

Article

Not peer-reviewed version

---

# Energy-Efficient Task Offloading in Wireless-Powered MEC: A Dynamic and Cooperative Approach

---

[Huaiwen He](#) , [Chenghao Zhou](#) , Feng Huang , Hong Shen , [Shuangjuan Li](#) \*

Posted Date: 1 July 2024

doi: 10.20944/preprints202407.0056.v1

Keywords: Mobile edge computing(MEC); wireless power transfer(WPT); user cooperation; Lyapunov optimization; convex optimization



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Energy-Efficient Task Offloading in Wireless-Powered MEC: A Dynamic and Cooperative Approach

Huaiwen He <sup>1</sup>, Chenghao Zhou <sup>1,2</sup>, Feng Huang <sup>1,2</sup>, Hong Shen <sup>3</sup> and Shuangjuan Li <sup>4,\*</sup>

<sup>1</sup> School of Computer, Zhongshan Institute, University of Electronic Science and Technology of China, China ; he\_huai\_wen@aliyun.com

<sup>2</sup> Computer Science and Engineering School, University of Electronic Science and Technology of China, Chengdu, China; lnblnbnbn@gmail.com, 202321080835@std.uestc.edu.cn

<sup>3</sup> School of Engineering and Technology, Central Queensland University, Australia; hongshen@gmail.com

<sup>4</sup> College of Mathematics and Informatics, South China Agricultural University, China; lishj2016@scau.edu.cn

\* Correspondence: lishj2016@scau.edu.cn

**Abstract:** Mobile Edge Computing (MEC) integrated with Wireless Power Transfer (WPT) is emerging as a promising solution to reduce task delays and extend the battery life of Mobile Devices (MDs). Cooperative user communication, or relay technology, enhances communication efficiency for users located far from the base station by mitigating the double near-far effect caused by distance. This is considered a key technology in Beyond 5G (B5G) and future communication systems. In this paper, we focus on maximizing the long-term energy efficiency (EE) of a user-cooperation WPT-MEC system, while taking into account the uncertain load dynamics at the edge MD and the time-varying state of the wireless channel. The joint optimization of wireless charging time fraction, MD offloading duration, helper node processing time, and transfer power decision presents significant challenges due to the coupling of data offloading among cooperative users and a volatile system environment. To address these challenges, we formulate the problem as a stochastic programming problem and propose an online control algorithm, DOUCA, to solve it. Our approach utilizes Dinkelbach's method and Lyapunov optimization theory to decouple the sequential decision problem into a deterministic sub-problem for each time slot. For the sub-problem, we use variable substitution to convert the non-convex problem into a convex one, containing only a small number of variables, which can be efficiently solved. Furthermore, we provide a mathematical analysis of our algorithm's performance. Extensive simulation results demonstrate the effectiveness of our proposed algorithm, as evidenced by an impressive energy efficiency improvement of over 20% compared to benchmark methods. Our algorithm also achieves a trade-off between EE and system stability.

**Keywords:** mobile edge computing (MEC); wireless power transfer (WPT); user cooperation; lyapunov optimization; convex optimization

## 1. Introduction

As communication technologies and the Internet of Things (IoT) continue to evolve, we are witnessing an expanding array of wireless device applications across diverse sectors. These include autonomous navigation, virtual reality (VR), intelligent urban planning, and telemedicine procedures [1]. Such applications demand substantial computational resources and are highly sensitive to latency [2,3]. However, they are often constrained by the limited processing capabilities and finite battery life of mobile devices [4]. Mobile Edge Computing (MEC) has emerged as a promising solution to address these constraints, which is a distributed computing paradigm that enhances the computational capabilities of networks. By decentralizing computational and storage resources to the edge of the network, in close proximity to end-users, MEC facilitates the offloading of computation-intensive tasks from wireless devices to nearby servers [5]. This approach not only conserves energy on mobile devices but also significantly reduces execution latency, thereby enhancing the performance and user experience of latency-sensitive applications[6].

However, the constraint of limited battery capacity in mobile devices poses a substantial challenge, particularly given the logistical difficulty of regularly replacing batteries in a vast number of devices. To address this challenge, Wireless Power Transfer (WPT) has been proposed as a sustainable energy solution [7]. In a Wireless Powered Mobile Edge Computing (WPT-MEC) network, a Hybrid Access Point (HAP) serves as the conduit for broadcasting Radio Frequency (RF) energy to wireless devices. Leveraging Energy Harvesting (EH) technology, edge devices can transduce the received RF signals into usable energy to recharge their batteries [8]. This harvested energy then enables the devices to accomplish computation tasks either locally or by offloading them to MEC servers. By exploiting WPT, it is possible to simultaneously extend the battery lifespan of the devices and significantly enhance their computational capabilities [9].

In a WPT-MEC network, Energy Efficiency (EE) is a critical performance indicator, defined as the ratio of data processed to the energy consumed by the system [10,11]. For instance, the authors in [11] introduced two iterative-based optimize algorithms aiming to maximize the computational efficiency of MEC system, considering both partial and binary offloading modes. [10] addressed the EE-delay tradeoff problem in a multi-user WPT-MEC network and proposed an online task offloading algorithm based on Lyapunov optimization method and convex optimization theory. [12] leveraged stochastic network optimization technique to design a task offloading scheme to optimize the network EE for in the device-to-device (D2D)-aided WPT-MEC system. However, these studies have not considered the double-near-far effect, which can significantly impact edge mobile nodes situated far from the HAP. When a mobile device (MD) is placed at a considerable distance from the HAP, it can result in degraded channel conditions due to the increased communication distance and reduced energy harvesting. This degradation can subsequently cause inefficient data transmission due to signal interference between MDs in close proximity and those further away.

Cooperative computing schemes have been introduced to mitigate the impact of the double-near-far effect [4,7,13–15]. These schemes leverage relay technology, where devices closer to the AP act as relays to transmit signals for devices situated further away, thereby enhancing data rates under unfavorable channel conditions. For example, [13] proposed an iterative algorithm to minimize the AP's total transmit energy, subject to computational task constraints. [4] tackled the user-cooperation problem for maximize the EE for a WPT-MEC integrated with Backscatter communication. However, these works mainly focused on immediate network performance and usually assume that a priori the load level at the edge node can be obtained, neglecting the dynamics of task arrival, battery level rate, and time-varying wireless channel. In a volatile network environment, the dynamic allocation of resources and task offloading for edge and auxiliary nodes become significantly challenging.

This study focus on the problem of maximizing energy efficiency in a user-cooperation WPT-MEC network by jointly considering stochastic task arrival and dynamic wireless channel variations. Furthermore, we integrate a battery to store the harvested energy for both the MD and the helper node. The problem presents significant challenges in two main aspects. First, the unpredictability of task arrivals and the randomness of channel states on both the data transmission and wireless charging channels result in a stochastic optimization problem. Second, the time coupling among wireless charging duration, task offloading at edge node, and task processing at helper node poses a great challenge in finding the optimal solution. To tackle this problem, we formulate it as a stochastic programming problem. By leveraging Dinkelbach's method [16] and Lyapunov optimization technique[17] , we transform the stochastic optimization problem into a deterministic problem for each slot. This problem, while non-convex, is then converted into a convex problem using variable substitution and convex optimization techniques. We proposed an efficient online control algorithm, the Dynamic Offloading for User Cooperation Algorithm (DOUCA) , which can be easily implemented and operates without prior knowledge of future system information.

### 1.1. Related Work

WPT-MEC has garnered significant attention from researchers since it helps alleviate energy limitations of IoT nodes while ensuring real-time performance of mobile applications [18–20]. The authors of [19] proposed an energy-saving scheme for a multi-user NOMA-assisted cooperative terahertz single-input multiple-output (SIMO) MEC system that aims to maximize the total user computation energy efficiency (CEE). In [20], an energy efficiency maximization algorithm based on multi-agent deep reinforcement learning was introduced. This algorithm enhances the computation offloading strategy to achieve maximum energy efficiency in MEC-supported vehicular networks. In [7], the authors addressed the practical nonlinear energy harvesting (EH) model by jointly optimizing various factors such as computation frequency, execution time of MEC servers and IoT devices, offloading time, EH time, transmission power of each IoT device, and power beacons (PB). Additionally, the authors of [21] proposed a distributed sleep control method, which autonomously decides whether to enter sleep mode, thereby reducing energy consumption and improving energy efficiency.

To mitigate the double-near-far effect and fully utilize available resources, many researchers focus on user cooperation-assisted WPT-MEC networks [4,9,22,23]. In [23], the authors proposed an algorithm to maximize computational efficiency for a user cooperation (UC) and non-orthogonal multiple access (NOMA) WPT-MEC network, taking into account a nonlinear energy harvesting model. The authors in [4] propose an innovative UC scheme integrating BackCom and AC to maximize user energy efficiency by leveraging a helper node as a relay between the source node and the HAP. In [24], a user cooperation scheme for a WPT-assisted NOMA-MEC network is developed to minimize the total energy consumption of the system using the Lagrangian method to convert the non-convex optimization problem into a convex one. In [25], the authors propose a novel multi-user cooperation scheme to maximize the weighted sum computation rate by considering partial task offloading and orthogonal frequency-division multiple access (OFDMA) communication technology. Due to the constraints of time-varying network environments, achieving a long-term stable optimal solution remains a significant challenge.

In volatile network environments, more and more research focuses on achieving long-term average system performance [1,15,26,27]. In [26], the authors proposed an online algorithm to minimize energy consumption based on the Lyapunov optimization framework and meta-heuristic methods. In [27], the authors introduced a deep reinforcement learning (DRL) algorithm to minimize long-term energy consumption and employed a concave-convex procedure (CCCP) algorithm to solve the computation and communication resource sub-problem for a MEC system with non-complete overlapping non-orthogonal multiple access (NCO-NOMA) technology. In [28], the authors proposed a dynamic optimization scheme based on queuing theory for a 5G MEC heterogeneous network with multiple energy-harvesting MDs, aiming to minimize the average execution delay of the system. In [29], the authors proposed a multi-agent reinforcement learning algorithm that combines federated learning and adopts a fine-grained training strategy to accelerate convergence in a dynamic community-based MEC environment. Unlike the aforementioned studies, this paper addresses the challenges of task offloading and user cooperation in dynamic WPT-MEC network environments. We consider the dynamic arrival of tasks, time-varying wireless channel conditions, and the time-slot coupling of battery levels. Additionally, the time coupling between user cooperative communication and wireless charging, as well as the data offloading coupling in cooperative communication, further complicate the problem.

