaining queue length stable. According to the Lyapunov optimization framework, instead of minimizing $\Psi_V(\mathbf{\Theta}(t))$ directly, we minimize its upper bound. We first derive the supremum bound of $\Psi_V(\mathbf{\Theta}(t))$ in the following theorem.

Theorem 2: If $R_i^{\max} \geq R_i(t)$ and $g^{\max} \geq g(t)$, then the value of $\Psi_V(\mathbf{\Theta}(t))$ satisfies

$$\begin{aligned}\Psi_V(\mathbf{\Theta}(t)) \leq H - V\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))|\mathbf{\Theta}(t)\} \\ +\sum_{i=1}^{N}Z_i(t)\mathbb{E}\{u_i(t)-a_i(t)|\mathbf{\Theta}(t)\} \\ +\sum_{i=1}^{N}Q_i^M(t)\mathbb{E}\{m_i(t)-x_i(t)|\mathbf{\Theta}(t)\} \\ +\sum_{i=1}^{N}Q_i^D(t)\mathbb{E}\{a_i(t)-s_i(t)-m_i(t)|\mathbf{\Theta}(t)\} \\ +K(t)\mathbb{E}\left\{\sum_{i=1}^{N}g(t)(w_i(t)+\alpha_0 x_i(t))-\bar{J}|\mathbf{\Theta}(t)\right\}\end{aligned} \tag{33}$$

where H is a constant. Specifically

$$\begin{aligned}H = \frac{1}{2}\Bigg\{\left(\frac{L_i^{\max}}{\rho_i}+R_i^{\max}\tau\right)^2+3\left(A_i^{\max}\right)^2+\left(\frac{L_i^{\max}}{\rho_i}\right)^2+\left(R_i^{\max}\tau\right)^2 \\ +\left[\sum_{i=1}^{N}g^{\max}\left(w_i^{\max}+\alpha_0\frac{L_i^{\max}}{\rho_i}\right)\right]^2+\bar{J}^2\Bigg\}.\end{aligned}$$

Proof: Squaring (5), we obtain

$$\begin{aligned}\left(Q_i^D(t+1)\right)^2-\left(Q_i^D(t)\right)^2 \leq (s_i(t)+m_i(t))^2+a_i^2(t) \\ +2Q_i^D(t)(a_i(t)-s_i(t)-m_i(t)).\end{aligned} \tag{34}$$

Similarly, according to (6), we have

$$\begin{aligned}\left(Q_i^M(t+1)\right)^2-\left(Q_i^M(t)\right)^2 \leq x_i^2(t)+m_i^2(t) \\ +2Q_i^M(t)(m_i(t)-x_i(t)).\end{aligned} \tag{35}$$

In addition, according to (29), we can obtain

$$Z_i^2(t+1)-Z_i^2(t) \leq u_i^2(t)+a_i^2(t)+2Z_i(t)(u_i(t)-a_i(t)). \tag{36}$$

Based on (24), the following equation can be obtained:

$$\begin{aligned}K^2(t+1)-K^2(t) \leq \left[\sum_{i=1}^{N}g(t)(w_i(t)+\alpha_0 x_i(t))\right]^2+\bar{J}^2+2K(t) \\ \times\left[\sum_{i=1}^{N}g(t)(w_i(t)+\alpha_0 x_i(t))-\bar{J}\right].\end{aligned} \tag{37}$$

Since $R_i(t) \leq R_i^{\max}$, we can have $m_i(t) \leq R_i^{\max}\tau$. In addition, according to $a_i(t)$, $u_i(t) \leq A_i^{\max}$, and $g(t) \leq g^{\max}$, it can be obtained

$$\begin{aligned}\mathbb{E}\Bigg\{(s_i(t)+m_i(t))^2+2a_i^2(t)+u_i^2(t)+x_i^2(t) \\ +m_i^2(t)+\left[\sum_{i=1}^{N}g(t)(w_i(t)+\alpha_0 x_i(t))\right]^2+\bar{J}^2|\mathbf{\Theta}(t)\Bigg\} \\ \leq \left(\frac{L_i}{\rho_i}+R_i^{\max}\tau\right)^2+3\left(A_i^{\max}\right)^2+\left(\frac{L_0}{\rho_i}\right)^2 \\ +\left(R_i^{\max}\tau\right)^2+\left[\sum_{i=1}^{N}g^{\max}\left(w_i^{\max}+\alpha_0\frac{L_0}{\rho_i}\right)\right]^2+\bar{J}^2.\end{aligned} \tag{38}$$

Then, according to (38), summing (34), (36), and (37), and adding $-V\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))|\mathbf{\Theta}(t)\}$, we can obtain (33). ■

B. Optimal Task Scheduling and Energy Management Algorithm

An optimal TSEM algorithm is devised to minimize the upper bound of $\Psi_V(\mathbf{\Theta}(t))$ in Theorem 2. The upper bound minimization problem is decomposed into four independent subproblems, and these subproblems can be solved in a distributed manner. Besides, it can be observed that the optimal solutions of different SBSs are also independent. Thus, TSEM could make the optimal decisions of all the SBSs concurrently.

