

Article

Not peer-reviewed version

Maximizing Computation Rate for Sustainable Wireless Powered MEC network: An Efficient Dynamic Task Offloading Algorithm with User Assistance

[Huaiwen He](#)*, Feng Huang, [Chenghao Zhou](#), Hong Shen, Yihong Yang

Posted Date: 18 July 2024

doi: 10.20944/preprints202407.1542.v1

Keywords: mobile edge computing (MEC); wireless power transfer (WPT); computation rate; 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.

Article

Maximizing Computation Rate for Sustainable Wireless Powered MEC network: An Efficient Dynamic Task Offloading Algorithm with User Assistance

Huaiwen He ^{1,*},[†] , Feng Huang ^{1,2,†}, Chenghao Zhou ^{1,2}, Hong Shen ³ and Yihong Yang ¹

¹ School of Computer, Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan 528400, China; 202321080835@std.uestc.edu.cn (F.H.); lnlnbnnb@gmail.com (C.Z.)

² Computer Science and Engineering School, University of Electronic Science and Technology of China, Chengdu 611731, China

³ School of Engineering and Technology, Central Queensland University, Rockhampton 4701, Australia; hongshen@gmail.com

* Correspondence: he_huai_wen@aliyun.com

[†] These authors contributed equally to this work.

Abstract: In the Internet of Things (IoT) era, Mobile Edge Computing (MEC) significantly enhances the efficiency of smart devices but is limited by battery life issues. Wireless Power Transfer (WPT) addresses this issue by providing a stable energy supply. However, effectively managing overall energy consumption remains a critical and under-addressed aspect for ensuring the network's sustainable operation and growth. In this paper, we consider a WPT-MEC network with user cooperation to migrate the double near-far effect for the mobile node (MD) far from the base station. We formulate the problem of maximizing long-term computation rates under a power consumption constraint as a multi-stage stochastic optimization (MSSO) problem. This approach is tailored for a sustainable WPT-MEC network, considering the dynamic and varying MEC network environment, including randomness in task arrivals and fluctuating channels. We introduce a virtual queue to transform the time-average energy constraint into a queue stability problem. Using the Lyapunov optimization technique, we decouple the stochastic optimization problem into a deterministic problem for each time slot, which can be further transformed into a convex problem and solved efficiently. Our proposed algorithm works efficiently online without requiring further system information. Rigorous mathematical analysis shows that our algorithm achieves $[O(1/V), O(V)]$ trade-off between computation rate and queues stability. Extensive simulation results demonstrate that our proposed algorithm outperforms baseline schemes, achieving approximately 4% enhancement while maintaining the queues stability.

Keywords: mobile edge computing (MEC); wireless power transfer (WPT); computation rate; Lyapunov optimization; convex optimization

1. Introduction

In the era of the Internet of Things (IoT) [1–4], the surge in the number of mobile devices and network traffic has created a high demand for data processing capabilities and response speeds. Mobile Edge Computing (MEC) technology effectively enhances the operational efficiency of smart devices by providing powerful computing resources at the network's edge, particularly in delay-sensitive applications such as autonomous driving and AR/VR [5]. By offloading complex computational tasks to MEC servers [6–11], resource-constrained mobile devices (MDs) can significantly alleviate the pressure experienced when running high-demand applications and achieve a notable leap in performance. However, the limited battery capacity of mobile devices has become a bottleneck that restricts their further development [12]. This limitation is especially evident for devices that require continuous operation over long periods and are not easily recharged, such as in remote areas or emergency situations, where insufficient battery life can severely impact the functionality and reliability of the devices. Therefore, despite the significant advantages of MEC in enhancing network performance, the limitation of battery endurance remains a key issue that urgently needs to be addressed.

In addition to battery constraints, reducing the energy consumption of IoT devices during the data offloading process is equally important. Trillions of tiny smart sensors make up the Internet of Things, facing significant limitations in computational capabilities and energy supplied by batteries. Advancements in wireless energy harvesting technologies, including renewable energy harvesting and wireless power transfer (WPT) [13], can alleviate the challenges previously posed by battery capacity limitations. Renewable energy sources like solar, wind, and ocean energy can provide power to some extent, but they are significantly influenced by natural conditions such as weather and climate [14]. To address this issue, green wireless charging technology has emerged. This technology can offer stable energy to devices through radio frequency signals and store it in the batteries of IoT nodes for future use, extending battery life [15,16]. To ensure that nodes do not fail due to energy depletion, green wireless charging adheres to the principle of energy neutrality operation [17] state, ensuring that the energy consumed in any operation will never exceed the energy collected. Green Wireless Powered Mobile Edge Computing (WPMEC) combines the strengths of WPT and MEC, enhancing devices' computational capabilities and energy self-sufficiency. In the upcoming 6G networks, green WPMEC will provide IoT devices with quick response and real-time experiences [18], while reducing operational costs and extending the lifespan of devices.

However, WPMEC networks face the challenge of the dual far-near effect [19] caused by positional differences, which has prompted the development of edge collaborative networks [20–22] to optimize application offloading performance. By introducing a user cooperation (UC) mechanism, where nearby users assist distant users in accelerating the processing of computational tasks while effectively offloading their own tasks, this collaborative approach leverages the superior channel conditions of nearby users to gain more energy during the WPT phase. This not only addresses the unfairness caused by geographical location but also enhances the efficiency of energy utilization. The dense deployment of smart IoT devices further facilitates the opportunity to utilize the unused computing resources of idle devices and wireless energy harvesting. These devices, by assisting in completing the computational tasks of distant users, contribute to improving the overall computational performance of the WPMEC network.

To demonstrate the effectiveness of UC, recent studies, such as references [23,24], have effectively addressed the dual far-near effect in WPMEC networks through UC. The D2D communication in reference [25] and the incentive mechanism in reference [26] are both designed to facilitate resource sharing and collaborative offloading. In references [20,24,27,28], authors have focused on studying the basic three-node WPMEC model, which involves a Far User (FU) being allowed to offload computational input data to a Near User (NU). In reference [29], researchers designed a Non-Orthogonal Multiple Access (NOMA) based computation offloading scheme aimed at enhancing the performance of multi-user MEC systems. Google has also developed federated learning technology, which enables multiple devices to collaborate on machine learning tasks. Despite this, these studies are often based on the assumption of determinism or predictability of future information, failing to fully integrate the dynamic changes of the network environment, which may affect the efficiency and success rate of task offloading and processing.

This paper investigates the basic three-node green WPMEC network shown in Fig. 1, focusing on the use of collaborative communication technology to accomplish the computation-intensive and delay-sensitive tasks powered by the HAP. Our goal is to maximize the network's data processing capability in a real-time dynamic offloading system, taking into account the randomness of data generation and the high dynamics of wireless channel conditions. The challenges we face in addressing this issue include the unpredictability of task arrivals and the dynamics of channel conditions, as well as the coupling of variables in resource allocation, which makes traditional convex optimization methods inapplicable. To tackle these challenges, we have designed an efficient dynamic task offloading algorithm, the User-Assisted Dynamic Resource Allocation Algorithm (UADRA). This algorithm employs Lyapunov optimization techniques to transform the problem into a simplified form that relies

on current information, and performs dynamic resource allocation in each time slot to enhance the network's data processing capability. Our primary contributions are summarized as follows:

- We propose a model for maximizing the long-term weighted computation rate in a green sustainable WPT-MEC network, subject to energy consumption constraints. Our model takes into account the randomness of task arrivals and the dynamic variations in wireless channel states. By extending the models in [28,30], we have effectively mitigated the double near-far effect and fostered collaboration among users. This is achieved by considering the different data weights of near and far nodes, which enhances data transmission efficiency, system flexibility, and alignment with real-world applications.
- We design a low-complexity dynamic control algorithm called UADRA, based on Lyapunov optimization theory, while providing rigorous mathematical theory analysis to demonstrate its performance. A virtual queue is introduced to transform the time-average energy constraint to a queue stable requirement. Leveraging the drift-plus-penalty technique, we decouple the original multi-stage stochastic problem into a non-convex deterministic sub-problem for each time slot. Furthermore, by using variable substitution and convex optimization theory, we convert these sub-problems into convex problems with a minimal number of decisions, that can be solved efficiently. Our algorithm can work efficiently without depending on the prior knowledge of the system information.
- We have conducted extensive simulations to evaluate the effectiveness and practicality of our proposed algorithm, particularly examining how the control parameter V , network bandwidth, energy constraint, task arrival rate, and geographical distance affect the average computation rate and network stability. Experimental results demonstrate that the proposed algorithm not only shows superior performance over benchmark methods but also achieves this improvement while simultaneously ensuring the stability of the system queues, with an enhancement of up to 4% in overall performance. Moreover, our algorithm effectively achieves a trade-off between computation rate and stability, adhering to the theoretical bounds of $[O(1/V), O(V)]$.

The remainder of this paper is organized as follows. Section II we presents the system model of the user-assistance green WPMEC network and sets forth a MSSO problem. In Section III, we utilize the Lyapunov optimization approach to tackle the problem, putting forward an effective dynamic offloading algorithm with an accompanying theoretical analysis of its performance. Section IV evaluate the efficacy of the suggested algorithm via simulation outcomes. Conclusively, Section V concludes the paper.