### 1.2. Motivations and Contributions

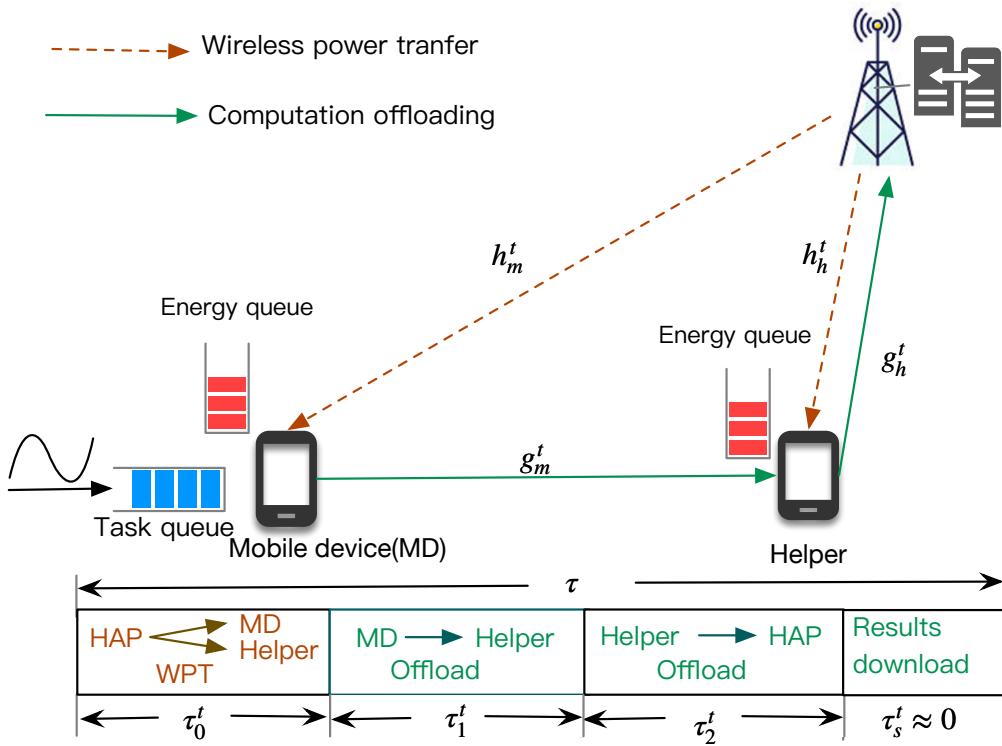
In this paper, we address the problem of long-term energy efficiency optimization in a user-cooperation WPT-MEC network by taking into account the uncertain load dynamic at edge node and time-varying wireless channel state, which has not been extensively investigated in the literature. The main contributions of our work are summarized as follows:

- We formulate the dynamic task offloading problem aiming to maximize the system energy efficiency for a WPT-MEC network. This is achieved by leveraging user cooperation to mitigate the double-near far effect. We extend the existing models in [4,13] to accommodate volatile network environments, eliminating the need for prior knowledge of stochastic task arrival and time-varying wireless channel states. Our model deftly balances the stability of the system network with energy efficiency, thereby providing enhanced flexibility and better alignment with real-world application scenarios.
- We propose DOUCA, a low-complexity online control algorithm designed to maximize long-term UEE, based on Lyapunov optimization theory. Utilizing the drift-plus-penalty technique, we decouple the stochastic programming problem into a non-convex deterministic optimization sub-problem for each slot. Through the use of variable substitution and convex optimization theory, we transform the sub-problem into a convex problem that contains a small number of variables, enabling efficient solutions. Furthermore, we provide a rigorous theoretical analysis to demonstrate its performance.
- We conduct extensive simulations to evaluate the effectiveness and practicality of our proposed algorithm on the impact of control parameter  $V$ , network bandwidth, task arrival rate, and geographical distance on energy efficiency and network stability. The results demonstrate that our algorithm achieves 20% higher efficiency than baseline algorithms and can achieve an explicit  $[O(1/V), O(V)]$  EE-stability tradeoff.

The rest of the paper is organized as follows. Section II presents the system model of the user-cooperation WPT-MEC network and formulates a stochastic programming problem. In Section III, we employ the Lyapunov optimizing technique to solve the problem and propose an efficient online algorithm, accompanied by a theoretical performance analysis. In Section IV, simulation results are presented to evaluate the proposed algorithm. Finally, Section V concludes our work and discusses the future directions.

## 2. System Model

As illustrated in Figure 1, the WPT-MEC system comprises two MDs and a HAP. One MD, situated at a considerable distance from the HAP, is burdened with a substantial computational workload. The other MD, in proximity to the HAP and in an idle state, acts as a helper. Both MDs operate on the same frequency band and are equipped with integrated batteries for energy storage. The HAP is fitted with a RF energy transmitter and a MEC server, which provide wireless energy and computation offloading services to edge nodes within the base station's coverage. To mitigate mutual interference, each MD employs a Time-Division Duplexing (TDD) approach to alternate between communication and energy harvesting operations.



**Figure 1.** Architecture of WPT-MEC network with user-assisted

We adopt a discrete time-slot model over a time horizon divided into  $T$  time blocks, each of duration  $\tau$ . At the commencement of each time slot, both nodes harvest energy from the RF signals emitted by the HAP, which is then stored in their batteries to facilitate subsequent data transmission or local task execution. A partial offloading strategy is implemented, allowing for the flexible offloading of a portion or the entirety of the computational data to a remote device. Due to poor channel conditions between the distant MD and the HAP, exacerbated by the double near-far effect, direct offloading to the MEC server is infeasible. Consequently, the MD offloads computation data to the helper, which then relays it to the HAP. The helper processes the offloaded tasks or further offloads a segment to the HAP. Upon completion, the HAP returns the computation results to the MD, facilitated by the helper. The key notation and definitions are list in Table 1.

**Table 1.** Key notations and definitions.

Notation	Definition
$\tau$	The time block
$\tau_0^t$	The time for WPT
$\tau_1^t$	The time for offloading of MD
$\tau_2^t$	The time for computation in helper
$e_m^{\text{eh}}(t), e_h^{\text{eh}}(t)$	The energy harvested by MD and helper in slot $t$
$h_m^t, h_h^t$	The WPT channel gain between MD and HAP, helper and HAP
$g_m^t, g_h^t$	The offloading channel gain between MD and helper, helper and HAP
$P_0, P_m^t, P_h^t$	The transmit power of HAP, MD and helper
$d_m^{\text{loc}}(t)$	The amount of tasks processed locally at MD in slot $t$
$d_m^{\text{off}}(t)$	The amount of tasks offloaded to helper at MD in slot $t$
$d_h^{\text{loc}}(t)$	The amount of tasks processed locally at helper in slot $t$
$d_h^{\text{off}}(t)$	The amount of tasks offloaded to HAP at helper in slot $t$
$e_m^{\text{loc}}(t)$	The energy consumed by processing tasks at MD in slot $t$
$e_m^{\text{off}}(t)$	The energy consumed by offloading tasks at MD in slot $t$
$e_h^{\text{loc}}(t)$	The energy consumed by processing tasks at helper in slot $t$
$e_h^{\text{off}}(t)$	The energy consumed by offloading tasks at helper in slot $t$
$d_m(t)$	The amount of tasks processed in slot $t$
$e_m(t)$	The energy consumed at MD in slot $t$
$e_h(t)$	The energy consumed at helper in slot $t$
$f_m, f_h$	The local CPU frequency at MD and helper
$\phi_m, \phi_h$	The CPU cycles required to compute one bit task at MD and helper
$b_m^{\text{max}}, b_h^{\text{max}}$	The maximum battery capacity
$\mu$	The energy conversion efficiency
$\kappa$	The computing energy efficiency
$W$	The channel bandwidth
$\varepsilon^2$	The additive white Gaussian noise

## 2.1. Wireless Powered Model

The HAP is equipped with a reliable power source and is responsible for transmitting RF energy to the array of WDs dispersed within its service area. In the first part of each time slot, the HAP broadcasts wireless energy to the MD and the helper for a  $\tau_0^t$  amount of time. Let  $e_m^{\text{eh}}(t), e_h^{\text{eh}}(t)$  denote the harvested energy of MD and helper from HAP at time slot  $t$  respectively. So we have

$$e_m^{\text{eh}}(t) = \mu h_m^t \tau_0^t P_0 \quad (1)$$

and

$$e_h^{\text{eh}}(t) = \mu h_h^t \tau_0^t P_0 \quad (2)$$

where  $0 < \mu < 1$  represents the energy conversion efficiency, and  $P_0$  denotes the RF energy transmit power of the HAP.  $h_m^t$  and  $h_h^t$  denote the channel gain between the MD and HAP, between the helper and HAP, respectively, which remain constant within the same time slot and vary across different time frames.

## 2.2. Task Offloading Model

As depicted in Figure 1, the task data arrival at MD in  $t$ th time slot is denoted as  $A(t) \in [A_{\min}, A_{\max}]$ , which is assumed that  $A(t)$  follows an independent and identically distributed (i.i.d.) in different time slots with exponential distribution with mean  $\lambda$ , e.g.  $\mathbb{E}[A(t)] = \lambda$ . The generated

computation task at time slot  $t$  will be placed in the data queue  $Q$  at MD and waiting for process in a First Come First Server(FCFS) manner. Let  $Q(t)$  denote the backlog of the data queue at slot  $t$ . Thus, the data queue update can be denoted as:

$$Q(t+1) = \max [Q(t) - d_m(t), 0] + A(t) \quad (3)$$

where  $d_m(t) = d_m^{\text{loc}} + d_m^{\text{off}}(t)$  denotes the total data process at MD at slot  $t$ .  $d_m^{\text{loc}}$  represents the data of task executed by the MD locally, and  $d_m^{\text{off}}$  denotes the task offloaded to the helper by wireless transmission.