As H, $\mathbf{\Theta}(t)$, and $K(t)$ are constants in each slot, the above upper bound minimization problem can be rewritten as

$$\begin{aligned}\mathbf{P2}:\ \min\ &\sum_{i=1}^{N}(Z_i(t)u_i(t)-V\log(1+u_i(t))) \\ &+\sum_{i=1}^{N}\left(Q_i^D(t)-Z_i(t)\right)a_i(t) \\ &+\sum_{i=1}^{N}\Big[\left(Q_i^M(t)-Q_i^D(t)\right)m_i(t)-Q_i^D(t)s_i(t) \\ &\qquad +K(t)g(t)w_i(t)\Big] \\ &+\sum_{i=1}^{N}\left(\alpha_0 K(t)g(t)-Q_i^M(t)\right)x_i(t) \\ \text{s.t.}\ &(1),\ (2),\ (4),\ (7),\ (10),\ (13),\ (23).\end{aligned} \tag{39}$$

Then, **P2** is decoupled into four independent subproblems: 1) auxiliary variable selection (AUS); 2) admission control decision (ACD); 3) TSEM in SBS (AES); and 4) task allocation in MBS (TAM). Next, we derive the optimal solutions to these four subproblems separately.

1) Auxiliary Variable Selection: Considering the first term in the objective of **P2** and constraint (23), the **AUS** subproblem is formulated as

$$\begin{aligned}\mathbf{AUS}:\ \min_{u_i(t)}\ &Z_i(t)u_i(t)-V\log(1+u_i(t)) \\ \text{s.t.}\ &0 \leq u_i(t) \leq A_i^{\max}.\end{aligned} \tag{40}$$

AUS is a convex optimization problem, which can be proved via the second-order derivation. Next, we drive the optimal solution to **AUS**.

For (40), we can obtain that its first-order derivation is $Z_i(t) - (V/[\ln 2(1+u_i(t))])$. Thus, when $u_i(t) = [V/(Z_i(t)\ln 2)] - 1$, $Z_i(t) - (V/[\ln 2(1+u_i(t))]) = 0$. Then, the optimal solution to **AUS** is

$$u_i^*(t) = \begin{cases} 0, & \frac{V}{Z_i(t)\ln 2} - 1 < 0 \\ \frac{V}{Z_i(t)\ln 2} - 1, & 0 \le \frac{V}{Z_i(t)\ln 2} - 1 < A_i^{\max} \\ A_i^{\max}, & \text{otherwise.} \end{cases} \tag{41}$$

2) Admission Control Decision: Considering the second term in the objective of **P2** and constraint (1), the **ACD** subproblem is formulated as

$$\begin{aligned} \textbf{ACD}: \min_{a_i(t)} \ & \big(Q_i^D(t) - Z_i(t)\big)a_i(t) \\ \text{s.t.} \ & 0 \le a_i(t) \le A_i(t). \end{aligned} \tag{42}$$

It can be seen that **ACD** is a simple linear programming problem. Its optimal solution is

$$a_i^*(t) = \begin{cases} A_i(t), & Q_i^D(t) - Z_i(t) \le 0 \\ 0, & \text{otherwise.} \end{cases} \tag{43}$$

In (43), $Q_i^D(t) - Z_i(t)$ can be regarded as the *admission cost* of each SBS. To reduce the *admission cost* as much as possible, when the *admission cost* of one SBS is less than 0, all the computation tasks in this SBS would be admitted. On the contrary, the SBS whose *admission cost* is larger than 0 is not allowed to admit any computation tasks. Moreover, with the rise of queue length [i.e., $Q_i^D(t)$], the *admission cost* also rises. This is a reasonable result because there would be a serious task congestion if the SBS with a larger queue length continues to admit computation tasks.