1.1. Related Work

The integration of WPT technology with MEC networks provides an effective solution for IoT devices, enhancing their energy and computing capabilities with controllable power supply and low-latency services. Recent research has extensively explored the potential of these wirelessly powered MEC networks. For instance, in [31], researchers optimized charging time and data offloading rates for WPT-MEC IoT sensor networks to improve computational rates in scenarios. Furthermore, authors in [32] investigated a NOMA-assisted WPT-MEC network with a nonlinear EH model, successfully enhancing the system's Computational Energy Efficiency (CEE) by fine-tuning key parameters within the network. Specifically, to meet the energy consumption requirements of devices, the authors in [33] proposed a Particle Swarm Optimization (PSO)-based algorithm. The goal was to reduce the latency of devices processing computational data streams by jointly optimizing charging and offloading strategies. Additionally, in [34], the authors focused on the computational latency issue in WPT-MEC networks. They found suitable offloading ratio strategies to achieve synchronized latency for all WDs, effectively reducing the duration of the overall computational task.

To tackle the dual far-near effect issue, researchers have begun to focus on user-assisted WPMEC networks and have confirmed their effectiveness in enhancing the computing performance of distant users. Specifically, in [35], the study analyzed a three-node system composed of distant users, nearby users, and the base station within a user-assisted MEC-NOMA network model, addressing the op-

timization problem of joint transmission time and power allocation for users. Furthermore, [36,37] respectively explored joint computing and communication collaboration schemes and the application of Device-to-Device (D2D) communication in MEC. The method proposed in [36] aims to maximize the total computing rate of the network with the assistance of nearby users, while [37] focuses on minimizing the overall network response delay and energy consumption through joint multi-user collaborative partial offloading, transmission scheduling, and computing allocation. [38] extended this research, expanding from a single collaboration pair to multiple collaboration pairs, proposing a scheme to achieve the minimization of the total energy consumption of the AP.

In user assistance networks, the online collaborative offloading method, which is highly adaptable and can promptly respond to changes in task arrivals, has garnered significant attention from the research community. For instance, in [39], to address the randomness of energy and data arrivals, a Lyapunov optimization-based method was proposed to maximize the long-term system throughput. Furthermore, in [40], the authors studied and proposed a Lyapunov-based Profit Maximization (LBPM) task offloading algorithm in the context of the Internet of Vehicles (IoV), which aims at maximizing the time-averaged profit as the optimization goal. Additionally, in [41], within the application of MEC in the industrial IoT, a Lyapunov-based privacy-aware framework was introduced, which not only addressed privacy and security issues but also achieved optimization effects by reducing energy consumption. In [42], focusing on a multi-device single MEC system, the energy-saving task offloading problem was formulated as a time-averaged energy minimization problem considering queue length and resource constraints.

Different from prior studies, this paper is dedicated to addressing the challenges of dynamic task offloading in a green sustainable WPMEC networks with user assistance. We take into account the total system's energy consumption constraint, the dynamically arriving tasks in real-time scenarios, and the high dynamics of wireless channel conditions. Moreover, the temporal coupling between WPT and user collaborative communication, as well as the coupling of data offloading timing and transmission power in collaborative communication, impose significant challenge to the problem.

2. System Model

2.1. Communication Model

As depicted in Figure 1, we consider a WPMEC system that comprises two MDs and one HAP. The HAP is equipped with both an RF energy transmitter and an MEC server, offering wireless energy and computation offloading services to MDs within the base station's coverage area. Both MDs work on the same frequency band and are tasked with completing computation-intensive, delay-sensitive operations. One MD, known as the Far User (FU), is situated at a considerable distance from the HAP. In contrast, the other MD, called the Near User (NU), is in closer proximity to the HAP and serves as a facilitator, aiding the FU by offloading a portion of its tasks to the HAP.

To minimize interference, each MD utilizes a Time Division Duplex (TDD) approach to switch between communication and energy harvesting phases. We implement a Time Division Multiple Access (TDMA) framework with each time slot lasting T seconds. At the start of each time slot, both MDs capture RF signals emitted by the HAP for energy harvesting. Given the suboptimal channel conditions between the FU and the HAP, along with the compounded near-far effect, the FU is capable of transferring some of its computational data to the NU. The NU then forwards this data to the HAP. Additionally, the NU is scheduled to offload its own computational tasks to the HAP. The primary symbols and definitions used are enumerated in Table 1.

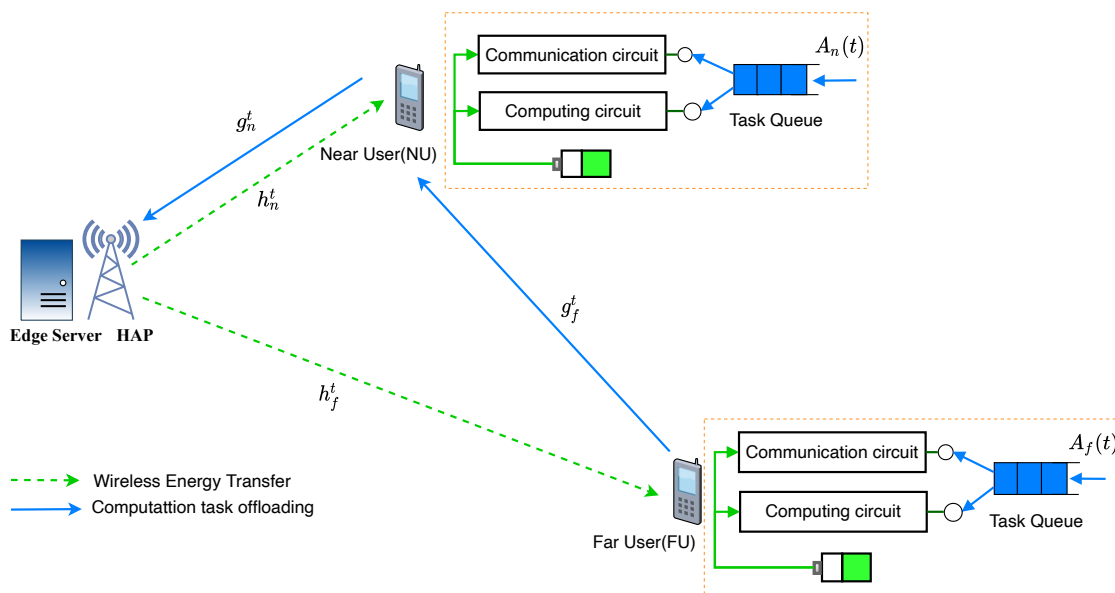


Figure 1. System model of WPMEC network with user-assisted.

Table 1. Key Notations and Definitions.

Notation	Definition
T	The time block
τ_0^t	The time for WPT
τ_1^t	The time for offloading of FU
τ_2^t	The time for NU to offload FU's data
τ_3^t	The time for NU to offload its own data.
$e_m^{\text{eh}}(t), e_h^{\text{eh}}(t)$	The energy harvested by MD and helper in slot t
h_f^t, h_n^t	The WPT channel gain between FU and HAP, NU and HAP
g_f^t, g_n^t	The offloading channel gain between FU and NU, NU and HAP
$P_0^t, P_1^t, P_2^t, P_3^t$	The transmit power for HAP, FU, NU to offload FU's data and NU to offload its own data in slot t
$d_f^{\text{loc}}(t)$	The amount of tasks processed locally at FU in slot t
$d_{f,n}^{\text{off}}(t)$	The amount of tasks offloaded to NU at FU in slot t
$d_n^{\text{loc}}(t)$	The amount of tasks processed locally at NU in slot t
$d_{n,a1}^{\text{off}}(t)$	The amount of tasks that NU offloads to HAP from FU in slot t
$d_{n,a2}^{\text{off}}(t)$	the amount of tasks that NU offloads to HAP from itself in slot t
$e_f^{\text{loc}}(t)$	The energy consumed by processing tasks at FU in slot t
$e_{f,n}^{\text{off}}(t)$	The energy consumed by offloading tasks at FU in slot t
$e_n^{\text{loc}}(t)$	The energy consumed by processing tasks at helper in slot t
$e_{n,a1}^{\text{off}}(t)$	The energy consumed by NU to offload FU's tasks in slot t
$e_{n,a2}^{\text{off}}(t)$	the energy consumed by NU to offload its own tasks in slot t
$d_m(t)$	The amount of tasks processed in slot t
f_f, f_n	The local CPU frequency at FU and NU
ϕ_f, ϕ_n	The CPU cycles required to compute one bit task at FU and NU
μ	The energy conversion efficiency
κ	The computing energy efficiency
W	The channel bandwidth
σ^2	The additive white Gaussian noise

2.2. Task Queue Model

Both FU and NU maintain a queue to buffer the randomly arriving tasks for awaiting processing. Let $Q_f(t)$ and $Q_n(t)$ denote the queue lengths at the FU and NU at slot t , respectively, which can be

observed at the beginning of time slot t . The backlog of task queue update according to the following equations:

$$Q_f(t+1) = \max\left[Q_f(t) - \left(d_f^{\text{loc}}(t) + d_f^{\text{off}}(t)\right), 0\right] + A_f(t) \quad (1)$$

$$Q_n(t+1) = \max\left[Q_n(t) - \left(d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t)\right), 0\right] + A_n(t) \quad (2)$$

where $A_f(t)$ and $A_n(t)$ represent the raw task data arriving at the FU and NU data queues during time slot t , respectively.