Let  $f_s$  denote local CPU frequency at MD, a constant value, and  $\phi_s$  represent the CPU cycles required to compute one bit task at MD. The raw data (in bits) processed locally on MD at slot  $t$  is

$$d_m^{\text{loc}}(t) = \frac{f_m \tau}{\phi_m} \quad (4)$$

Note that here  $d_m^{\text{loc}}(t)$  is a constant value, so we rewrite  $d_m^{\text{loc}}(t)$  as  $d_m^{\text{loc}}$ . Meanwhile, the corresponding energy consumed for local computing at slot  $t$  is

$$e_m^{\text{loc}}(t) = \kappa (f_m)^3 \tau \quad (5)$$

where  $\kappa > 0$  denotes the computing energy efficiency parameter, which depends on the circuit structure. Here we adopt the partial task offloading strategy, which means a portion of task data will be offloaded to the helper. Let  $P_m^t$  denote the transmit power of MD, which is constrained by the maximum power  $P_m^t \leq P_m^{\text{max}}$ , and  $\tau_1$  is the amount of offloading time for MD. Thus, according to Shannon's theorem, the offloading data to the helper can be expressed as

$$d_m^{\text{off}}(t) = \tau_1^t W \log \left( 1 + \frac{P_m^t g_m^t}{\sigma^2} \right) \quad (6)$$

where  $W$  denotes the channel bandwidth,  $g_m^t$  denotes the channel gain from MD to helper at slot  $t$ , and  $\sigma^2$  is additive white Gaussian noise. Here there is an upper bound of  $d_m^{\text{off}}(t)$ , that is  $d_m^{\text{off}}(t) \leq d_m^{\text{off, max}}$ , The corresponding energy consumption for task offloading is

$$e_m^{\text{off}}(t) = P_m^t \tau_1^t \quad (7)$$

The MD maintains an energy queue to store the harvested energy from HAP for local computation and task offloading, the energy queue  $B_m$  evolved as follows:

$$B_m(t+1) = \min \left\{ \max [B_m(t) - e_m(t), 0] + e_m^{\text{eh}}(t), b_m^{\text{max}} \right\} \quad (8)$$

where  $e_m(t) = e_m^{\text{loc}}(t) + e_m^{\text{off}}(t)$ , represents the total energy consumption of MD at slot  $t$  and  $b_m^{\text{max}}$  represents the maximum battery capacity of MD.

### 2.3. User Helper Model

We assume that the helper also adopts the partial offloading mode, which means the helper can process the offloading task from MD, while offloading computation task to the edge server. After the initial  $(\tau_0^t + \tau_1^t)$  time, offloading task have arrived the helper, the helper should determine the transmission power  $P_h$  to offloading data to edge server. Similar to the MD, the amount of local computing task data and the corresponding energy consumption at helper at slot  $t$  can be derived as

$$d_h^{\text{loc}}(t) = \frac{f_h \tau_2^t}{\phi_h} \quad (9)$$

$$e_h^{\text{loc}}(t) = \kappa (f_h)^3 \tau_2^t \quad (10)$$

where  $f_h$  denote local CPU frequency at helper, and  $\phi_h$  represent the CPU cycles required to compute one bit task at helper. The amount of offloading data to edge server and the corresponding energy is

$$d_h^{\text{off}}(t) = \tau_2^t W \log \left( 1 + \frac{P_h^t g_h^t}{\sigma^2} \right) \quad (11)$$

$$e_m^{\text{off}}(t) = P_h^t \tau_2^t \quad (12)$$

where  $P_h^t \leq P_h^{\max}$  denotes the transmit power of helper and  $g_h^t$  represents the channel gain from helper and HAP at slot  $t$ . Note that here the helper must process the total offloading task data from MD at each slot  $t$ , so there is not a data queue at the helper, and we have the following constraint

$$d_h^{\text{off}}(t) + d_h^{\text{loc}}(t) \geq d_m^{\text{off}} \quad (13)$$

Similar to the MD, the helper also maintains an energy queue  $B_h$  to store the harvested energy from HAP to support the local computing and task offloading, the battery level of helper updates as

$$B_h(t+1) = \min \left\{ \max [B_h(t) - e_h(t), 0] + e_h^{\text{eh}}(t), b_h^{\max} \right\} \quad (14)$$

where  $e_h(t) = e_h^{\text{loc}}(t) + e_h^{\text{off}}(t)$ , represents the amount of energy consumption of helper at slot  $t$  and  $b_h^{\max}$  represents the maximum energy can be stored in the battery of helper.

#### 2.4. Network stability and Utility

**Definition 1.** (Queue Stability): The task data queue is strong stable if it satisfies

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{Q(t)\} < \infty$$

The network utility here is defined as the ratio of the total achieved computation data to the total energy consumption, The total accomplished computation data and the total energy consumption of user-assisted network at slot  $t$  can be expressed as, respectively

$$D_{\text{tot}}(t) = d_m^{\text{off}}(t) + d_m^{\text{loc}}(t) \quad (15)$$

$$E_{\text{tot}}(t) = e_m(t) + e_h(t) \quad (16)$$

The EE of the network is defined as the time-average achieved computation data by using a unit energy consumption, which is define as the ration of the long term processed data to the total energy consumption as follows:

$$\eta_{\text{EE}} = \frac{\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{D_{\text{tot}}(t)\}}{\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{E_{\text{tot}}(t)\}} = \frac{\overline{D(t)}}{\overline{E(t)}} \quad (17)$$

#### 2.5. Problem Formulation

In this paper, our objective is to design a dynamic control algorithm that maximizes the time-average network EE for a user-assisted WPT-MEC system, all under the constraint of network stability. For each time slot, we make decisions regarding the time allocation for WPT, task offloading time to the helper, task processing time at the helper, and transmit power at MD and helper, without knowing the future channels and data arrivals. Let  $\vec{\tau}(t) = [\tau_0^t, \tau_1^t, \tau_2^t]$  denote the time allocation at slot

$t$  and  $\vec{P}(t) = [P_m^t, P_h^t]$  denote the transmit power of MD and helper. The problem can be formulated a multi-stage stochastic optimization problem as follows:

$$P0 : \max_{\vec{\tau}(t), \vec{P}(t)} \eta_{EE}(t) \quad (18a)$$

$$s.t. \quad \tau_0^t + \tau_1^t + \tau_2^t \leq \tau, \quad (18b)$$

$$B_m(t) - e_m(t) + e_m^{eh}(t) \geq 0, \quad (18c)$$

$$B_h(t) - e_h(t) + e_h^{eh}(t) \geq 0, \quad (18d)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{Q(t)\} < \infty, \quad (18e)$$

$$d_m(t) \leq Q(t), \quad (18f)$$

$$d_h^{off}(t) + d_h^{loc}(t) \geq d_m^{off}(t), \quad (18g)$$

$$\tau_0^t, \tau_1^t, \tau_2^t \geq 0, \quad (18h)$$

$$0 \leq p_m^t \leq P_m^{\max}, 0 \leq p_h^t \leq P_h^{\max}, \quad (18i)$$

Constraint (18b) represents the time allocation constraint. Constraints (18c) and (18d) correspond to the battery energy constraints for MD and helper, respectively, indicating that the energy level in the battery must be greater than zero. Constraint (18e) ensures the network stability of the system. (18f) defines the upper bound of the data processed at slot  $t$ , meaning that the amount of data processed will not exceed the length of the current data queue  $Q$ . Constraint (18g) ensures that the data offloaded to the helper should be completely processed within the current time slot.

Problem P0 is a fractional optimization problem, which is typically non-convex. To handle this, we first utilize the Dinkelbach method [30] to transform P0 into a more manageable problem, similar to [31]. We denote the optimal value of  $\eta_{EE}^{\text{opt}}$  as  $\eta_{EE}$  and obtain the following Theorem 1.

**Theorem 1.** *The optimal  $\eta_{EE}^{\text{opt}}$  is achieved if and only if*

$$\max_{\vec{\tau}(t), \vec{P}(t)} \overline{D(t)} - \eta_{EE}^{\text{opt}} \overline{E(t)} = 0 \quad (19)$$

**Proof.** For brevity, here we omit the proof details. See Proposition 3.1 of [32].  $\square$

Since  $\eta_{EE}^{\text{opt}}$  is unknown during the solution process, (18) is still infeasible to tackle. In accordance with the methodology employed in [33], we introduce a new parameter  $u(t)$  and define it as

$$u(t) = \frac{1}{t} \sum_{i=1}^t \frac{D_{\text{tot}}(i)}{E_{\text{tot}}(i)} \quad (20)$$

We set  $u(0) = 0$  at the beginning of the problem. Replacing  $\eta_{EE}(t)$  in (18), the problem P1 can be transformed into

$$P1 : \max_{\vec{\tau}(t), \vec{P}(t)} \overline{D(t)} - u(t) \overline{E(t)} \quad (21a)$$

$$s.t. \quad (18b) - (18i) \quad (21b)$$

where  $u(t)$  is a given parameter that should be updated through the resolution process. It should be noted that  $u(t)$  obtained by (20) will get closer to  $\eta_{EE}^{\text{opt}}$  as time goes by [12]. Therefore, this transformation is reasonable and has the same optimal solution with P0.

### 3. Algorithm Design

To decouple the battery energy level across time slots and ensure the stability of task queue, we leverage the Lyapunov network optimization technique to transform the long-term average problem in to a deterministic optimization problem for each time slot.

#### 3.1. Lyapunov Optimization Formulation

To simply the battery energy queue at MD and helper, we introduce two virtual queues  $\widehat{B}_m(t) = B_m(t) - b_m^{\max}$  and  $\widehat{B}_h(t) = B_h(t) - b_h^{\max}$ . Following the Lyapunov optimization framework, we define a combined queue vector  $\Theta(t) \triangleq [Q(t), \widehat{B}_m(t), \widehat{B}_h(t)]$  and the quadratic Lyapunov function as:

$$L(\Theta(t)) \triangleq \frac{1}{2} (Q^2(t) + \widehat{B}_m^2(t) + \widehat{B}_h^2(t)) \quad (22)$$

Then, we obtain the one-period conditional Lyapunov drift [34] as follows:

$$\Delta(\Theta(t)) \triangleq \mathbb{E}\{L(t+1) - L(t) \mid \Theta(t)\} \quad (23)$$

Note that  $L(\Theta(t))$  represents the congestion of all queues  $Q(t), \widehat{B}_m(t)$  and  $\widehat{B}_h(t)$ . According to the Lyapunov optimization theory [34], we derive the one slot drift-plus-penalty expression as

$$\Delta_V(\Theta(t)) = \Delta(\Theta(t)) - V\mathbb{E}\{D_{\text{tot}}(t) - u(t)E_{\text{tot}}(t) \mid \Theta(t)\} \quad (24)$$

where control parameter  $V$  is a positive value, used to balance the trade-off between network EE and network stability. Actually,  $V$  acts as a weighting factor of the cost optimality in the drift-plus-penalty expression. Increasing the value of  $V$  implies algorithm pay more attention to the network EE, that also means resulting in a larger backlog of task queue  $Q$ . We derive an upper bound of  $\Delta_V(\Theta(t))$  as Lemma 1.