3) Task Scheduling and Energy Management in SBS: Considering the third term in the objective of **P2** and constraints (2), (4), (10), and (13), the **AES** subproblem is formulated as

$$\begin{aligned} \textbf{AES}: \min_{s_i(t),m_i(t),w_i(t)} \ & \big(Q_i^M(t) - Q_i^D(t)\big)m_i(t) \\ & - Q_i^D(t)s_i(t) + K(t)g(t)w_i(t) \\ \text{s.t.} \ & \rho_i s_i(t) \le L_i, \quad m_i(t) \le R_i(t)\tau \\ & w_i(t) \le w_i^{\max} \\ & \alpha_i \cdot s_i(t) + P_i \cdot \frac{m_i(t)}{R_i(t)} \le Q_i^E(t) + w_i(t). \end{aligned} \tag{44}$$

To derive the optimal solutions to **AES**, we transform it into the following problem:

$$\begin{aligned} \textbf{AES-1}: \min_{s_i(t),m_i(t),w_i(t)} \ & w_i(t) - \frac{Q_i^D(t)}{K(t)g(t)\alpha_i}\alpha_i s_i(t), \\ & - \frac{\big(Q_i^D(t) - Q_i^M(t)\big)R_i(t)}{K(t)g(t)P_i} \cdot \frac{P_i m_i(t)}{R_i(t)} \\ \text{s.t.} \ & \rho_i s_i(t) \le L_i, \quad m_i(t) \le R_i(t)\tau \\ & w_i(t) \le w_i^{\max} \\ & - Q_i^E(t) \le w_i(t) - \alpha_i s_i(t) - \frac{P_i m_i(t)}{R_i(t)}. \end{aligned} \tag{45}$$

It can be observed that **AES-1** is a linear programming problem. Next, we give its optimal solutions in the different cases.

1) When $(Q_i^D(t))/(K(t)g(t)\alpha_i) \le 1$ and $((Q_i^D(t) - Q_i^M(t))R_i(t))/(K(t)g(t)P_i) \le 1$, the optimal solutions are $s_i^*(t) = m_i^*(t) = w_i^*(t) = 0$.
2) When $((Q_i^D(t) - Q_i^M(t))R_i(t))/(K(t)g(t)P_i) \le 1 < (Q_i^D(t))/(K(t)g(t)\alpha_i)$, the optimal $m_i(t)$ is $m_i^*(t) = 0$:
   a) if $w_i^{\max} \le ((\alpha_i L_i)/\rho_i) - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, and the optimal $s_i(t)$ is $s_i^*(t) = (w_i^{\max} + Q_i^E(t))/\alpha_i$;
   b) if $w_i^{\max} > (\alpha_i L_i/\rho_i) - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = (\alpha_i L_i/\rho_i) - Q_i^E(t)$, and the optimal $s_i(t)$ is $s_i^*(t) = (L_i/\rho_i)$.
3) When $(Q_i^D(t))/(K(t)g(t)\alpha_i) \le 1 < ((Q_i^D(t) - Q_i^M(t))R_i(t))/(K(t)g(t)P_i)$, the optimal $s_i(t)$ is $s_i^*(t) = 0$:
   a) if $w_i^{\max} \le P_i\tau - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, and the optimal $m_i(t)$ is $m_i^*(t) = ((w_i^{\max} + Q_i^E(t))R_i(t))/P_i$;
   b) if $w_i^{\max} > P_i\tau - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = (R_i(t)P_i\tau)/(R_i(t)) - Q_i^E(t)$, and the optimal $m_i(t)$ is $m_i^*(t) = R_i(t)\tau$.
4) When $1 < (Q_i^D(t))/(K(t)g(t)\alpha_i) \le ((Q_i^D(t) - Q_i^M(t))R_i(t))/(K(t)g(t)P_i)$:
   a) if $w_i^{\max} \le P_i\tau - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, the optimal $m_i(t)$ is $m_i^*(t) = ((w_i^{\max} + Q_i^E(t))R_i(t))/P_i$ and the optimal $s_i(t)$ is $s_i^*(t) = 0$;
   b) if $P_i\tau - Q_i^E(t) < w_i^{\max} \le P_i\tau - Q_i^E(t) + (\alpha_i L_i/\rho_i)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, the optimal $m_i(t)$ is $m_i^*(t) = R_i(t)\tau$, and the optimal $s_i(t)$ is $s_i^*(t) = (w_i^{\max} + Q_i^E(t) - P_i\tau)/\alpha_i$;
   c) if $P_i\tau - Q_i^E(t) + (\alpha_i L_i/\rho_i) < w_i^{\max}$, the optimal $w_i(t)$ is $w_i^*(t) = P_i\tau + (\alpha_i L_i/\rho_i) - Q_i^E(t)$, the optimal $m_i(t)$ is $m_i^*(t) = R_i(t)\tau$, and the optimal $s_i(t)$ is $s_i^*(t) = (L_i/\rho_i)$.
5) When $1 < ((Q_i^D(t) - Q_i^M(t))R_i(t))/(K(t)g(t)P_i) < (Q_i^D(t))/(K(t)g(t)\alpha_i)$:
   a) if $w_i^{\max} \le (\alpha_i L_i/\rho_i) - Q_i^E(t)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, the optimal $s_i(t)$ is $s_i^*(t) = (w_i^{\max} + Q_i^E(t))/\alpha_i$, and the optimal $m_i(t)$ is $m_i^*(t) = 0$;
   b) if $(\alpha_i L_i/\rho_i) - Q_i^E(t) < w_i^{\max} \le P_i\tau - Q_i^E(t) + (\alpha_i L_i/\rho_i)$, the optimal $w_i(t)$ is $w_i^*(t) = w_i^{\max}$, the optimal $s_i(t)$ is $s_i^*(t) = (L_i/\rho_i)$, and the optimal $m_i(t)$ is $m_i^*(t) = (w_i^{\max} - (\alpha_i L_i/\rho_i) + Q_i^E(t))(R_i(t)/P_i)$;
   c) if $P_i\tau - Q_i^E(t) + (\alpha_i L_i/\rho_i) < w_i^{\max}$, the optimal $w_i(t)$ is $w_i^*(t) = P_i\tau + (\alpha_i L_i/\rho_i) - Q_i^E(t)$, the optimal $s_i(t)$ is $s_i^*(t) = (L_i/\rho_i)$, and the optimal $m_i(t)$ is $m_i^*(t) = R_i(t)\tau$.