In reality, the data queue lengths Q_n and Q_f are always positive. Additionally, since the data processing amount by the MDs cannot exceed the current queue length, we have the following constraints:

$$d_f^{\text{loc}}(t) + d_{f,n}^{\text{off}}(t) \leq Q_f(t) \quad (3)$$

$$d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t) \leq Q_n(t) \quad (4)$$

Additionally, if the local computation of the MD is capable of processing the entire current queue, then local computation is given priority, which is also in line with practical requirements.

2.3. Computation Model

(1) Local Computation Mode: MDs are capable of performing continuous local computations. Let f_f and f_n denote the local CPU frequencies of the FU and NU, respectively. Additionally, let ϕ_f and ϕ_n represent the CPU cycles required to process one bit of data at the FU and NU, respectively. Consequently, the amount of raw data (in bits) processed locally by MDs within a time slot t and the corresponding energy expenditure over that period can be determined by the following threshold rules.

For the FU, the local data processing in bits is given by

$$d_f^{\text{loc}}(t) = \begin{cases} \frac{f_f T}{\phi_f}, & \text{if } \frac{f_f T}{\phi_f} < Q_f(t) \\ Q_f(t), & \text{others} \end{cases} \quad (5)$$

At the begin of each time slot t , we obtain the queue length of the FU. If the queue length exceeds its local processing capacity, the FU will utilize a user collaboration mechanism to offload partial tasks; if the queue length is within the computing capacity, the FU will perform complete local computation (that is, $\tau_1 = 0, \tau_2 = 0$), and the corresponding energy consumed during this period, as combined with $d(t) = ft/\phi$ (where $d(t)$ denotes the amount of data processed locally at slot t) [30], is:

$$e_f^{\text{loc}}(t) = \begin{cases} \kappa (f_f)^3 T, & \text{if } \frac{f_f T}{\phi_f} < Q_f(t) \\ \kappa (f_f)^2 Q_f(t) \phi_f, & \text{others} \end{cases} \quad (6)$$

Similarly, for the NU, the local data processing is

$$d_n^{\text{loc}}(t) = \begin{cases} \frac{f_n T}{\phi_n}, & \text{if } \frac{f_n T}{\phi_n} < Q_n(t) \\ Q_n(t), & \text{others} \end{cases} \quad (7)$$

When the NU engages in complete local computation, $\tau_3 = 0$, with the energy consumed being

$$e_n^{\text{loc}}(t) = \begin{cases} \kappa (f_n)^3 T, & \text{if } \frac{f_n T}{\phi_n} < Q_n(t) \\ \kappa (f_n)^2 Q_n(t) \phi_n, & \text{others} \end{cases} \quad (8)$$

Here, $\kappa > 0$ represents the computational energy efficiency coefficient of the processing chip [43].

(2) Offloading Mode: We adopt a partial task offloading strategy for both FU and NU. As illustrated in Figure 2. Initially, the FU transfers a portion of its task data to the NU during time τ_1^t , using a transmission power P_1^t . Subsequently, the NU forwards the FU's data to the HAP during time τ_2^t with a transmission power P_2^t . Furthermore, the NU offloads its own data to the HAP during time τ_3^t using transmission power P_3^t . Throughout this process, the NU consume the energy it has harvested. It is essential that the amount of data offloaded by the FU for remote computation at the HAP corresponds to the data initially offloaded to the NU.

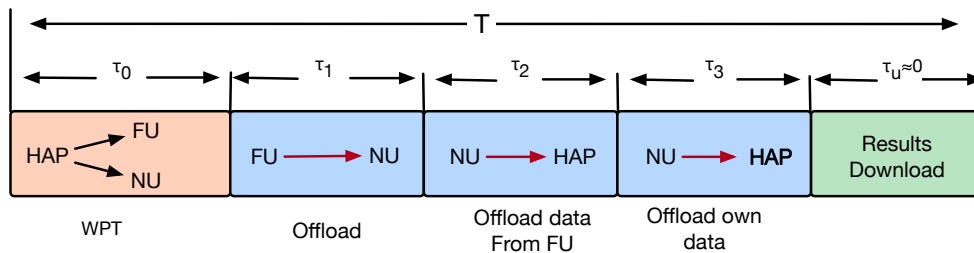


Figure 2. An illustrative time division structure.

In practical applications, the HAP possesses significantly greater computational capacity and transmission power compared to the MDs. As a result, the time required for data computation and feedback at the HAP is negligible. Therefore, the total time allocated for energy harvesting and task offloading by MDs must not exceed the time slot duration T , as represented by the following inequality:

$$\tau_0^t + \tau_1^t + \tau_2^t + \tau_3^t \leq T \quad (9)$$

We focus solely on the energy consumption of the MDs during the offloading process. The transmission power of the FU P_1^t is constrained by $P_1^t \leq P_f^{max}$. According to Shannon's theorem, the amount of data offloaded from the FU to the NU is given by

$$d_{f,n}^{off}(t) = \tau_1^t W \log_2 \left(1 + \frac{P_1^t g_f^t}{\sigma^2} \right) \quad (10)$$

The corresponding energy consumption for this process is

$$e_{f,n}^{off}(t) = P_1^t \tau_1^t \quad (11)$$

where W is the channel bandwidth, g_f^t is the channel gain from the FU to the NU during time slot t , and σ^2 represents the power of additive white Gaussian noise.

After the initial time $(\tau_0^t + \tau_1^t)$, the NU has received the offloaded tasks from the FU. The NU must then determine the transmission powers P_2^t for relaying the FU's data to the HAP and P_3^t for offloading its own data. The quantity of data relayed by the NU from the FU to the HAP is expressed as

$$d_{n,a1}^{off}(t) = \tau_2^t W \log_2 \left(1 + \frac{P_2^t g_n^t}{\sigma^2} \right) \quad (12)$$

with the associated energy consumption being

$$e_{n,a1}^{off}(t) = P_2^t \tau_2^t \quad (13)$$

Similarly, the amount of data offloaded by the NU to the HAP is

$$d_{n,a2}^{off}(t) = \tau_3^t W \log_2 \left(1 + \frac{P_3^t g_n^t}{\sigma^2} \right) \quad (14)$$

and the corresponding energy consumption is

$$e_{n,a2}^{\text{off}}(t) = P_3^t \tau_3^t \quad (15)$$

Here, $P_2^t, P_3^t \leq P_n^{\text{max}}$ denote the transmission power of the NU, and g_n^t represents signifies the channel gain from the NU to the HAP during the respective time slot.

During each time slot t , the NU is required to process all offloading tasks received from the FU, adhering to the constraint that the amount of data offloaded by the FU does not exceed that of the NU's capability to relay it

$$d_{f,n}^{\text{off}}(t) \leq d_{n,a1}^{\text{off}}(t) \quad (16)$$

2.4. Energy Consumption Model

As illustrated in Figure 2, during the initial phase τ_0^t , the HAP disseminates radio frequency energy to the MDs. The energy harvested by each MD can be represented as

$$e_f^{\text{eh}}(t) = \alpha_1 \tau_0^t P_0^t \quad (17)$$

$$e_n^{\text{eh}}(t) = \alpha_2 \tau_0^t P_0^t \quad (18)$$

where $e_f^{\text{eh}}(t)$ and $e_n^{\text{eh}}(t)$ denote the energy harvested by the FU and NU from the HAP during time slot t , respectively. The coefficients α_1 and α_2 are defined as $\alpha_1 = \mu h_f^t$, $\alpha_2 = \mu h_n^t$, where $0 < \mu < 1$, represents the energy conversion efficiency. P_0^t signifies the RF energy transmission power of the HAP, and the channel power gains from the HAP to the FU and NU are represented by h_f^t and h_n^t , respectively. Assuming block fading, these gains remain consistent within a time slot but fluctuate independently between frames. It is important to note that the WPT from the HAP is the sole energy source for performing computational tasks.

In each time slot t , the energy consumption of the FU and NU must adhere to the following constraints:

$$e_f^{\text{loc}}(t) + e_{f,n}^{\text{off}}(t) \leq \alpha_1 P_0^t \tau_0^t \quad (19)$$

$$e_n^{\text{loc}}(t) + e_{n,a1}^{\text{off}}(t) + e_{n,a2}^{\text{off}}(t) \leq \alpha_2 P_0^t \tau_0^t \quad (20)$$

For a sustainable wireless charging IoT network, all energy for the wireless nodes originates from the RF signals emitted remotely by the HAP. Therefore, controlling the energy consumption of the HAP is of significant importance to the development of the entire IoT network. Consequently, we assume that the time-average energy consumption of the HAP is subject to the following constraints.

$$\overline{P_0^t \tau_0^t} \leq \gamma \quad (21)$$

where $\bar{x} \triangleq \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{x\}$. $P_0^t \tau_0^t$ is the energy consumption for HAP at the t th time slot, and γ is the energy threshold.

2.5. Problem Formulation

For a dynamically changing WPT-MEC network system, maintaining system stability is crucial due to the stochastic arrival of tasks and the dynamic changes in the channel environment [44]. Therefore, we first provide the definition of system queue stability as follows.