**Lemma 1.** For any control strategy  $\{\vec{\tau}(t), \vec{P}(t)\}$  at each time slot  $t$ , the one slot Lyapunov drift-plus-penalty  $\Delta_V(\Theta(t))$  is bounded by the following inequality:

$$\begin{aligned} \Delta_V(\Theta(t)) &\leq B - V\mathbb{E}\{D_{\text{tot}}(t) - u(t)E_{\text{tot}}(t) \mid \Theta(t)\} + Q(t)\mathbb{E}\{A(t) - d_m(t) \mid \Theta(t)\} \\ &\quad + \widehat{B}_m(t)\mathbb{E}\{e_m^{\text{eh}}(t) - e_m(t) \mid \Theta(t)\} + \widehat{B}_h(t)\mathbb{E}\{e_h^{\text{eh}}(t) - e_h(t) \mid \Theta(t)\} \end{aligned} \quad (25)$$

where  $B$  is a constant that satisfies the following  $\forall t$ :

$$\begin{aligned} B &\geq \frac{1}{2}\mathbb{E}\{d_m(t)^2 + A^2(t) \mid \Theta(t)\} + \frac{1}{2}\mathbb{E}\{e_m^{\text{eh}}(t)^2 + e_m(t)^2 \mid \Theta(t)\} \\ &\quad + \frac{1}{2}\mathbb{E}\{e_h^{\text{eh}}(t)^2 + e_h(t)^2 \mid \Theta(t)\} \end{aligned} \quad (26)$$

**Proof.** By using the inequality  $(\max[a - b, 0] + c)^2 \leq a^2 + b^2 + c^2 + 2a(c - b)$ ,  $\forall a, b, c \geq 0$  and combining Eq. (3), we have

$$\Delta Q(t) = \frac{1}{2} (Q(t+1)^2 - Q(t)^2) \leq \frac{d_m(t)^2 + A^2(t)}{2} + Q(t)(A(t) - d_m(t)) \quad (27)$$

Based on the definition of battery energy queue  $\widehat{B}_m(t)$  and  $\widehat{B}_h(t)$ , we have

$$\Delta \widehat{B}_m(t) = \frac{1}{2} (\widehat{B}_m(t+1)^2 - \widehat{B}_m(t)^2) \leq \frac{1}{2} [e_m^{\text{eh}}(t)^2 + e_m(t)^2] + \widehat{B}_m(t)(e_m^{\text{eh}}(t) - e_m(t)) \quad (28)$$

$$\Delta \widehat{B}_h(t) = \frac{1}{2} \left( \widehat{B}_h(t+1)^2 - \widehat{B}_h(t)^2 \right) \leq \frac{1}{2} \left[ e_h^{\text{eh}}(t)^2 + e_h(t)^2 \right] + \widehat{B}_h(t) \left( e_h^{\text{eh}}(t) - e_h(t) \right) \quad (29)$$

Combining the above inequalities (27), (28) and (29), we obtain the upper bound of the Lyapunov drift-plus-penalty.  $\square$

According to the drift-plus-penalty technique in Lyapunov optimization theory [34], we seek to greedily minimize the upper bound of  $\Delta_V(\Theta(t))$  at each time slot  $t$ , then we can obtain a close-to-optimal solution of problem P2. Therefore, we transform problem P2 to a minimization problem of the RHS (right hand side) of (25). Note that we can observe the value of  $A(t)$ ,  $Q(t)$ ,  $\widehat{B}_m(t)$  and  $\widehat{B}_h(t)$  at the beginning of each slot  $t$ , so we can solve the optimizing problem at each slot. Then the one time slot problem can be represented as

$$\begin{aligned} \text{P2.1 : } \min_{\vec{\tau}(t), \vec{P}(t)} \quad & Q(t) \left[ A(t) - d_m^{\text{loc}} + d_m^{\text{off}}(t) \right] + \widehat{B}_m(t) \left[ e_m^{\text{eh}}(t) - e_m(t) \right] \\ & + \widehat{B}_h(t) \left[ e_h^{\text{eh}}(t) - e_h(t) \right] - V [D_{\text{tot}}(t) - u(t) E_{\text{tot}}(t)] \end{aligned} \quad (30a)$$

$$\text{s.t. } (18b) - (18d), (18f) - (18i) \quad (30b)$$

The proposed problem P2.1 is a non-convex problem and can not be easily solved by classic convex optimization methods. To address this issue, we first introduce auxiliary variables  $\varphi_1 = p_m^t \tau_1^t$  and  $\varphi_2 = p_h^t \tau_2^t$ . Then the problem P2.1 can be simplified as:

$$\text{P2.2 : } \min_{\vec{\tau}(t), \vec{\varphi}(t)} \quad c_1 W \tau_1^t \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right) + c_2 \tau_0^t + c_3 \varphi_1 + c_4 \varphi_2 + k t_2 \quad (31a)$$

$$\text{s.t. } \tau_0^t + \tau_1^t + \tau_2^t \leq \tau \quad (31b)$$

$$a_1 - \varphi_1 + \tau_0^t h_1 \geq 0 \quad (31c)$$

$$a_2 - k \tau_2^t - \varphi_2 + \tau_0^t h_2 \geq 0 \quad (31d)$$

$$\tau_1^t W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right) \leq Q_t - a_3 \quad (31e)$$

$$a_4 \tau_2^t + \tau_2^t W \log \left( 1 + \frac{\varphi_2}{\tau_2^t} s_2 \right) \geq \tau_1^t W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right) \quad (31f)$$

$$\tau_0^t, \tau_1^t, \tau_2^t \geq 0 \quad (31g)$$

$$0 \leq \varphi_1 \leq \tau P_m^{\max}, 0 \leq \varphi_2 \leq \tau P_h^{\max} \quad (31h)$$

where  $a_1 = \widehat{B}_m(t) - \kappa(f_m)^3 \tau$ ,  $a_2 = \widehat{B}_h(t) - \kappa(f_h)^3 \tau$ ,  $a_3 = \frac{f_m \tau}{\phi_m}$ ,  $a_4, c_1 = -[Q(t) + V], c_2 = \widehat{B}_m(t) \mu h_m^t \tau_0^t P_0 + \widehat{B}_h(t) \mu h_h^t \tau_0^t P_0$ ,  $c_3 = Vu - \widehat{B}_m(t)$ ,  $c_4 = Vu(t) - \widehat{B}_h(t)$ ,  $k = c_3 \kappa(f_m)^3$ ,  $h_1 = \mu h_m^t P_0$ ,  $h_2 = \mu h_h^t P_0$ ,  $s_1 = \frac{g_m^t}{\sigma^2}$ ,  $s_2 = \frac{g_h^t}{\sigma^2}$ . The problem P2.2 is still a non-convex problem due to the non-convex constraint (31f). In (31f), both sides of the equation is concave that do not satisfy the the conditions for convex constraints. We introduce auxiliary variable  $\psi$  to replace the concave function  $\tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right)$ .  $\psi$  satisfies

$$\psi \leq \tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right) \quad (32)$$

Bring in  $\psi$  and constraint (32), the problem P2.2 can be transformed into

$$P3 : \min_{\vec{\tau}(t), \vec{\varphi}(t)} c_1\psi + c_2\tau_0^t + c_3\varphi_1 + c_4\varphi_2 + kt_2 \quad (33a)$$

$$s.t. \quad \tau_0^t + \tau_1^t + \tau_2^t \leq \tau \quad (33b)$$

$$a_1 - \varphi_1 + \tau_0^t \mu h_m^t P_0 \geq 0 \quad (33c)$$

$$a_2 - k\tau_2^t - \varphi_2 + \tau_0^t \mu h_h^t P_0 \geq 0 \quad (33d)$$

$$a_4\tau_2^t + \tau_2^t W \log \left( 1 + \frac{\varphi_2}{\tau_2^t} s_2 \right) \geq \psi \quad (33e)$$

$$\psi \leq \tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right) \quad (33f)$$

$$\psi \leq Q_t - a_3 \quad (33g)$$

$$\tau_0^t, \tau_1^t, \tau_2^t \geq 0 \quad (33h)$$

$$0 \leq \varphi_1 \leq \tau P_m^{\max}, 0 \leq \varphi_2 \leq \tau P_h^{\max}, \quad (33i)$$

**Lemma 2.**  $\psi = \tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right)$  when the problem P3 reaches the optimal solution, which is consistent with P2. P3 is a convex optimization problem, which can be efficiently solved by convex optimization tools, such as CVX[35].

**Proof.** step 1. Assuming that  $\psi < \tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right)$ , we can reduce the objective function by simply increasing  $\psi$  to  $\tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right)$ . Thus, a better solution is found.

step 2. In problem P3, the objective function (33a) is linear with respect to all variables. Constraints (33b)-(33d) and (33g) are all linear inequality constraints. What's more, for constraint (33e),  $\tau_2^t W \log \left( 1 + \frac{\varphi_2}{\tau_2^t} s_2 \right)$  is the perspective of  $W \log(1 + \varphi_2 s_2)$ , which is a concave function of  $\varphi_2$ . Since the perspective operation preserves convexity [36],  $\tau_2^t W \log \left( 1 + \frac{\varphi_2}{\tau_2^t} s_2 \right)$  is concave with respect to  $\varphi_2$  and  $\tau_2^t$ . It is obvious that  $a_4\tau_2^t$  and  $\psi$  are linear functions. Thus, the (33e) is a convex constraint. For the same reason,  $\tau_1 W \log \left( 1 + \frac{\varphi_1}{\tau_1^t} s_1 \right)$  in (33f) is concave with respect to  $\varphi_1$  and  $\tau_1^t$  so that (33f) is a convex constraint. Thus P3 is proved to be convex.  $\square$

According to Lemma 2, at each time slot, we only need to solve a convex problem, P3, which contains a small number of variables. By doing so, we can achieve optimal long-term average EE, even without knowledge of future system information. Our proposed algorithm, the Dynamic Offloading for User Cooperation Algorithm (DOUCA), is summarized as Algorithm 1.