4) Task Allocation in MBS: Considering the last term in the objective of **P2** and constraint (7), the **TAM** subproblem is formulated as follows:

$$\textbf{TAM}: \min_{x_i(t)} \ \sum_{i=1}^{N} \big(\alpha_0 K(t)g(t) - Q_i^M(t)\big)x_i(t)$$

Algorithm 1 TSEM Algorithm

1: Get all queues' length, i.e., $Q_i^D(t)$ and $Q_i^M(t)$.
2: For each SBS, obtain optimal $u_i^*(t)$ and $a_i^*(t)$ based on (41) and (43), respectively.
3: For each SBS, solve (45) and obtain the optimal $s_i^*(t)$, $m_i^*(t)$, and $w_i^*(t)$.
4: For MBS, obtain the optimal $x_i^*(t)$ according to (47).

$$\text{s.t.} \sum_{i=1}^{N} \rho_i x_i(t) \le L_0. \tag{46}$$

It can be observed that **TAM** is a min-weight problem, where $x_i(t)$ is weighted by $\alpha_0 K(t) g(t) - Q_i^M(t)$. Thus, the optimal solution to **TAM** is

$$x_i^*(t) = \begin{cases} \frac{L_0}{\rho_i}, & \alpha_0 K(t) g(t) - Q_{i^*}^M(t) < 0 \text{ and } i = i^* \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

where

$$i^* = \arg \min_{i=\{1,2,\ldots,N\}} \frac{\alpha_0 K(t) g(t) - Q_i^M(t)}{\rho_i}.$$

Then, Algorithm 1 shows the proposed optimal algorithm (TSEM) in detail.

Next, we give the time complexity of TSEM. In Algorithm 1, for lines 1 and 2, the optimal solutions can be obtained independently by each SBS. Thus, the time complexity is $O(1)$. For line 3, MBS would traverse each queue once in the worst case, and it will terminate within $O(N)$ operations. Thus, the time complexity of TSEM is $O(N)$, where N is the number of SBSs.

V. Algorithm Analysis

We mathematically analyze the performance of TSEM. Specifically, we prove that the system utility of TSEM can approximate the optimum arbitrarily, and there exists a supremum bound for the system queue length.

We give Lemma 1 to analyze the performance of TSEM.