Definition 1. (Queue Stability): The task data queue is strong stable if it satisfies [44]

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K \mathbb{E}\{Q_f(t)\} < \infty$$

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K \mathbb{E}\{Q_n(t)\} < \infty$$

In this paper, we aim to design a dynamic offloading algorithm to maximize the long-term average weighted sum computation rate for all MDs, by making decision of time allocation $\tau^t = [\tau_0^t, \tau_1^t, \tau_2^t, \tau_3^t]$, and the power allocation $\mathbf{P}^t = [P_0^t, P_1^t, P_2^t, P_3^t]$ at each time slot t under the long term power consumption constraint of IoT network. Our proposed algorithm should ensure the stability of data queues and work without prior knowledge of random channel conditions and data arrival patterns, optimizing within each time slot. By denoting $\tau = \{\tau^t\}_{t=1}^K$, $\mathbf{P} = \{\mathbf{P}^t\}_{t=1}^K$, the maximization of weighted computation rate problem of WPT-MEC can be formulated as a MSSO problem (P1) as:

$$P1 : \max_{\tau, \mathbf{P}} \quad \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K D_{tot}(t) \quad (22a)$$

$$s.t. \quad \tau_0^t + \tau_1^t + \tau_2^t + \tau_3^t \leq \tau, \quad (22b)$$

$$e_f^{\text{loc}}(t) + e_{f,n}^{\text{off}}(t) \leq \alpha_1 P_0^t \tau_0^t, \quad (22c)$$

$$e_n^{\text{loc}}(t) + e_{n,a1}^{\text{off}}(t) + e_{n,a2}^{\text{off}}(t) \leq \alpha_2 P_0^t \tau_0^t, \quad (22d)$$

$$\overline{P_0^t \tau_0^t} \leq \gamma, \quad (22e)$$

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K \mathbb{E}\{Q_f(t)\} < \infty, \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=1}^K \mathbb{E}\{Q_n(t)\} < \infty, \quad (22f)$$

$$d_f^{\text{loc}}(t) + d_{f,n}^{\text{off}}(t) \leq Q_f(t), \quad (22g)$$

$$d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t) \leq Q_n(t), \quad (22h)$$

$$d_{f,n}^{\text{off}}(t) \leq d_{n,a1}^{\text{off}}(t), \quad (22i)$$

$$\tau_0^t, \tau_1^t, \tau_2^t, \tau_3^t \geq 0, \quad (22j)$$

$$0 \leq p_0^t \leq P_a^{\max}, 0 \leq p_1^t \leq P_f^{\max}, 0 \leq p_2^t, p_3^t \leq P_n^{\max}, \quad (22k)$$

where $D_{tot}(t) = \omega_1 (d_f^{\text{loc}}(t) + d_{f,n}^{\text{off}}(t)) + \omega_2 (d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t))$, ω_1 and ω_2 denotes the fixed weight of the FU and NU, respectively. Constraint (22b) ensures the total time allocation does not exceed the time slot. Constraints (22c) and (22d) represent the energy constraints for the FU and Near NU, respectively. Constraint (22e) captures the average power constraints for the system. Constraints (22f) guarantee the stability of data queues. Constraints (22g) and (22h) denote the maximum data processing limits for the FU and NU within time slot t . Constraint (22i) ensures that the data offloaded from the FU can be processed within the same time slot. Problem (P1) poses a complex stochastic optimization challenge that conventional convex optimization methods cannot easily solve. To address this, we introduce a Lyapunov-based optimization algorithm that transforms the stochastic problem into a deterministic one within each time slot.

3. Algorithm Design

3.1. Lyapunov Optimization

To address the average power constraints, we introduce a virtual energy queue $Y(t+1) = [Y(t) - \gamma + P_0^t \tau_0^t]^+$, where $[x]^+ \triangleq \max\{0, x\}$ [44]. Here $Y(t)$ can be seen as a queue with random "energy arrivals" $P_0^t \tau_0^t$ and a fixed "service rate" γ . Thus, we derive the following Lemma 1.

Lemma 1. *The long-term average power constraints will be met if the virtual queue $Y(t)$ satisfies average rate stability*

$$\lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E}\{Y(t)\} = 0$$

Proof. According to the above formula $Y(t+1) = [Y(t) - \gamma + P_0^t \tau_0^t]^+$, we can conclude that

$$Y(t+1) \geq Y(t) - \gamma + P_0^t \tau_0^t \quad (23)$$

we can expand all the terms of $t \in \{0, 1, 2, \dots, K-1\}$, sum them up, and then take the average, yielding

$$\frac{Y(K) - Y(0)}{K} \geq \frac{1}{K} \sum_{t=0}^{K-1} P_0^t \tau_0^t - \gamma \quad (24)$$

Next, we simultaneously take the expectation on both sides and set K , obtaining:

$$\limsup_{K \rightarrow \infty} \frac{\mathbb{E}\{Y(t)\}}{K} \geq \limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{P_0^t \tau_0^t\} - \gamma \quad (25)$$

By our assumption $\lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E}\{Y(t)\} = 0$, it follows that $\limsup_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{P_0^t \tau_0^t\} - \gamma \leq 0$, i.e., $\overline{P_0^t \tau_0^t} \leq \gamma$, the lemma is proven. \square

By defining a network queue vector $\Theta(t) = \{Q_n(t), Q_f(t), Y(t)\}$, which encompasses the queue lengths for the NU, FU, and the virtual queue, respectively, we can obtain the associated Lyapunov function $L(\Theta(t))$ and the Lyapunov drift $\Delta(\Theta(t))$ as

$$L(\Theta(t)) \triangleq \frac{1}{2} (Q_n(t)^2 + Q_f(t)^2 + Y(t)^2) \quad (26)$$

$$\Delta(\Theta(t)) \triangleq \mathbb{E}\{L(t+1) - L(t) \mid \Theta(t)\} \quad (27)$$

Employing the Lyapunov optimization theory, we drive the drift-plus-penalty as

$$\Delta_V(\Theta(t)) = \Delta(\Theta(t)) - V \mathbb{E}\{D_{\text{tot}}(t) \mid \Theta(t)\} \quad (28)$$

Here, $V > 0$ signifies the penalty's importance weight. The Lyapunov optimization method aims to minimize the upper bound of the drift plus penalty, thereby maximizing the objective function while ensuring queue stability. Optimizing the objective function across each time slot leads to long-term optimality. It's important to note that a higher value of V prioritizes objective maximization, whereas a lower value emphasizes queue stability. To obtain the upper bound of $\Delta_V(\Theta(t))$, we derive the following Lemma 2.

Lemma 2. *At each time slot t , for any control strategy $\{\tau^t, P^t\}$, the one-slot Lyapunov drift-plus-penalty $\Delta_V(\Theta(t))$ is bounded as per the following inequality*

$$\begin{aligned} \Delta_V(\Theta(t)) &\leq B - V \mathbb{E}\{D_{\text{tot}}(t) \mid \Theta(t)\} + Q_f(t) \mathbb{E}\{A_f(t) - d_f^{\text{loc}}(t) - d_{f,n}^{\text{off}}(t) \mid \Theta(t)\} \\ &\quad + Q_n(t) \mathbb{E}\{A_n(t) - d_n^{\text{loc}}(t) - d_{n,a2}^{\text{off}}(t) \mid \Theta(t)\} + Y(t) \mathbb{E}\{P_0^t \tau_0^t - \gamma \mid \Theta(t)\} \end{aligned} \quad (29)$$

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

$$\begin{aligned} B \geq & \frac{1}{2} \mathbb{E} \left\{ \left(d_f^{\text{loc}}(t) + d_{f,n}^{\text{off}}(t) \right)^2 + A_f(t)^2 \mid \Theta(t) \right\} \\ & + \frac{1}{2} \mathbb{E} \left\{ \left(d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t) \right)^2 + A_n(t)^2 \mid \Theta(t) \right\} \\ & + \frac{1}{2} \mathbb{E} \left\{ \gamma^2 + (P_0^t \tau_0^t)^2 \mid \Theta(t) \right\} \end{aligned} \quad (30)$$

Proof. For all $\forall a, b, c \geq 0$, we have the inequality $(\max[a - b, 0] + c)^2 \leq a^2 + b^2 + c^2 + 2a(c - b)$. By using the inequality, we have

$$\begin{aligned} \Delta Q_f(t) &= \frac{1}{2} \left(Q_f(t+1)^2 - Q_f(t)^2 \right) \\ &\leq \frac{\left(d_f^{\text{loc}}(t) + d_{f,n}^{\text{off}}(t) \right)^2 + A_f(t)^2}{2} + Q_f(t) \left(A_f(t) - d_f^{\text{loc}}(t) - d_{f,n}^{\text{off}}(t) \right) \end{aligned} \quad (31)$$

$$\begin{aligned} \Delta Q_n(t) &= \frac{1}{2} \left(Q_n(t+1)^2 - Q_n(t)^2 \right) \\ &\leq \frac{\left(d_n^{\text{loc}}(t) + d_{n,a2}^{\text{off}}(t) \right)^2 + A_n(t)^2}{2} + Q_n(t) \left(A_n(t) - d_n^{\text{loc}}(t) - d_{n,a2}^{\text{off}}(t) \right) \end{aligned} \quad (32)$$

$$\begin{aligned} \Delta Y(t) &= \frac{1}{2} \left(Y(t+1)^2 - Y(t)^2 \right) \\ &\leq \frac{(P_0^t \tau_0^t)^2 + \gamma^2}{2} + Y(t) (P_0^t \tau_0^t - \gamma) \end{aligned} \quad (33)$$