**Algorithm 1:** Dynamic Offloading for User Cooperation Algorithm (DOUCA)

---

**Input:** the task arrival queue  $A(t)$ ; the channel gain  $\{h_m^t, h_h^t, g_m^t, g_h^t\}$ .

1 Initialize  $t \leftarrow 0, Q(0), \hat{B}_m(0), \hat{B}_h(0), u(0)$ ; **for**  $t = 1 : T$  **do**

2    obtain the environmental parameters  $A(t), h_m^t, h_h^t, g_m^t, g_h^t$

3    cvx\_begin

4     Minimize P3

5     Subject to (33b)-(33i)

6    cvx\_end

7    Update the data queue  $Q(t)$

8    Update the battery queue  $\hat{B}_m(t)$  and  $\hat{B}_h(t)$

9    Update the energy efficiency  $u(t)$

**Output:**  $\vec{\tau}(t), \vec{\varphi}(t)$

---

### 3.2. Algorithm Complexity Analysis

At each time slot, we are required to solve a simple convex optimization problem, P3, which contains only five decision variables. This can be efficiently solved using mature methods such as the interior point method, which has a computational complexity of approximately  $O(n^{3.5} \log(1/\epsilon))$ , where  $n$  is the number of decision variables. In our case, we efficiently solve P3 using CVX.

### 3.3. Algorithm Performance Analysis

In this section, we analyze that the proposed scheme can achieve the optimal long-term time-average solution. First, we give some assumptions as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} D_{\text{tot}}(t) = d^{\text{ave}} \quad (34)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E_{\text{tot}}(t) = e^{\text{ave}} \quad (35)$$

$$\lim_{T \rightarrow \infty} u(t) = \frac{\frac{1}{T} \sum_{t=0}^{T-1} D_{\text{tot}}(t)}{\frac{1}{T} \sum_{t=0}^{T-1} E_{\text{tot}}(t)} = u^{\text{ave}} \quad (36)$$

then, we can obtain that the expectation can also converge the same solutions

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[D_{\text{tot}}(t)] = d^{\text{ave}} \quad (37)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[E_{\text{tot}}(t)] = e^{\text{ave}} \quad (38)$$

$$\lim_{T \rightarrow \infty} \mathbb{E}[u(t)] = u^{\text{ave}} \quad (39)$$

**Lemma 3.** Based on (35)–(37), we have

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[u(t)E_{\text{tot}}(t)] = \overline{D(t)} \quad (40)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[u(t)] \leq u \quad (41)$$

To start with, we give the existence of the optimal solution based on the current queue status.

**Lemma 4.** *If the problem (P1) is feasible, there exists a policy  $\{\vec{\tau}(t), \vec{P}(t)\}^*$  that satisfies the following conditions  $\forall t, \varepsilon > 0$ :*

$$\mathbb{E}[D_{\text{tot}}^*] \leq (u^* - \varepsilon) \mathbb{E}[E_{\text{tot}}^*] \quad (42)$$

$$\mathbb{E}[e_m^*(t) - e_m^{\text{eh},*}(t)] \leq \varepsilon \quad (43)$$

$$\mathbb{E}[e_h^*(t) - e_h^{\text{eh},*}(t)] \leq \varepsilon \quad (44)$$

$$\mathbb{E}[A(t) - d_m^*(t)] \leq \varepsilon \quad (45)$$

where  $*$  represents the value under optimal solution.

**Proof.** See the parts 4 and 5 of [17].  $\square$

**Theorem 2.** *The optimal long-average utility function obtained by P1 is limited by a lower bound that is independent with the time space. The following solutions can be achieved by the algorithm,*

1.  $u(t) \geq u^* - B/(Ve^{\text{ave}})$ ,
2. All queues  $Q(t), \hat{B}_m(t), \hat{B}_h(t)$  are mean rate stable, and thus the constraints are satisfied

**Proof.** For any  $\varepsilon > 0$ , let us consider the policy and queue state in (42)–(44). Since the result values  $E_m$  and  $d_m$  are independent of queue status  $\Theta(t)$ , we have

$$\mathbb{E}[D_{\text{tot}}^* | \Theta(t)] \leq (u^* - \varepsilon) \mathbb{E}[E_{\text{tot}}^* | \Theta(t)] \quad (46)$$

$$\mathbb{E}[e_m^*(t) - e_m^{\text{eh},*}(t) | \Theta(t)] \leq \varepsilon \quad (47)$$

$$\mathbb{E}[A(t) - d_m^{\text{loc}} - d_m^{\text{off},*}(t) | \Theta(t)] \leq \varepsilon \quad (48)$$

$$\mathbb{E}[-d_{\text{mec}}(t) + d_m^{\text{off},*}(t) | \Theta(t)] \leq \varepsilon \quad (49)$$

By integrating these results to (25) and making  $\varepsilon \rightarrow 0$ , we have

$$\begin{aligned} \Delta_V(\Theta(t)) &\leq B - V \mathbb{E}\{D_{\text{tot}}^*(t) - u^* E_{\text{tot}}^*(t)\} + Q(t) \mathbb{E}\{A(t) - d_m^{\text{loc}} - d_m^{\text{off},*}(t)\} \\ &\quad + \hat{B}_m(t) \mathbb{E}\{e_m^{\text{eh},*}(t) - e_m^{\text{loc},*}(t) - e_m^{\text{off},*}(t)\} + \hat{B}_h(t) \mathbb{E}\{e_h^{\text{eh},*}(t) - e_h^{\text{loc},*}(t) - e_h^{\text{off},*}(t)\} \\ &\leq B - V \mathbb{E}\{D_{\text{tot}}^*(t)\} + Vu^* \mathbb{E}\{E_{\text{tot}}^*(t)\} \end{aligned}$$

Note that  $u(t)$  is a constant value, which is independent of the current queue status  $\Theta(t)$ . Utilizing the iterated expectation and obtaining the sum of the above inequality over time  $t \in \{0, 1, \dots, T-1\}$ , we have

$$\begin{aligned} \mathbb{E}\{L[\Theta(T)]\} - \mathbb{E}\{L[\Theta(0)]\} &- V \sum_{t=0}^{T-1} \mathbb{E}\{D_{\text{tot}}(t)\} + V \sum_{t=0}^{T-1} \mathbb{E}\{u(t) E_{\text{tot}}(t)\} \\ &\leq T[B - Vu^* \mathbb{E}\{E_{\text{tot}}^*(t)\}] + V \left[ \sum_{t=0}^{T-1} \mathbb{E}\{u^*\} E_{\text{tot}}^*(t) \right] \end{aligned}$$

Dividing both sides of (63) by  $VT$ , utilizing the Jensens inequality and the fact that  $\mathbb{E}\{L[\Theta(T)]\} \geq 0$ , we have

$$-\left[\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{D_{tot}(t)\}\right] + \left[\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{uE_{tot}(t)\}\right] \leq \frac{B}{V} - u^* \mathbb{E}\{E_{tot}^*(t)\} + \left(\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{u(t)\}\right) E_{tot}^*(t) \quad (50)$$

Furthermore, letting  $T \rightarrow \infty$ , we have

$$\begin{aligned} & -\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{D_{tot}(t)\} + \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{u(t)E_{tot}(t)\} \\ & \leq \frac{B}{V} - u^* \mathbb{E}\{E_{tot}^*(t)\} + \left(\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{u(t)\}\right) E_{tot}^*(t) \end{aligned} \quad (51)$$

considering  $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{u(t)\} \leq u$ , we have

$$0 \leq \frac{B}{V} - u^* \mathbb{E}\{E_{tot}^*(t)\} + u \mathbb{E}\{E_{tot}^*(t)\} \quad (52)$$

Furthermore we obtain

$$u \geq u^* - \frac{B}{Vd^{ave}} \quad (53)$$

□

**Theorem 3.** Let  $e_m^{\text{upper}}$  be the upper bound of  $e_m(t)$ , the time-average sum rate of queue length is bounded by

$$Q \leq \frac{B}{\varepsilon} + \frac{ue_m^{\text{upper}}}{\varepsilon} \quad (54)$$

**Proof.** By taking iterated expectation and using telescoping sums over  $t \in \{0, 1, \dots, T-1\}$ , we have

$$\begin{aligned} & \mathbb{E}\{L[\Theta(T)]\} - \mathbb{E}\{L[\Theta(0)]\} - V\mathbb{E}\{D_{tot}(t) - u(t)E_{tot}(t) \mid \Theta(t)\} \\ & \leq TB - \varepsilon \sum_{t=0}^{T-1} \mathbb{E}\{Q(t) + \widehat{B}_m(t) + \widehat{B}_h(t)\} + V\mathbb{E}\{u(t)E_{tot}(t)\} \end{aligned} \quad (55)$$

Dividing both sides of (69) by  $T\varepsilon$ , taking  $T \rightarrow \infty$ , rearranging terms yield

$$\frac{B}{\varepsilon} - \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{Q(t) + \widehat{B}_m(t) + \widehat{B}_h(t)\} + VuE_{tot}^{\text{upper}} \geq 0 \quad (56)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{Q(t) + \widehat{B}_m(t) + \widehat{B}_h(t)\} \leq \frac{B}{\varepsilon} + V \frac{uE_{tot}^{\text{upper}}}{\varepsilon} \quad (57)$$

□

Theorems 2 and 3 provide a rigorous mathematical performance analysis for our proposed algorithm. They demonstrate that the time-average  $\eta_{EE}$  increases at a rate of  $O(1/V)$ , while the queue length increases at a rate of  $O(V)$ . The WPT-MEC system EE  $\eta_{EE}$  can be improved by adjusting the value of  $V$ . However, the time-average task queue  $Q$  will increase with  $V$ . Therefore, we can tune  $V$  to achieve a  $[O(1/V), O(V)]$  trade-off between  $\eta_{EE}$  network EE and task queue length. According to Little's law [33], the latency is proportional to the time-average task queue length. This also implies

that our proposed algorithm can achieve a EE-latency trade-off. This balance is critical in many real-world applications where both efficiency and response time are important.