Lemma 1: For any task arrival rate $\boldsymbol{\lambda}$, there always exists an optimal policy π_1, which is independent with the queue length. Besides, π_1 satisfies the following conditions:

$$\begin{aligned} &\mathbf{E}\left\{\sum_{i=1}^{N} \bar{\psi}(u_i(t))|\pi_1\right\} = \sum_{i=1}^{N} \bar{\psi}_i^*(\boldsymbol{\lambda}) \\ &\mathbf{E}\{u_i(t)|\pi_1\} \le \mathbf{E}\{a_i(t)|\pi_1\} \\ &\mathbf{E}\{m_i(t)|\pi_1\} \le \mathbf{E}\{x_i(t)|\pi_1\} \\ &\mathbf{E}\{a_i(t)|\pi_1\} \le \mathbf{E}\{s_i(t) + m_i(t)|\pi_1\} \\ &\mathbf{E}\left\{\sum_{i=1}^{N} g(t)(w_i(t) + \alpha_0 x_i(t))|\pi_1\right\} \le \bar{J} \end{aligned}$$

where $\sum_{i=1}^{N} \bar{\psi}_i^*(\boldsymbol{\lambda})$ is the maximum system utility with $\boldsymbol{\lambda}$.

Proof: We can prove Lemma 1 with Caratheodory's theorem [25]. However, the details are omitted here for the sake of brevity. ∎

Note that there exist an upper bound $\hat{\psi}$ and a lower bound $\check{\psi}$ for the system utility. Define $\bar{Q} = \lim_{T\to\infty}(1/T)\sum_{t=0}^{T-1}\sum_{i=1}^{N}[Q_i^D(t) + Q_i^M(t)]$. Based on Lemma 1, we drive Theorem 3 to analyze the performance of TSEM.

Theorem 3: When the task arrival rate is $\boldsymbol{\lambda} + \varsigma$, under TSEM, we can obtain

$$\sum_{i=1}^{N} \bar{\psi}_i^* - \sum_{i=1}^{N} \bar{\psi}_i^{\text{TSEM}} \le \frac{H}{V} \tag{48}$$

$$\bar{Q} \le \frac{H + V\left(\hat{\psi} - \check{\psi}\right)}{\varsigma} \tag{49}$$

where H is the constant in (33).

Proof: Recall that TSEM can minimize the *drift-minus-utility*. Then, for any policy π, we have

$$\begin{aligned} &\Delta(\boldsymbol{\Theta}(t)) - V\sum_{i=1}^{N} \mathbb{E}\{\log(1 + u_i(t))|\boldsymbol{\Theta}(t)\} \\ &\le H - V\sum_{i=1}^{N} \mathbb{E}\{\log(1 + u_i(t))|\boldsymbol{\Theta}(t), \pi\} \\ &\quad + \sum_{i=1}^{N} Z_i(t)\mathbb{E}\{u_i(t) - a_i(t)|\boldsymbol{\Theta}(t), \pi\} \\ &\quad + \sum_{i=1}^{N} Q_i^M(t)\mathbb{E}\{m_i(t) - x_i(t)|\boldsymbol{\Theta}(t), \pi\} \\ &\quad + \sum_{i=1}^{N} Q_i^D(t)\mathbb{E}\{a_i(t) - s_i(t) - m_i(t)|\boldsymbol{\Theta}(t), \pi\} \\ &\quad + K(t)\mathbb{E}\left\{\sum_{i=1}^{N} g(t)(w_i(t) + \alpha_0 x_i(t)) - \bar{J}|\boldsymbol{\Theta}(t), \pi\right\}. \end{aligned} \tag{50}$$

Then, based on Lemma 1, we can derive that there exists an optimal policy π_2 satisfying

$$\begin{aligned} &\mathbf{E}\left\{\sum_{i=1}^{N} \bar{\psi}(u_i(t))|\pi_2\right\} = \sum_{i=1}^{N} \bar{\psi}_i^*(\boldsymbol{\lambda} + \varsigma) \\ &\mathbf{E}\{u_i(t)|\pi_2\} \le \mathbf{E}\{a_i(t)|\pi_2\} - \varsigma \\ &\mathbf{E}\{m_i(t)|\pi_2\} \le \mathbf{E}\{x_i(t)|\pi_2\} - \varsigma \\ &\mathbf{E}\{a_i(t)|\pi_2\} \le \mathbf{E}\{s_i(t) + m_i(t)|\pi_2\} - \varsigma \\ &\mathbf{E}\left\{\sum_{i=1}^{N} g(t)(w_i(t) + \alpha_0 x_i(t))|\pi_2\right\} \le \bar{J} - \varsigma. \end{aligned} \tag{51}$$

Substituting (51) into (50), it yields

$$\begin{aligned} &\Delta(\boldsymbol{\Theta}(t)) - V\sum_{i=1}^{N} \mathbb{E}\{\log(1 + u_i(t))|\boldsymbol{\Theta}(t)\} \\ &\le H - V\sum_{i=1}^{N} \bar{\psi}_i^*(\boldsymbol{\lambda} + \varsigma) - \varsigma\sum_{i=1}^{N} Z_i(t) - \varsigma\sum_{i=1}^{N} Q_i^M(t) \\ &\quad - \varsigma\sum_{i=1}^{N} Q_i^D(t) - \varsigma K(t). \end{aligned} \tag{52}$$