Upon combining inequalities (31), (32), and (33), the resulting expression yields the upper bound of the Lyapunov drift-plus-penalty. \square

By applying the drift-plus-penalty minimization technique, and eliminating the constant term observable at the start of time slot t in (29), we can obtain the optimal solution to problem (P1) by solving the following problem in each individual time slot.

$$\begin{aligned} P2 : \min_{\tau^t, P^t} \quad & Y(t) P_0^t \tau_0^t - [Q_f(t) + \omega_1 V] \left\{ \tau_1^t W \log_2 \left(1 + \frac{P_1^t g_f^t}{\sigma^2} \right) + d_f^{\text{loc}}(t) \right\} \\ & - [Q_n(t) + \omega_2 V] \left\{ \tau_3^t W \log_2 \left(1 + \frac{P_3^t g_n^t}{\sigma^2} \right) + d_n^{\text{loc}}(t) \right\} \end{aligned} \quad (34a)$$

$$s.t. \quad (22b) - (22d), (22g) - (22k) \quad (34b)$$

Due to the non-convex constraints (34b), (P2) remains a non-convex problem. We introduce auxiliary variables $\varphi_0 = p_0^t \tau_0^t$, $\varphi_1 = p_1^t \tau_1^t$, $\varphi_2 = p_2^t \tau_2^t$, $\varphi_3 = p_3^t \tau_3^t$ here. According to equation (22j), we have $0 \leq \varphi_0 \leq \tau_0 P_a^{\max}$, $0 \leq \varphi_1 \leq \tau_1 P_f^{\max}$, $0 \leq \varphi_2 \leq \tau_2 P_n^{\max}$, $0 \leq \varphi_3 \leq \tau_3 P_n^{\max}$. We denote $\boldsymbol{\varphi} = [\varphi_0, \varphi_1, \varphi_2, \varphi_3]$. To simplify the mathematical expressions, we omit t here. So (P2) can be equivalently substituted as

$$P3 : \max_{\tau, \varphi} \quad Y\varphi_0 + C_1 \left\{ \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) + d_f^{\text{loc}} \right\} + C_2 \left\{ \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right) + d_n^{\text{loc}} \right\} \quad (35a)$$

$$\text{s.t.} \quad \tau_0 + \tau_1 + \tau_2 + \tau_3 \leq \tau, \quad (35b)$$

$$\varphi_1 + k(f_f)^3 T \leq \alpha_1 \varphi_0, \quad (35c)$$

$$\varphi_2 + \varphi_3 + k(f_n)^3 T \leq \alpha_2 \varphi_0, \quad (35d)$$

$$\tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) \leq \tau_2 W \log_2 \left(1 + \frac{\varphi_2 B}{\tau_2} \right), \quad (35e)$$

$$\frac{f_f T}{\phi_f} + \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) \leq Q_f, \quad (35f)$$

$$\frac{f_n T}{\phi_n} + \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right) \leq Q_n, \quad (35g)$$

$$0 \leq \varphi_0 \leq \tau_0 P_a^{\text{max}}, 0 \leq \varphi_1 \leq \tau_1 P_f^{\text{max}}, \quad (35h)$$

$$0 \leq \varphi_2 \leq \tau_2 P_n^{\text{max}}, 0 \leq \varphi_3 \leq \tau_3 P_n^{\text{max}}, \quad (35i)$$

(22j)

The coefficients are defined as $C_1 = -[Q_f + \omega_1 V]$ and $C_2 = -[Q_n + \omega_2 V]$, where $A = \frac{g_f}{\sigma^2}$, and $B = \frac{g_n}{\sigma^2}$. Owing to the non-convex constraint (35e), problem (P3) remains non-convex. To address this, we introduce auxiliary variables ψ_1 and ψ_2 , defined such that $\psi_1 \leq \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right)$, $\psi_2 \leq \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right)$. By making these definitions, problem (P3) can be reformulated in terms of these auxiliary variables as follows:

$$P4 : \max_{\tau, \varphi} \quad Y\varphi_0 + C_1 \left\{ \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) + d_f^{\text{loc}} \right\} + C_2 \left\{ \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right) + d_n^{\text{loc}} \right\} \quad (36a)$$

$$\text{s.t.} \quad \tau_0 + \tau_1 + \tau_2 + \tau_3 \leq \tau, \quad (36b)$$

$$\varphi_1 + k(f_f)^3 T \leq \alpha_1 \varphi_0, \quad (36c)$$

$$\varphi_2 + \varphi_3 + k(f_n)^3 T \leq \alpha_2 \varphi_0, \quad (36d)$$

$$\psi_1 \leq \tau_2 W \log_2 \left(1 + \frac{\varphi_2 B}{\tau_2} \right), \quad (36e)$$

$$\psi_1 \leq \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right), \quad (36f)$$

$$\psi_1 \leq Q_f - \frac{f_f T}{\phi_f}, \quad (36g)$$

$$\psi_2 \leq \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right), \quad (36h)$$

$$\psi_2 \leq Q_n - \frac{f_n T}{\phi_n}, \quad (36i)$$

$$0 \leq \varphi_0 \leq \tau_0 P_a^{\text{max}}, 0 \leq \varphi_1 \leq \tau_1 P_f^{\text{max}}, \quad (36j)$$

$$0 \leq \varphi_2 \leq \tau_2 P_n^{\text{max}}, 0 \leq \varphi_3 \leq \tau_3 P_n^{\text{max}}, \quad (36k)$$

(22j),

$\psi_1 = \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right), \psi_2 = \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right)$, when the problem (P4) reaches the optimal solution, it aligns with the original problem (P3). Specifically, for constraint (36e), $\tau_2 W \log_2 \left(1 + \frac{\varphi_2 B}{\tau_2} \right)$ is a concave function of τ_2 since the perspective operation applied to $W \log_2(1 + \varphi_2 B)$ preserves convexity [45]. Thus, constraint (36e) is convex, making problem (P4) a convex optimization problem. To further analyze the solution, we will employ convex optimization tools, such as CVX [46].

We introduce an efficient dynamic task offloading algorithm with user assistance to tackle Problem 4. Additionally, we apply the Lagrange method to gain meaningful insights into the optimal solution's properties.

Theorem 1. Given non-negative Lagrange multipliers $\lambda_i, i = 1, 2, \dots, 12$, the optimal power allocation $P = [P_0, P_1, P_2, P_3]$ must fulfill certain conditions

$$P_0 = \begin{cases} 0, & \text{if } \tau_0 = 0 \\ [P_a^{\max}]^+, & \text{others} \end{cases} \quad (37a)$$

$$P_1 = \begin{cases} 0, & \text{if } \tau_1 = 0 \\ \left[\frac{W(Q_f + V + \lambda_5)}{\ln 2(\lambda_2 + \lambda_{10})} - \frac{\sigma^2}{g_f} \right]^+, & \text{others} \end{cases} \quad (37b)$$

$$P_2 = \begin{cases} 0, & \text{if } \tau_2 = 0 \\ \left[\frac{\lambda_4 W}{\ln 2(\lambda_3 + \lambda_{11})} - \frac{\sigma^2}{g_n} \right]^+, & \text{others} \end{cases} \quad (37c)$$

$$P_3 = \begin{cases} 0, & \text{if } \tau_3 = 0 \\ \left[\frac{W(Q_n + V + \lambda_7)}{\ln 2(\lambda_3 + \lambda_{12})} - \frac{\sigma^2}{g_n} \right]^+, & \text{others} \end{cases} \quad (37d)$$

Proof. Let $\lambda_i \geq 0 (i = 1, 2, \dots, 12)$ for denote the Lagrange multipliers corresponding to the constraints. The Lagrangian function for problem (P4), constructed based on these multipliers, is as follow

$$\begin{aligned} L(\tau, \varphi, \lambda) = & Y(t)\varphi_0 + C_1 \left\{ \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) + \frac{f_f T}{\phi_f} \right\} + C_2 \left\{ \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right) + \frac{f_n T}{\phi_n} \right\} \\ & + \lambda_1 [\tau_0^t + \tau_1^t + \tau_2^t + \tau_3^t - \tau] \\ & + \lambda_2 [\varphi_1 + k(f_f)^3 T - \alpha_1 \varphi_0] \\ & + \lambda_3 [\varphi_2 + \varphi_3 + k(f_n)^3 T - \alpha_2 \varphi_0] \\ & + \lambda_4 \left[\psi_1 - \tau_2 W \log_2 \left(1 + \frac{\varphi_2 B}{\tau_2} \right) \right] \\ & + \lambda_5 \left[\psi_1 - \tau_1 W \log_2 \left(1 + \frac{\varphi_1 A}{\tau_1} \right) \right] \\ & + \lambda_6 \left[\psi_1 - Q_f - \frac{f_f T}{\phi_f} \right] \\ & + \lambda_7 \left[\psi_2 - \tau_3 W \log_2 \left(1 + \frac{\varphi_3 B}{\tau_3} \right) \right] \\ & + \lambda_8 \left[\psi_2 - Q_n - \frac{f_n T}{\phi_n} \right] \\ & + \lambda_9 [\varphi_0 - \tau_0 P_a^{\max}] \\ & + \lambda_{10} [\varphi_1 - \tau_1 P_f^{\max}] \\ & + \lambda_{11} [\varphi_2 - \tau_2 P_n^{\max}] \\ & + \lambda_{12} [\varphi_3 - \tau_3 P_n^{\max}] \end{aligned} \quad (38a)$$