#### 4. Simulation Results

In this section, we conduct numerical simulation to evaluate the performance of our proposed algorithm. The simulation experiments were executed on a platform equipped with an Intel(R) Xeon(R) Silver 4116 CPU operating at 2.10 GHz, featuring 48 cores, and supplemented by four GeForce RTX 2080 Ti GPUs. We employed a free-space path-loss channel model, and the averaged channel gain  $\bar{h}$  is denoted as

$$\bar{h} = A_d \left( \frac{3 \times 10^8}{4\pi f_c d_i} \right)^{d_e}, \quad (58)$$

where  $A_d = 3$  denotes the antenna gain,  $f_c = 915$  MHz denotes the carrier frequency,  $d_e = 3$  denotes the path loss exponent, and  $d_i$  in meters denotes the distance between two nodes. The time-varying WPT channel gain and task offloading channel gain  $\mathbf{h}^t = [a_1^t h_m^t, a_2^t h_h^t, a_3^t g_m^t, a_4^t g_h^t]$  follows Rayleigh fading channel model, where  $a_i$  are the random channel fading factor following an exponential distribution with unit mean. Let  $\mathbf{a}^t$  represents  $[a_1^t, a_2^t, a_3^t, a_4^t]$ . For simplicity here, we assume that  $\mathbf{a}^t = [1.0, 1.0, 1.0, 1.0]$  and the channel gains remain the same in a single slot. The interval between task arrivals follows an exponential distribution with a constant average rate  $\lambda$ . The other parameters are set similar to [37] and listed in Table 2.

**Table 2.** Simulation parameters.

Symbol	Value
Time slot length	1 s
Transmit power of the AP $P_0$	4 W
Noise power $\sigma^2$	$10^{-4}$ W
Distance between the AP and the MD $d_{ma}$	230 m
Distance between the MD and the Helper $d_{mh}$	140 m
Distance between the AP and the Helper $d_{hs}$	200 m
CPU frequency of SN $f_s$	160MHz
CPU frequency of Helper $f_h$	220 MHz
CPU cycles to compute 1 bit task of SN $\phi_s$	180 cycles/bit
CPU cycles to compute 1 bit task of Helper $\phi_h$	200 cycles/bit
Equal computing efficiency parameter $\kappa_s$	$10^{-8}$
Max battery capacity	15 J

##### 4.1. Impact of System Parameters on Algorithm Performance

Figure 2, illustrates the variation curves of EE and average task queue length  $Q$  over the 5000 time slots under different control parameters  $V = 30, 50$  and the task arrival rate  $\lambda = 2.2$ Mbps. As shown in Figure 2, the EE exhibits significant fluctuations during the initial phase, but as time goes on, the curve gradually stabilizes and rapid convergence. Meanwhile, the task queue length  $Q$  decreases with time slot  $t$  increasing and becomes stable over time, demonstrating the effectiveness of our proposed algorithm. Furthermore, we observe that larger control parameter  $V$  result in higher EE. However, the average queue length also increases accordingly, which is consistent with the theoretical analysis of algorithm.

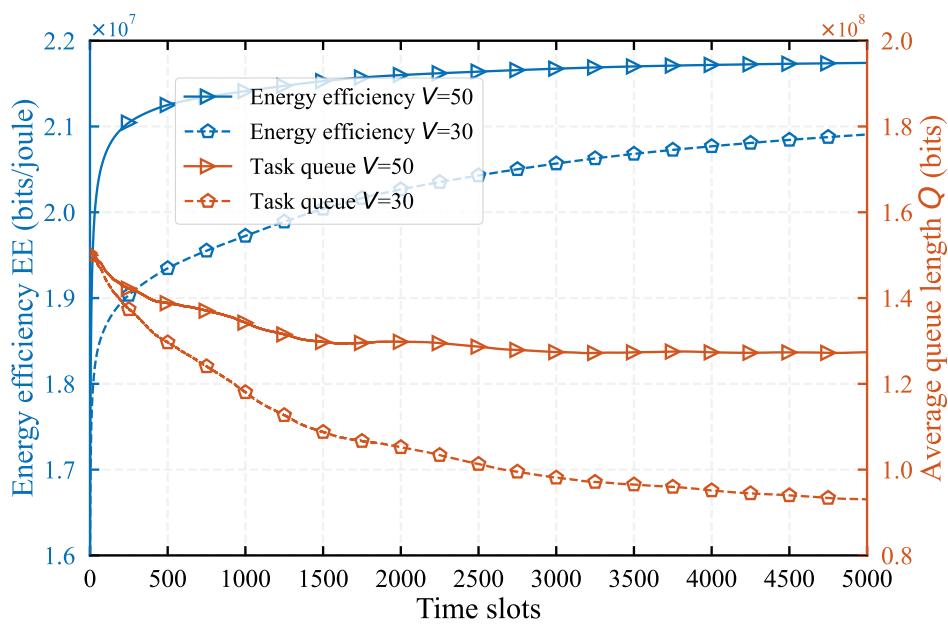


Figure 2. Convergence performance of energy efficiency EE and task queue  $Q$  over time slots

Figure 3 demonstrates the influence of control parameter  $V$  on EE and average task queue length. We find that as  $V$  increases from 10 to 100, the EE escalates from  $1.95 \times 10^7$  bits/joule to  $2.28 \times 10^7$  bits/joule, while the backlog of queue  $Q$  expands from  $0.5 \times 10^8$  bits to  $2 \times 10^8$  bits. This trend signifies that both EE and queue length augment with an increasing  $V$ . This is because with the increasing of the value of  $V$ , our algorithm will focus towards to optimize EE, and paying less attention to the network queue stability. Here,  $V$  acts as a knob to balance the trade-off between EE and network queue stability.

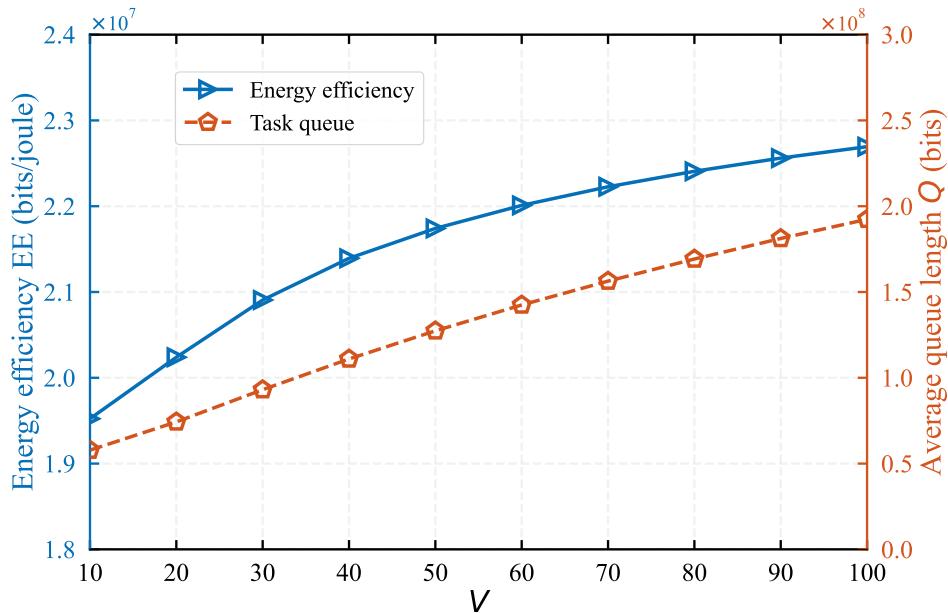


Figure 3. Energy efficiency EE and task queue  $Q$  with different control parameter  $V$

In Figure 4, we evaluate the the impact of network bandwidth  $W$  on system performance under  $V = 50$ . As the network bandwidth  $W$  increases from  $0.85 \times 10^7$ Hz to  $1.35 \times 10^6$ Hz, the EE rises from  $1.7 \times 10^7$ bits/joule to  $2.8 \times 10^7$ bits/joule, while the task queue length  $Q$  decreases from  $3.6 \times 10^8$ bits to  $0.25 \times 10^8$ bits. This is because the increase in network bandwidth improves the speed of task data upload, enabling more tasks to be offloaded to the helper and HAP. Consequently, the amount of tasks processed at MD also increases, leading to a rise in energy efficiency. Additionally, the increased network bandwidth enhances the MD's data processing rate, contributing to the reduction in task queue length. Moreover, Figure 4 shows that when the network bandwidth is below 1, the variation in bandwidth has a more significant impact on the task queue length than on EE. Conversely, when the bandwidth exceeds 1, its variation has a greater impact on EE. This analysis indicates that appropriately increasing the network bandwidth can significantly enhance system performance.

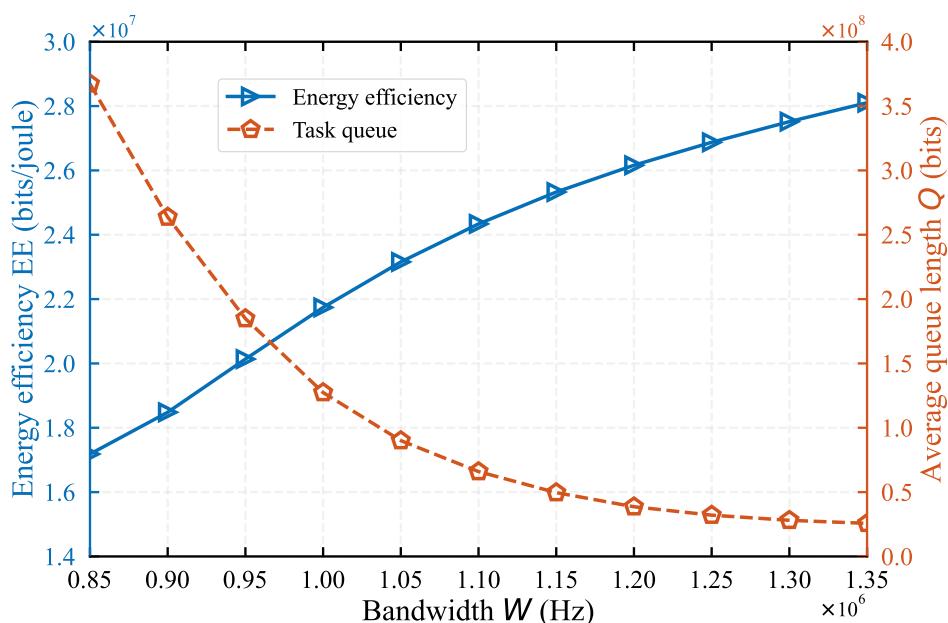
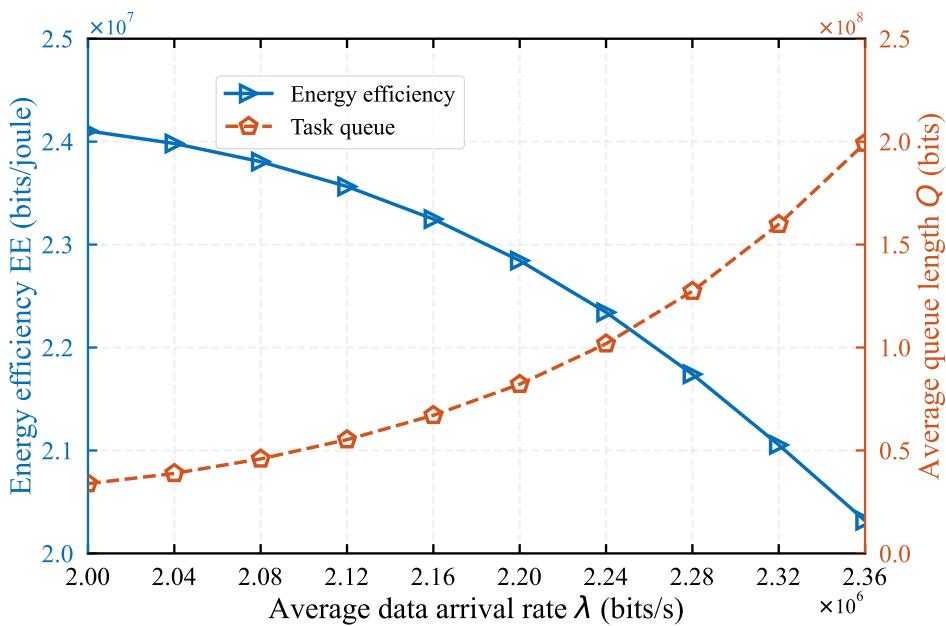