Taking expectations on (52) and summing over time, we obtain

$$\begin{aligned}&\mathbf{E}\{L(\Theta(T))\}-\mathbf{E}\{L(\Theta(0))\}-V\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))\}\\&\le HT-VT\sum_{i=1}^{N}\bar{\psi}_i^*(\boldsymbol{\lambda}+\varsigma)-\varsigma\sum_{t=0}^{T-1}\sum_{i=1}^{N}Z_i(t)\\&\quad-\varsigma\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^M(t)-\varsigma\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^D(t)-\varsigma\sum_{t=0}^{T-1}K(t).\end{aligned}\tag{53}$$

Without loss of generality, $L(\Theta(0))=0$ and $L(\Theta(0))\ge 0$. Thus, dividing (53) by T, we can obtain

$$\begin{aligned}&-V\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))\}\\&\le H-V\sum_{i=1}^{N}\bar{\psi}_i^*(\boldsymbol{\lambda}+\varsigma)-\varsigma\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}Z_i(t)\\&\quad-\varsigma\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^M(t)-\varsigma\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^D(t)-\varsigma\sum_{t=0}^{T-1}K(t).\end{aligned}\tag{54}$$

Letting $\varsigma\to 0$ and $T\to\infty$, dividing by V, we can obtain (48) in Theorem 3.

Rearranging (54), we obtain

$$\begin{aligned}&\varsigma\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^M(t)+\varsigma\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}Q_i^D(t)\\&\le H+V\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\mathbb{E}\{\log(1+u_i(t))\}-V\sum_{i=1}^{N}\bar{\psi}_i^*(\boldsymbol{\lambda}+\varsigma)\\&\le H+V\left(\hat{\psi}-\check{\psi}\right).\end{aligned}\tag{55}$$

Taking expectations and dividing (55) by ς, letting $T\to\infty$, we have (49) in Theorem 3. ■

In Theorem 3, it is proved that the gap between the result of TSEM and the optimal value of the original problem is upper bounded. Equation (48) demonstrates that the gap between the system utility of TSEM and the optimal result is upper bounded by H/V. If V is sufficiently large, the system utility of TSEM can approximate the optimal value. In addition, in (49), the average queue length of TSEM is proved to be bounded. This shows that TSEM can always make all the queues stable. Therefore, TSEM is also able to guarantee the queueing delay of the tasks in all SBSs. Besides, combining with (48) and (49), TSEM can realize a tradeoff between system utility and queue length by tuning V.

VI. Evaluation

We conduct the experiments to evaluate the performance of TSEM. In the experiments, we consider a heterogeneous MEC network consisting of 15 SBSs and 1 MBS. The transmission power P_i of each SBS is uniformly distributed in [5, 15] W [26]. Similar to [27], the task arrival rate $A_i(t)$ is randomly generated in [1, 8] Mb, and the required CPU number to compute 1-bit task of each SBS $\phi_i\sim U[1000, 23\,000]$ cycles/b. The computing capacity of SBS is $U[8,\ 10]$ GHz, and that of MBS is 20 GHz. A Rayleigh fading channel is considered, in which the channel gain $h_i(t)$ is the exponential distribution with unit mean [11]. Furthermore, the EH rate $e_i(t)$ is randomly generated in an interval [0, 200] J, and the price of power grid energy follows a folded normal distribution $\mathcal{N}(3, 3)$ [5]. The energy consumption coefficient of MEC servers $\alpha_i=10^{-4}$ J/b [8], and the energy budget $\bar{J}$ is 3500. Besides, $B=10$ MHz [28], $\sigma=10^{-10}$ W, and $\tau=1$ s. The time length of the experiments is 3000 s.

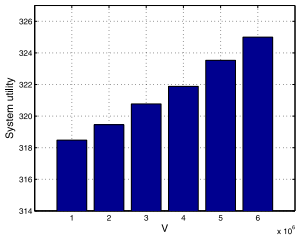

Fig. 1. System utility with different V.

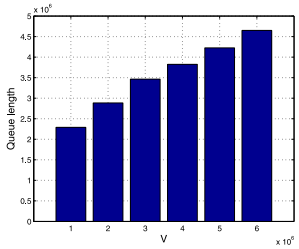

Fig. 2. Queue length with different V.