We can use the first-order optimality conditions. Taking the derivative of the Lagrangian function yields

$$\varphi_0 = [\tau_0 P_a^{\max}]^+, \quad (39a)$$

$$\varphi_1 = \left[\frac{\tau_1 W (Q_f + V + \lambda_5)}{\ln 2 (\lambda_2 + \lambda_{10})} - \frac{\tau_1 \sigma^2}{g_f} \right]^+, \quad (39b)$$

$$\varphi_2 = \left[\frac{\tau_2 \lambda_4 W}{\ln 2 (\lambda_3 + \lambda_{11})} - \frac{\tau_2 \sigma^2}{g_n} \right]^+, \quad (39c)$$

$$\varphi_3 = \left[\frac{\tau_3 W (Q_n + V + \lambda_7)}{\ln 2 (\lambda_3 + \lambda_{12})} - \frac{\tau_3 \sigma^2}{g_n} \right]^2, \quad (39d)$$

By examining the first-order derivatives, we can establish the necessary conditions for optimality. The relationship between the auxiliary variables and the original variables is leveraged to derive the theorem. \square

According to this theorem, we can infer that during the process of radio frequency energy transfer, higher power leads to better results. As the value of W increases, both FU and NU entities are more motivated to offload data, which results in a reduction of the computational tasks performed locally. Furthermore, an increase in V prompts FU and NU to offload a larger portion of their tasks. This shift towards offloading is driven by the desire to enhance the computation rate, which in turn leads the MDS to increase the volume of data offloaded to meet its objectives.

The process of solving the original MSSO problem, denoted as (P1), is encapsulated within Algorithm 1.

Algorithm 1: User-Assisted Dynamic Resource Allocation Algorithm

Input: the task arrival queue $A_f(t)$ and $A_n(t)$; the channel gain $\{h_f^t, h_n^t, g_f^t, g_n^t\}$.

- 1 **Initialize:** $t \leftarrow 0, Q_f(0), Q_n(0), Y(0)$.
- 2 **for** $t = 1 : K$ **do**
- 3 Observe the system state to obtain $Q_f(t), Q_n(t), Y(t)$;
- 4 Calculate $d_f^{loc}(t)$ with (5) and $d_n^{loc}(t)$ with (7);
- 5 **if** $d_f^{loc}(t) < Q_f(t)$ or $d_n^{loc}(t) < Q_n(t)$ **then**
- 6 Determine $d_f^{loc}(t), d_f^{loc}(t), d_f^{loc}(t), d_f^{loc}(t)$ according to (5)-(7);
- 7 cvx_begin
- 8 Minimize P4
- 9 Subject to (36b)-(36k)
- 10 cvx_end
- 11 **else**
- 12 Complete local computation
- 13 **Output:** The optimal time allocation $\tau^{t,*}$ and transmit power $P^{t,*}$ at slot t ;
- 14 Update the data queue $Q_f(t)$ with (1);
- 15 Update the data queue $Q_n(t)$ with (2);
- 16 Update the virtual queue $Y(t)$;

3.2. Algorithm Complexity Analysis

We have employed the Lyapunov algorithm, which not only ensures the stability of the system but also allows us to effectively decompose the complex overall problem into multiple sub-problems (P2). By implementing Algorithm 1, we are able to solve (P2) within each time slot. Therefore, the solution complexity at each time slot is crucial in determining the overall performance and responsiveness of

the algorithm. When solving the problem (P4), we have utilized the interior-point method, which possesses a computational complexity of approximately $O(n^{3.5} \log(1/\epsilon))$.

In this context, n denotes the total of decision variables, and ϵ reflects the degree of precision in computation. In our algorithm, with $n = 8$, the number of decision variables is sufficiently small. This design not only ensures the efficiency of the algorithm but also enables us to complete the optimization of resource allocation within a reasonable time, thus meeting the performance requirements in practical applications.

3.3. Algorithm Performance Analysis

In this section, we demonstrate that the proposed scheme can achieve an optimal long-term time-average solution. First, we establish a key assumption as follows

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} D_{\text{tot}}(t) = \overline{D(t)} \quad (40)$$

Subsequently, we deduce that the expected value will also converge to the same set of solutions

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{D_{\text{tot}}(t)\} = \overline{D(t)} \quad (41)$$

Furthermore, we establish the existence of an optimal solution founded on the existing conditions of the queue as follows.

Lemma 3. *Should problem (P1) be solvable, there is a set of decisions $\{\tau^t, P^t\}^*$ that adhere to the conditions $\forall t, \epsilon > 0$*

$$\mathbb{E}\left\{A_f(t) - d_f^{\text{loc},*}(t) - d_f^{\text{off},*}(t)\right\} \leq \epsilon \quad (42)$$

$$\mathbb{E}\left\{A_n(t) - d_n^{\text{loc},*}(t) - d_{n,a2}^{\text{off},*}(t)\right\} \leq \epsilon \quad (43)$$

$$\mathbb{E}\left\{P_0^{t,*} \tau_0^{t,*} - \gamma\right\} \leq \epsilon \quad (44)$$

Here, the asterisk $*$ denotes the value associated with the optimal solution.

Proof. Here we omit the proof details for brevity. See the parts 4 and 5 of [44]. \square

Theorem 2. *The optimal long-term average weighted computation rate derived from (P1) is bounded below by a lower bound, which is independent of time and space. The algorithm is capable of achieving the following solutions:*

1. $\overline{D(t)} \geq \mathbb{E}\{D_{\text{tot}}^*(t)\} - B/V$,
2. All queues $Q_f(t)$, $Q_n(t)$, $Y(t)$ are mean rate stable, thereby satisfying the constraints.

Proof. For any $\epsilon > 0$, we consider the policy and queue state as defined in equations (40). Given that the values $P_0^{t,*} \tau_0^{t,*}$, $d_f^{\text{loc},*}(t)$, $d_n^{\text{loc},*}(t)$, $d_{n,a2}^{\text{off},*}(t)$, $d_f^{\text{off},*}(t)$, are independent of the queue status $\Theta(t)$, we can deduce that

$$\mathbb{E}\left\{A_f(t) - d_f^{\text{loc},*}(t) - d_f^{\text{off},*}(t) \mid \Theta(t)\right\} = \mathbb{E}\left\{A_f(t)\right\} - \mathbb{E}\left\{d_f^{\text{loc},*}(t) + d_f^{\text{off},*}(t)\right\} \leq \epsilon \quad (45a)$$

$$\mathbb{E}\left\{A_n(t) - d_n^{\text{loc},*}(t) - d_{n,a2}^{\text{off},*}(t) \mid \Theta(t)\right\} = \mathbb{E}\left\{A_n(t)\right\} - \mathbb{E}\left\{d_n^{\text{loc},*}(t) + d_{n,a2}^{\text{off},*}(t)\right\} \leq \epsilon \quad (45b)$$

$$\mathbb{E}\left\{P_0^{t,*} \tau_0^{t,*} - \gamma \mid \Theta(t)\right\} = \mathbb{E}\left\{P_0^{t,*} \tau_0^{t,*}\right\} - \mathbb{E}\{\gamma\} \leq \epsilon \quad (45c)$$

By integrating these results into equation (29) and taking $\varepsilon \rightarrow 0$, we obtain

$$\begin{aligned} \Delta_V(\Theta(t)) &\leq B - V\mathbb{E}\{D_{\text{tot}}^*(t)\} + Q_f(t)\mathbb{E}\{A_f(t) - d_f^{\text{loc},*}(t) - d_{f,n}^{\text{off},*}(t)\} \\ &\quad + Q_n(t)\mathbb{E}\{A_n(t) - d_n^{\text{loc},*}(t) - d_{n,a2}^{\text{off},*}(t)\} + Y(t)\mathbb{E}\{P_0^{t,*}\tau_0^{t,*} - \gamma\} \\ &\leq B - V\mathbb{E}\{D_{\text{tot}}^*(t)\} \end{aligned} \quad (46)$$

Utilizing the iterated expectation, and summing the aforementioned inequality over the time horizon $t \in \{0, 1, \dots, T-1\}$, we derive the following result

$$\begin{aligned} \mathbb{E}\{L(\Theta(K))\} - \mathbb{E}\{L(\Theta(0))\} - V \sum_{t=0}^{K-1} \mathbb{E}\{D_{\text{tot}}(t)\} \\ \leq K[B - V\mathbb{E}\{D_{\text{tot}}^*(t)\}] \end{aligned} \quad (47a)$$

By dividing both sides of equation (47a) by VK , applying Jensen's inequality, and considering that $\mathbb{E}\{L(\Theta(t))\} \geq 0$, we obtain

$$-\frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{D_{\text{tot}}(t)\} \leq \frac{B}{V} - \mathbb{E}\{D_{\text{tot}}^*(t)\} \quad (48)$$