Figure 4. Energy efficiency EE and task queue  $Q$  with different bandwidth  $W$

In Figure 5, we evaluate the impact of task arrival rate  $\lambda$  on system performance when  $V = 50$ . As observed in Figure 5, an increase in task load corresponds to a decrease in EE while the task queue length  $Q$  exhibits an increasing trend, which is consistent with real-world expectations. The reason is that as the task arrival load increases, the processing efficiency of both MD and helper remains unchanged, causing an accumulation of task data in the MD's task queue. Furthermore, Figure 5 indicates that when  $\lambda$  exceeds  $2.3 \times 10^6$ bits/s, the queue length  $Q$  increases rapidly, while energy efficiency continues to decrease linearly. This analysis implies that an excessive load can negatively impact system performance. Consequently, it is crucial to either expand the bandwidth or improve the processing capacity of the MD to address this issue.



**Figure 5.** Energy efficiency EE and task queue  $Q$  with different data arrival rate  $\lambda$

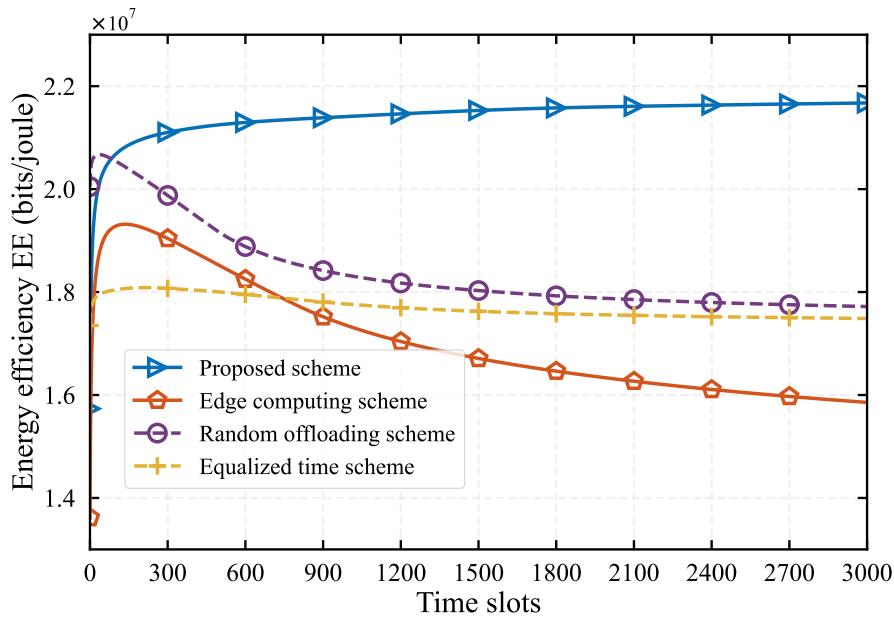
#### 4.2. Comparing with Baseline Algorithms

To evaluate the performance of our algorithm we consider the following three representative benchmarks:

1. Edge computing scheme: The MD does not perform local computation and offloading all task to the helper and HAP.
2. Random offloading scheme: The MD randomly selects part of tasks to offload to the helper and HAP.
3. Equalized time allocation scheme: Allocate task offloading time equally to the MD and helper, which means  $\tau_1 = \tau_2$  in our model.

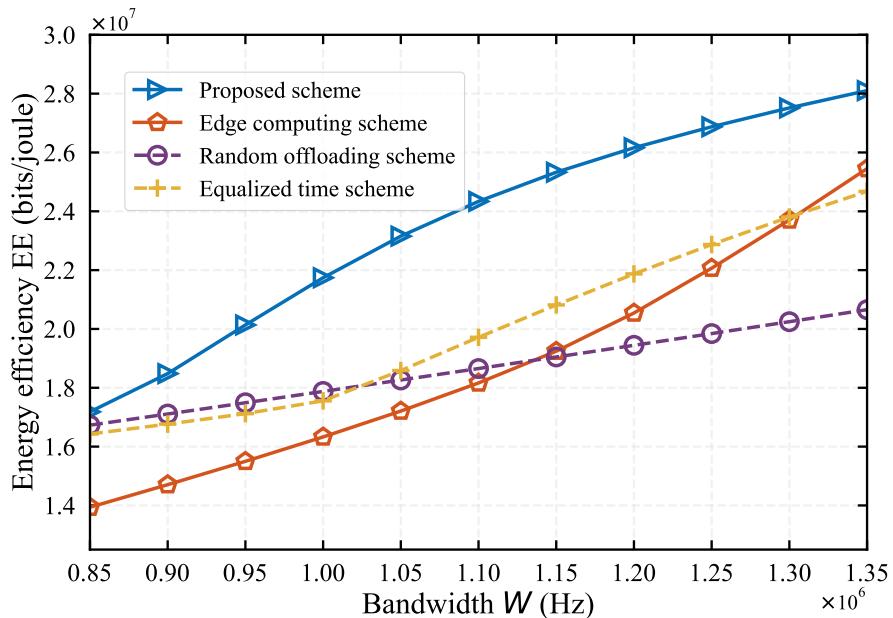
For fairness, it is essential to maintain network queue stability across all methods. Therefore, the three baseline approaches mentioned above are implemented based on the Lyapunov optimization framework.

In Figure 6, we evaluate the performance of our algorithm and the three baseline algorithms with the whole time period 3000 time slots under  $V = 50$ . All algorithm converge after 1000 time slots. Our algorithm achieves the best EE, followed by the random offloading approach, with the equalized time allocation scheme ranking third, and the edge computing method performing the worst. Our algorithm outperforms the other three by 10%, 10%, and 20% respectively. This superior performance can be attributed to our algorithm's consideration of the relationship among charging time, offloading time, and the helper's cooperative time. It also leverages both the local computing resources of MD and the computing resources of the edge server. The edge computing method, which offloads all tasks to the edge server, only considers edge computing resources and overlooks the computing resources of the MD endpoint, resulting in inferior performance. The random offloading algorithm, which can utilize both edge server and local resources, ranks second in performance. Furthermore, the equalized time allocation method ignores priority of MD and Helper when offloading tasks, leading to an inefficient offloading process, and thus reducing the overall system performance.



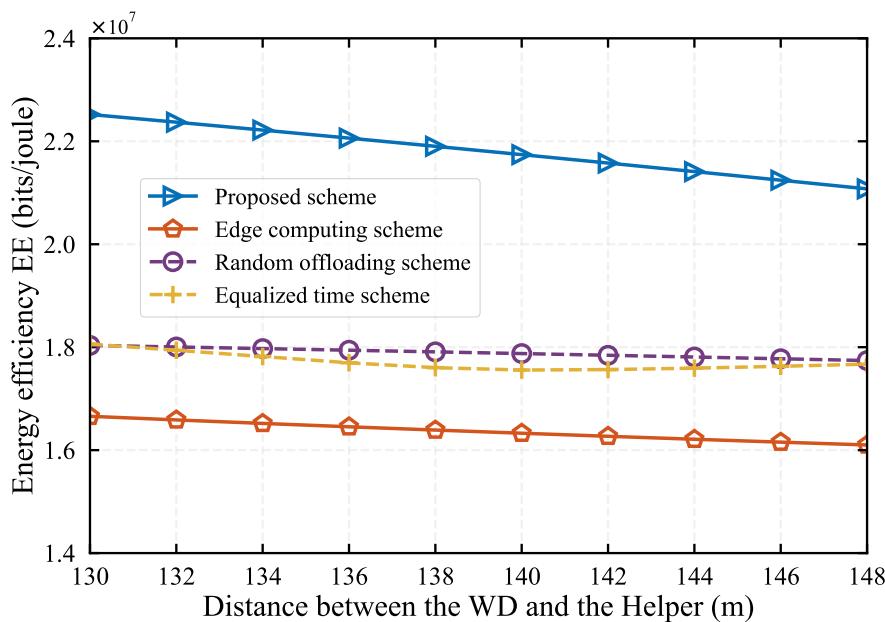
**Figure 6.** Energy efficiency EE in different schemes over time slots

Figure 7 shows the impact of network bandwidth changes, ranging from  $0.85 \times 10^6$ Hz to  $1.35 \times 10^6$ Hz, on network performance across different algorithms. As can be seen from the figure, the EE achieved by all schemes increases with the growth of network bandwidth. This is because all schemes utilize edge computing resources for task computation, and an increase in bandwidth means more tasks can be uploaded to the edge server. This demonstrates the significant influence of network bandwidth on mobile edge networks. Furthermore, our algorithm consistently achieves the best EE across different bandwidths, demonstrating its superiority. Also, it can be seen from the figure that when the network bandwidth is around 1.0Mbps, the superiority of our algorithm is most evident, far exceeding that of other baseline algorithms.