A. Parameter Analysis

1) Impact of Tradeoff Parameter: Fig. 1 shows the average system utility of TSEM with different V. As the tradeoff parameter V increases, the system utility also increases. This result is consistent with the conclusion obtained by (48) in Theorem 3, where the larger value of V brings the higher system utility. It is because setting a large V means that the system utility is emphasized more than the queue length. TSEM would dynamically tune task scheduling decisions to improve the system utility. In Fig. 2, we show the queue length of TSEM with different V. In the experiments, the queue length refers to the average amount of unprocessed tasks in all queues (in bit). We can see that with the rise of V, the queue length also grows. This agrees with the conclusion obtained by (49) in Theorem 3, where increasing V would lead to a larger queue length. Together with Figs. 1 and 2, it can be observed that TSEM can achieve an arbitrary *utility–queue* tradeoff by changing the value of the tradeoff parameter V.

2) Impact of Task Arrival Rate: Fig. 3 shows the average system utility of TSEM with different task arrival rates. In the experiments, the task arrival rate is set to $\vartheta\cdot A_i(t)$. Specifically, three different settings of ϑ are considered, i.e., $\vartheta=0.8$, 1, and 1.2. From Fig. 3, it can be seen that the system utility rises with the increase of the task arrival rate. It is because

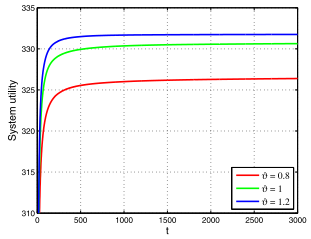

Fig. 3. System utility with different task arrival rates.

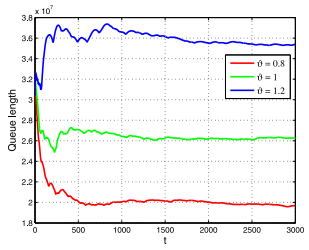

Fig. 4. Queue length with different task arrival rates.

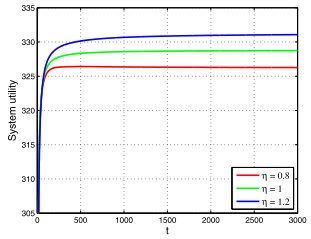

Fig. 5. System utility with different MEC computing capacities.

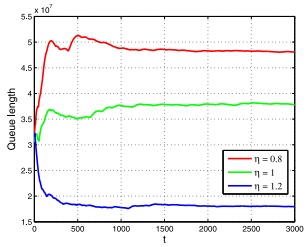

Fig. 6. Queue length with different MEC computing capacities.

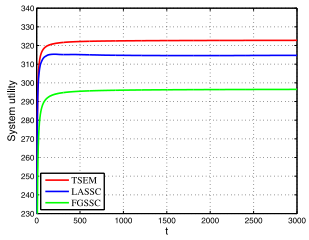

Fig. 7. System utility with the three different algorithms.

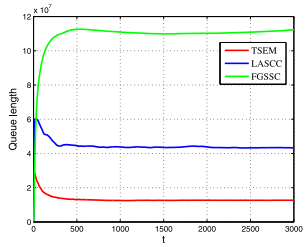

Fig. 8. Queue length with the three different algorithms.

if the task arrival rate becomes larger, TSEM would admit more computation tasks. Fig. 4 shows the queue length of TSEM with different task arrival rates. When the task arrival rate rises, the queue length also rises. However, we can see that TSEM can always stabilize the queue length. Combining Figs. 3 and 4, it can be seen that TSEM can adapt to the different task arrival rates and improve the system utility while still guaranteeing the queue stability.

3) Impact of MEC Computing Capacity: Fig. 5 shows the system utility of TSEM with different MEC computing capacities. In the experiments, the MEC computing capacity of SBSs is set to $\eta \cdot L_i$. Similarly, three different settings of η are considered, i.e., $\eta = 0.8$, 1, and 1.2. From Fig. 5, we can observe that when the MEC computing capacity becomes larger, the system utility would rise. This is because, with the increase of MEC computing capacity, the processing ability of the MEC system also rises. In this case, TSEM would increase the amount of admitted computation tasks, which brings the larger system utility. Fig. 6 shows the queue length of TSEM with different MEC computing capacities. As expected, with MEC computing capacity rises, the queue length would reduce. It is because the system processing ability becomes larger when the computing capacity increases. Consequently, the queue length would decrease. From Figs. 5 and 6, we can see that with the rise of MEC computing capacity, TSEM can improve the system utility and stabilize the queue length.