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

$$-\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{D_{\text{tot}}(t)\} \leq \frac{B}{V} - \mathbb{E}\{D_{\text{tot}}^*(t)\} \quad (49)$$

From Equation (41), we have

$$-\overline{D(t)} \leq \frac{B}{V} - \mathbb{E}\{D_{\text{tot}}^*(t)\} \quad (50)$$

Furthermore we obtain

$$\overline{D(t)} \geq \mathbb{E}\{D_{\text{tot}}^*(t)\} - \frac{B}{V} \quad (51)$$

□

Theorem 3. A defined upper limit confines the time-averaged sum of queue lengths

$$Q^\Sigma \leq \frac{B}{\varepsilon} + V \frac{\overline{D(t)}}{\varepsilon} \quad (52)$$

Proof. By employing the method of iterated expectations and applying telescoping sums iteratively for each $t \in \{0, 1, \dots, T-1\}$, we can derive

$$\begin{aligned} \mathbb{E}\{L(\Theta(K))\} - \mathbb{E}\{L(\Theta(0))\} - V\mathbb{E}\{D_{\text{tot}}(t) \mid \Theta(t)\} \\ \leq KB - \varepsilon \sum_{t=0}^{K-1} \mathbb{E}\{Q_n(t) + Q_f(t) + Y(t)\} \end{aligned} \quad (53)$$

Dividing both sides of (53) by $K\varepsilon$, and considering $T \rightarrow \infty$, we rearrange the terms to obtain the desired result

$$\frac{B}{\varepsilon} - \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{t=0}^{K-1} \mathbb{E}\{Q_n(t) + Q_f(t) + Y(t)\} + V \frac{\overline{D(t)}}{\varepsilon} \geq 0 \quad (54)$$

Specifically

$$Q^\Sigma \leq \frac{B}{\varepsilon} + V \frac{\overline{D(t)}}{\varepsilon} \quad (55)$$

□

Theorems 2 and 3 underpin our proposed algorithm by establishing that as V increases, the computation rate $\overline{D(t)}$ improves at a rate of $O(1/V)$, whereas the queue length grows at a rate of $O(V)$. This suggests that by choosing an appropriately large value for V , we can achieve an optimal $\overline{D(t)}$. Furthermore, the time-averaged queue length, denoted as Q^Σ , is shown to increase linearly with V . This linear relationship implies a trade-off $[O(1/V), O(V)]$ between the optimized computation rate and the queue length. In line with Little's Law [47], which posits that delay is proportional to the time-averaged length of the data queue. In other words, our proposed algorithm enables a trade-off between $\overline{D(t)}$ and the average network latency.

4. Simulation Results

In this section, plenty of numerical simulations are performed to assess the efficiency of our proposed algorithm. Our experiments were conducted on a computational platform with an Intel(R) Xeon(R) Gold 6148 CPU 2.40 GHz, 20 cores and four GeForce RTX 3070 GPUs. In our simulations, we employed a free-space path-loss model to depict the wireless channel characteristics [48]. The averaged channel gain \bar{h} here is denoted as

$$\bar{h} = A_d \left(\frac{3 \times 10^8}{4\pi f_c d_i} \right)^{d_e}, \quad (56)$$

where A_d denotes the antenna gain, f_c denotes the carrier frequency, d_e denotes the path loss exponent, and d_i in meters denotes the distance between two nodes. The time-varying WPT and task offloading channel gains are represented by the vector $\mathbf{h}^t = [a_1^t h_m^t, a_2^t h_h^t, a_3^t g_m^t, a_4^t g_h^t]$, adhering to the Rayleigh fading channel model. In this model, the random channel fading factors a_i are characterized by an exponential distribution with a unit mean, capturing the variability inherent in wireless communication channels. For the sake of simplicity, we assume that the vector of fading factors $[a_1^t, a_2^t, a_3^t, a_4^t]$ is constant and equal to $[1.0, 1.0, 1.0, 1.0]$ for any given time slot, implying that the channel gains are static within that slot. The interval between task arrivals at FU and NU both follow an exponential distribution with constant average rates λ_f and λ_n respectively. The parameters are all listed in Table 2.

Table 2. Simulation Parameters.

Symbol	Value
Time slot length	1 s
Noise power σ^2	10^{-4} W
Distance between the HAP and the FU d_{hf}	230 m
Distance between the FU and the NU d_{fn}	140 m
Distance between the HAP and the NU d_{hn}	200 m
CPU frequency of FU f_f	160 MHz
CPU frequency of NU f_n	220 MHz
CPU cycles to compute 1 bit task of FU ϕ_f	180 cycles/bit
CPU cycles to compute 1 bit task of NU ϕ_n	200 cycles/bit
Equal computing efficiency parameter κ_s	10^{-8}
Weight of the computation rate of FU ω_1	0.55
Weight of the computation rate of NU ω_2	0.45
the antenna gain in channel model A_d	3
the carrier frequency in channel model f_c	915 MHz
the path loss exponent in channel model d_e	3

4.1. Impact of System Parameters on Algorithm Performance

Figure 3 illustrates trend for the average task computation rate D_{tot} and the average task queue lengths of FU and NU over a 5000 time slots period. The the task arrival rates for FU and NU are set at 1.75 Mbps and 2 Mbps, respectively. Initially, D_{tot} is low, but it rapidly increases and eventually stabilizes as time progresses. This initial surge in D_{tot} is due to the system's initial adjustment to the initial task queue fluctuations, which demands more intensive processing for FU tasks, resulting in increased energy consumption and a temporary reduction in the overall computation rate. Moreover, the average queue length decreases and stabilizes, reflecting the system's ability to self-regulate and reach a steady state.

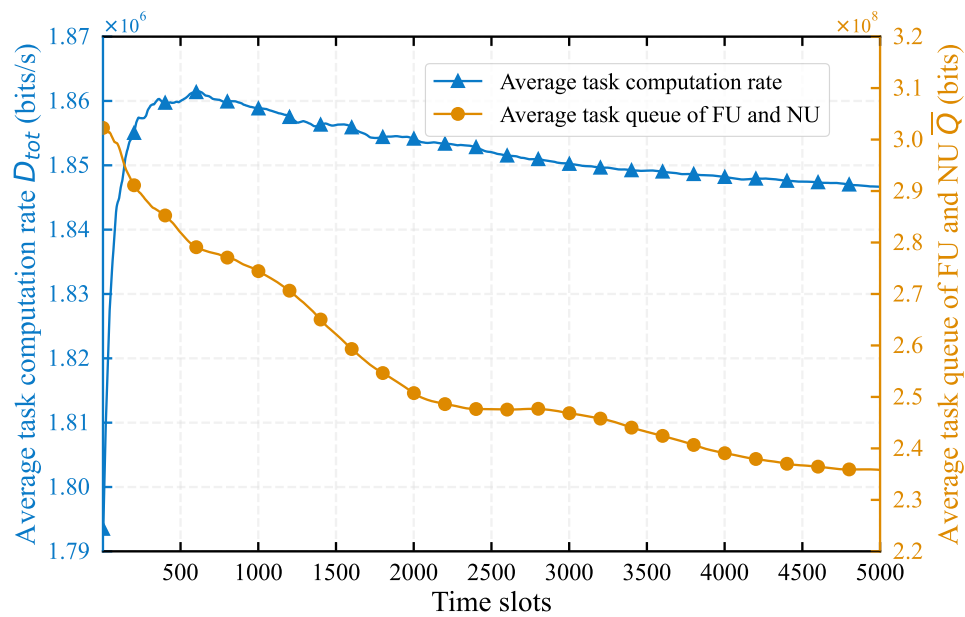


Figure 3. Average task computation rate and average task queue length over time slots.

Figure 4 demonstrates the average task computation rate of our proposes algorithm under different control parameter $V = 100, 300, 500, 700$. The results show that the average task computation rates converge similarly across different V . Notably, as V increases, there is a corresponding increase in the average task computation rates. This trend is attributed to the fact that a larger V compels the algorithm to prioritize computation rates over queue stability, which is consistent with theoretical analysis, which is corresponding to Theorems 2. Here, the parameter V serves as a balancing factor between the task computation rate and queue stability, reflecting a trade-off that is consistent with our theoretical predictions.

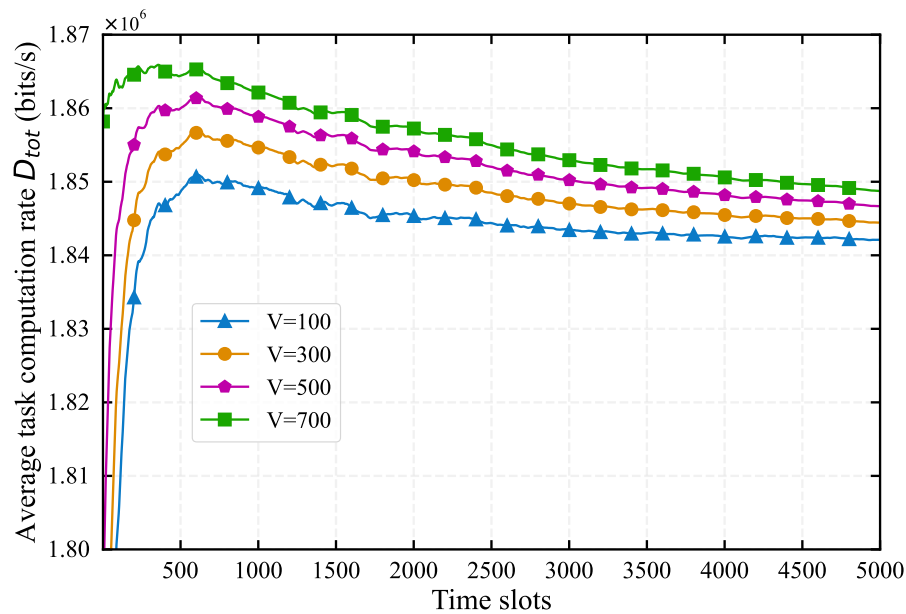


Figure 4. Average task computation rates with different control parameter V .