**Figure 7.** Energy efficiency EE in different schemes with different bandwidth W

In Figure 8, we evaluate the system performance under different distance between MD and helper for all different algorithms, where the distance varies in [130, 148]. We find that the EE of all algorithms decreases as the distance increases. Across all distances, our algorithm achieves the best EE. When the distance  $d = 148$ m, our algorithm's performance improves by 17%, 17%, and 31% compared to the other three algorithms, demonstrating that our algorithm can more effectively utilize network resources and edge computing resources. Additionally, it can be seen from Figure 8 that as the distance increases, the advantages of EE achieved by our algorithm tends to decline. This suggests that in practical environments, the distance between edge node devices and relays should not be too large, as it could lead to a rapid decline in network performance.



**Figure 8.** Energy efficiency EE in different schemes with different distances between MD and helper

## 5. Conclusions

The joint optimization of computation offloading and resource allocation in WPT-MEC systems poses a significant challenge due to the time-varying network environments and the time-coupling nature of battery charging and discharging. In this study, we concentrate on maximizing the long-term EE of a WPT-MEC system through user collaboration. We formulated an EE maximization problem that takes into account the uncertainty of load dynamics and the time-varying wireless channel. This problem presents substantial difficulties due to the coupling of multiple parameters. To address this issue, we propose an efficient online control algorithm, termed DOUCA. This algorithm employs Dinkelbach's method and Lyapunov optimization theory to decouple the sequential decision problem into a deterministic optimization problem for each time slot. Extensive simulation results validate the effectiveness of our proposed algorithm, which achieves a remarkable improvement in energy efficiency compared to benchmark methods, striking a balance between EE and system stability. An intriguing direction for future research is to investigate scenarios with multiple MDs, which could further enhance the practicality and applicability of the system in real-world settings.

**Author Contributions:** Methodology, H.H.; Validation, Z.C. and L.S.; Formal analysis, H.H.; Investigation, H.H. and H.F.; Resources, H.H.; Data curation, H.H. and S.H.; Writing—original draft, H.H. and H.F.; Writing—review and editing, H.H.; Supervision, L.S. and S.H.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Science and Technology Planning Project of Guangdong Province, China (No.2021A0101180005).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** We thank all of the reviewers for their valuable comments.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Wang, X.; Li, J.; Ning, Z.; Song, Q.; Guo, L.; Guo, S.; Obaidat, M.S. Wireless powered mobile edge computing networks: A survey. *ACM Computing Surveys* **2023**, *55*, 1–37.
2. Wu, T.; He, H.; Shen, H.; Tian, H. Energy-Efficiency Maximization for Relay-Aided Wireless-Powered Mobile Edge Computing. *IEEE Internet of Things Journal* **2024**, *11*, 18534–18548. doi:10.1109/JIOT.2024.3366982.
3. Mi, X.; He, H.; Shen, H. A Multi-Agent RL Algorithm for Dynamic Task Offloading in D2D-MEC Network with Energy Harvesting. *Sensors* **2024**, *24*. doi:10.3390/s24092779.
4. He, Y.; Wu, X.; He, Z.; Guizani, M. Energy efficiency maximization of backscatter-assisted wireless-powered MEC with user cooperation. *IEEE Transactions on Mobile Computing* **2023**.
5. Guo, H.; Liu, J.; Zhang, J.; Sun, W.; Kato, N. Mobile-edge computation offloading for ultradense IoT networks. *IEEE Internet of Things Journal* **2018**, *5*, 4977–4988. Publisher: IEEE.
6. Hu, Y.C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile edge computing—A key technology towards 5G. *ETSI white paper* **2015**, *11*, 1–16.
7. Shi, L.; Ye, Y.; Chu, X.; Lu, G. Computation energy efficiency maximization for a NOMA-based WPT-MEC network. *IEEE Internet of Things Journal* **2021**, *8*, 10731–10744.
8. Li, G.; Zeng, M.; Mishra, D.; Hao, L.; Ma, Z.; Dobre, O.A. Latency minimization for IRS-aided NOMA MEC systems with WPT-enabled IoT devices. *IEEE Internet of Things Journal* **2023**.
9. Mao, S.; Wu, J.; Liu, L.; Lan, D.; Taherkordi, A. Energy-efficient cooperative communication and computation for wireless powered mobile-edge computing. *IEEE Systems Journal* **2020**, *16*, 287–298. Publisher: IEEE.
10. Mao, S.; Leng, S.; Maharjan, S.; Zhang, Y. Energy Efficiency and Delay Tradeoff for Wireless Powered Mobile-Edge Computing Systems With Multi-Access Schemes. *IEEE Transactions on Wireless Communications* **2019**, *PP*, 1–1.
11. Zhou, F.; Hu, R.Q. Computation efficiency maximization in wireless-powered mobile edge computing networks. *IEEE Transactions on Wireless Communications* **2020**, *19*, 3170–3184. Publisher: IEEE.
12. Sun, M.; Xu, X.; Huang, Y.; Wu, Q.; Tao, X.; Zhang, P. Resource management for computation offloading in D2D-aided wireless powered mobile-edge computing networks. *IEEE Internet of Things Journal* **2020**, *8*, 8005–8020.
13. Hu, X.; Wong, K.K.; Yang, K. Wireless powered cooperation-assisted mobile edge computing. *IEEE Transactions on Wireless Communications* **2018**, *17*, 2375–2388.
14. He, B.; Bi, S.; Xing, H.; Lin, X. Collaborative Computation Offloading in Wireless Powered Mobile-Edge Computing Systems. 2019 IEEE Globecom Workshops (GC Wkshps), 2019, pp. 1–7. doi:10.1109/GCWkshps45667.2019.9024424.
15. Hu, H.; Song, W.; Wang, Q.; Hu, R.Q.; Zhu, H. Energy Efficiency and Delay Tradeoff in an MEC-Enabled Mobile IoT Network. *IEEE Internet of Things Journal* **2022**, *9*, 15942–15956. doi:10.1109/JIOT.2022.3153847.
16. Boyd, S.P.; Vandenberghe, L. *Convex optimization*; Cambridge university press, 2004.
17. Neely, M. *Stochastic network optimization with application to communication and queueing systems*; Springer Nature, 2022.
18. Zhou, F.; Hu, R.Q. Computation Efficiency Maximization in Wireless-Powered Mobile Edge Computing Networks. *IEEE Transactions on Wireless Communications* **2020**, *19*, 3170–3184. doi:10.1109/TWC.2020.2970920.
19. MEC, N.A.C.T.S. Energy-Efficient Optimization of Multi-User NOMA-Assisted Cooperative THz-SIMO MEC Systems **2023**.
20. Ernest, T.Z.H.; Madhukumar, A. Computation Offloading in MEC-Enabled IoV Networks: Average Energy Efficiency Analysis and Learning-Based Maximization. *IEEE Transactions on Mobile Computing* **2023**.
21. Park, J.; Lim, Y. Bio-inspired sleep control for improving the energy efficiency of a mec system. *Applied Sciences* **2023**, *13*, 2620.

22. He, B.; Bi, S.; Xing, H.; Lin, X. Collaborative Computation Offloading in Wireless Powered Mobile-Edge Computing Systems. 2019 IEEE Globecom Workshops (GC Wkshps), 2019.
23. Su, B.; Ni, Q.; Yu, W.; Pervaiz, H. Optimizing computation efficiency for NOMA-assisted mobile edge computing with user cooperation. *IEEE Transactions on Green Communications and Networking* **2021**, *5*, 858–867.
24. Wang, R.; Chen, J.; He, B.; Lv, L.; Zhou, Y.; Yang, L. Energy consumption minimization for wireless powered NOMA-MEC with user cooperation. 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2021, pp. 1–5.
25. Wu, X.; He, Y.; Saleem, A. Computation rate maximization in multi-user cooperation-assisted wireless-powered mobile edge computing with OFDMA. *China Communications* **2023**, *20*, 218–229.
26. Zeng, Y.; Chen, S.; Cui, Y.; Du, J. Efficient Trajectory Planning and Dynamic Resource Allocation for UAV-Enabled MEC System. *IEEE Communications Letters* **2024**.
27. Luo, Y.; Lin, S.; Hong, X.; Shi, J. DRL-Assisted Resource Allocation for Non-Completely Overlapping NOMA Based Dynamic MEC Systems. *IEEE Internet of Things Journal* **2024**.
28. Liu, L.; Yuan, X.; Chen, D.; Zhang, N.; Sun, H.; Taherkordi, A. Multi-User Dynamic Computation Offloading and Resource Allocation in 5G MEC Heterogeneous Networks With Static and Dynamic Subchannels. *IEEE Transactions on Vehicular Technology* **2023**.
29. Sun, W.; Zhao, Y.; Ma, W.; Guo, B.; Xu, L.; Duong, T.Q. Accelerating convergence of federated learning in mec with dynamic community. *IEEE Transactions on Mobile Computing* **2023**.
30. Dinkelbach, W. On nonlinear fractional programming. *Management science* **1967**, *13*, 492–498.
31. Yu, Y.; Bu, X.; Yang, K.; Wu, Z.; Han, Z. Green large-scale fog computing resource allocation using joint benders decomposition, Dinkelbach algorithm, ADMM, and branch-and-bound. *IEEE Internet of Things Journal* **2018**, *6*, 4106–4117.
32. Zappone, A.; Jorswieck, E.; others. Energy efficiency in wireless networks via fractional programming theory. *Foundations and Trends® in Communications and Information Theory* **2015**, *11*, 185–396.
33. Mao, S.; Leng, S.; Maharjan, S.; Zhang, Y. Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes. *IEEE Transactions on Wireless Communications* **2019**, *19*, 1855–1867.
34. Neely, M.J. Stochastic network optimization with non-convex utilities and costs. 2010 Information Theory and Applications Workshop (ITA). IEEE, 2010, pp. 1–10.
35. Diamond, S.; Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* **2016**, *17*, 1–5.
36. Zawawi, Z.B.; Huang, Y.; Clerckx, B. Multiuser wirelessly powered backscatter communications: Nonlinearity, waveform design, and SINR-energy tradeoff. *IEEE Transactions on Wireless Communications* **2018**, *18*, 241–253.
37. Bi, S.; Huang, L.; Wang, H.; Zhang, Y.J.A. Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks. *IEEE Transactions on Wireless Communications* **2021**, *20*, 7519–7537.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.