B. Comparison Experiment

We further compare the system utility and queue length of TSEM with two benchmark algorithms. The two benchmark algorithms are listed as follows.

1) *Lya-Based Algorithm With Single-Slot Constraint (LASSC) [22]:* Instead of considering the time-average energy constraint (17) in the long term, this energy budget constraint is satisfied in every single slot. Then, Lyapunov optimization techniques are applied to solve the optimization problem with this single-slot constraint.
2) *Fair Greed With Single-Slot Constraint (FGSSC):* Inspired by [24], each SBS is prioritized by $1/\sum_{t=0}^{t-1} a_i(t)$, and the computation tasks from a subset of SBSs are admitted according to the obtained priority. In addition, in order to save as much energy as possible, computation tasks prefer to be processed at the local SBSs.

Fig. 7 shows the system utility with the three different algorithms. It can be observed that the system utility of TSEM is the largest among the three algorithms, and the system utility of LASSC is larger than that of FGSSC. It is because that TSEM is able to dynamically tune the TSEM decisions according to the dynamic task arrival, wireless channel as well as energy price. However, LASSC does not take the dynamics in the energy price into account when making energy management decision, and FGSSC makes decisions only based on

the current system information. Therefore, TSEM can outperform LASSC and FGSSC. Fig. 8 shows the queue length with three different algorithms. We can see that the queue length of TSEM is shorter than those of LASSC and FGSSC, and stabilizes quickly. In addition, the queue length of LASSC is shorter than that of FGSSC. From Figs. 7 and 8, we can see the superiorities of TSEM on improving the system utility and stabilizing the queue length.

VII. Conclusion

In this article, we have studied the TSEM problem in heterogeneous MEC with hybrid energy supply. We have formulated it as a stochastic optimization problem aiming at maximizing the average system utility while keeping the queue stable and meeting the energy budget. An online and distributed algorithm called TSEM has been proposed to derive the optimal solutions. The experimental results showed that TSEM could achieve a high system utility while keeping the queue length stable. For our future work, we would adopt the techniques of deep reinforcement learning to address the TSEM problem. In addition, the transmission interference among different SBSs would be considered in our future work.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[2] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[3] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 675–687, Mar. 2017.

[4] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.

[5] D. Zhang *et al.*, "Resource allocation for green cloud radio access networks with hybrid energy supplies," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1684–1697, Feb. 2018.

[6] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[7] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[8] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[9] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," *Future Gener. Comput. Syst.*, vol. 102, pp. 847–861, Jan. 2020.

[10] J. L. D. Neto, S. Yu, D. F. Macedo, J. M. S. Nogueira, R. Langar, and S. Secci, "ULOOF: A user level online offloading framework for mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2660–2674, Nov. 2018.

[11] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[12] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[13] F. Guo, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Joint optimization of caching and association in energy-harvesting-powered small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6469–6480, Jul. 2018.

[14] H. Wu, L. Chen, C. Shen, W. Wen, and J. Xu, "Online geographical load balancing for energy-harvesting mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[15] Z. Chen, B. Yang, C. Chen, and X. Guan, "A pre-allocation design for cost minimization and delay constraint in vehicular offloading system," 2019. [Online]. Available: arXiv:1911.09316.

[16] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[17] L. P. Qian, Y. Wu, B. Ji, and X. Shen, "Optimal ADMM-based spectrum and power allocation for heterogeneous small-cell networks with hybrid energy supplies," *IEEE Trans. Mobile Comput.*, early access, Oct. 17, 2019, doi: 10.1109/TMC.2019.2948014.

[18] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[19] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min–max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[20] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Jun. 20, 2019, doi: 10.1109/TCC.2019.2923692.

[21] H. Xu, L. Zhang, X. Zhou, and Z. Han, "Stackelberg differential game based power control in small cell networks powered by renewable energy," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 1–6.

[22] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

[23] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data Networks*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall Int, 1992.

[24] X. Lyu *et al.*, "Optimal schedule of mobile edge computing for Internet of Things using partial information," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov. 2017.

[25] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[26] J. Yang, Q. Yang, Z. Shen, and K. S. Kwak, "Suboptimal online resource allocation in hybrid energy supplied OFDMA cellular networks," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1639–1642, Aug. 2016.

[27] X. Lyu, C. Ren, W. Ni, H. Tian, and R. P. Liu, "Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 574–586, Mar. 2018.

[28] T. Han and N. Ansari, "Network utility aware traffic load balancing in backhaul-constrained cache-enabled small cell networks with hybrid power supplies," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2819–2832, Oct. 2017.