Figure 5 shows the trend of average task queue lengths of FU and NU after with different V . As V increases from 100 to 900, the task queue lengths of FU and HD declines from 3.1×10^8 bits to 2.9×10^8 bits and from 3.11×10^8 bits to 2.84×10^8 bits respectively. In the user-assisted offloading paradigm, processing a task from NU involves only a single offloading step, which is markedly more efficient than the two-step offloading data from FU. Consequently, with the increasing of V , indicating a shift in the algorithm's focus towards prioritizing computation rates over queue stability, the algorithm will naturally prefer to offload tasks from NU, resulting in a more rapid decline in the NU's task queue length. By tuning the value of V , an optimal balance in task processing between FU and HU can be effectively achieved, reflecting the algorithm's adaptability to different operational priorities.

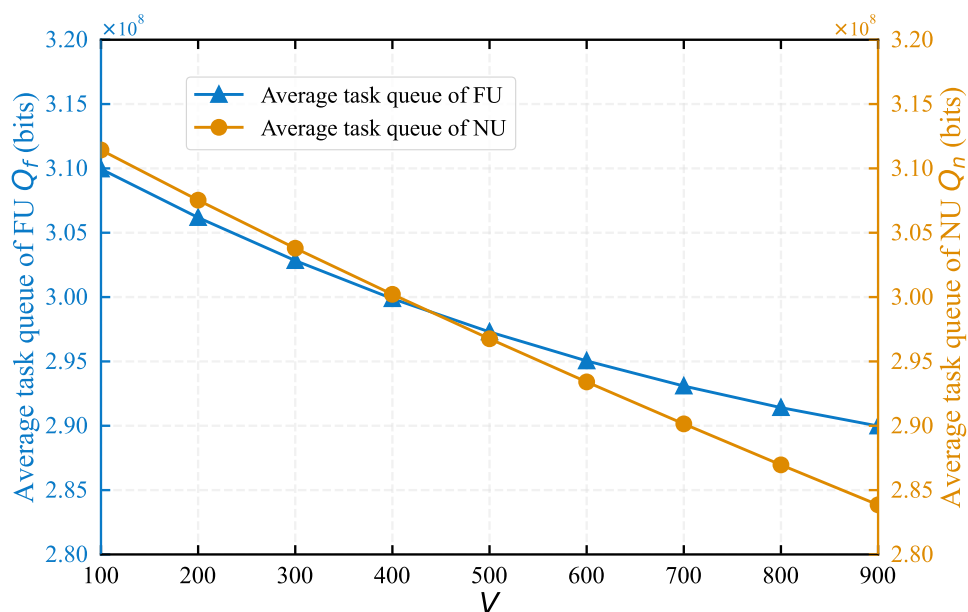


Figure 5. Task queue lengths with different control parameter V .

Figure 6 evaluates the impact of the average energy constraint γ on system performance with V fixed at 500. As γ increases from 1.75 joule to 2.25 joule, the average task computation rate rises from 1.79×10^6 bits/s to 1.86×10^6 bits/s, while the average task queue length of FU and NU decreases from 4×10^8 bits to 2.5×10^8 bits. This reduction in queue length and increase in computation rate are attributed to the higher energy availability for WPT, enabling FU and NU to offload tasks more effectively. Notably, after the average energy constraint γ reaching 2.1 joule, the variation of task computation rate and task queue is reduced. This observation suggests that there is an upper bound to the energy consumption of our algorithm. Beyond this threshold, additional energy has minimal impact on performance enhancement. Hence, energy constraint γ as a critical parameter, significantly influences both the data processing rates and the stability of task queues within the system. The findings underscore the importance of energy management in optimizing system performance.

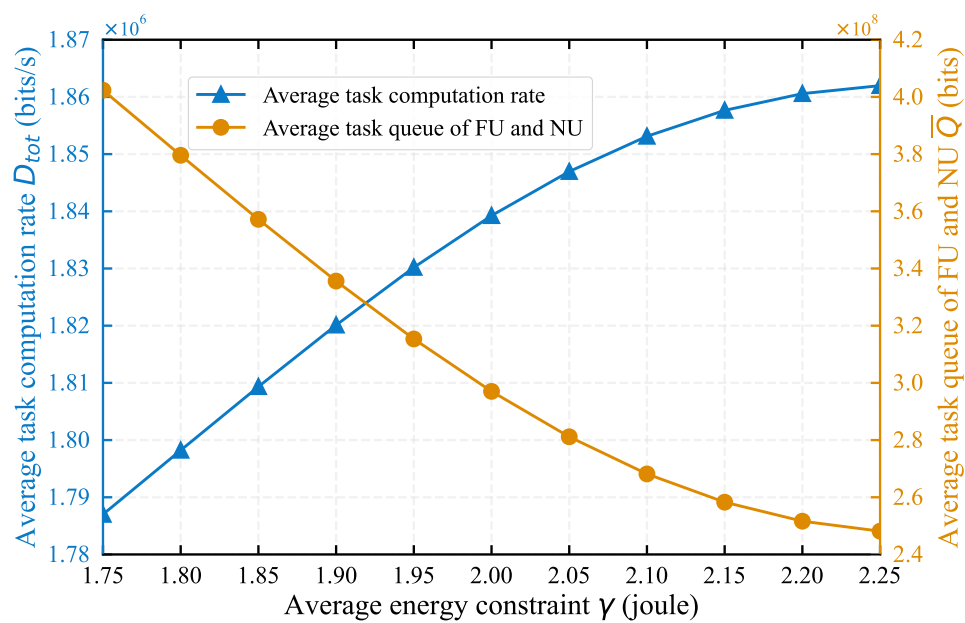


Figure 6. Average task computation rate and task queue length with different energy constraint γ .

Figure 7 presents the total system energy consumption of our proposed algorithm under different values of the parameter V , specifically $V = 300$ and $V = 700$. Initially, the total energy consumption exhibits substantial fluctuations. However, with the progression of time, the system's energy consumption stabilizes and hovers around the average energy constraint γ after approximately 2500 time slots. Notably, an elevated V value is correlated with higher average energy consumption, as the algorithm pay more attention to the system's computation rate, consequently incurring greater energy costs. Figure 7 highlight the algorithm's efficacy in managing average energy consumption, a critical feature for the sustainability of IoT networks.

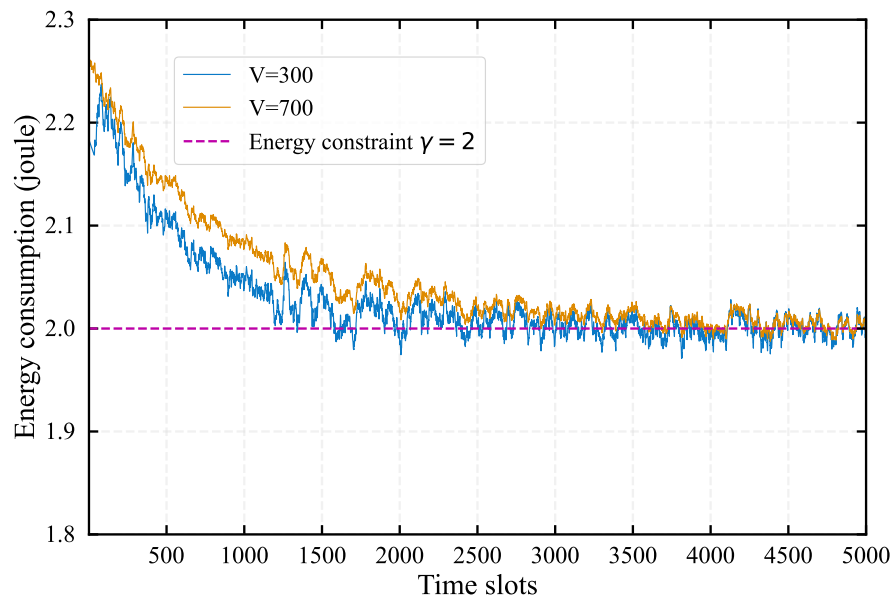


Figure 7. Convergence performance of energy consumption with different parameter V .

In Figure 8, we evaluate the offloading power across varying bandwidths W , ranging from 0.96×10^6 Hz to 1.05×10^6 Hz. It is observed that all the offloading powers increases as bandwidth W escalates. Consistent with the results of the analysis in Theorem 1, the increase in bandwidth makes the system more inclined to perform task offloading, reflected in the increase in offloading power.

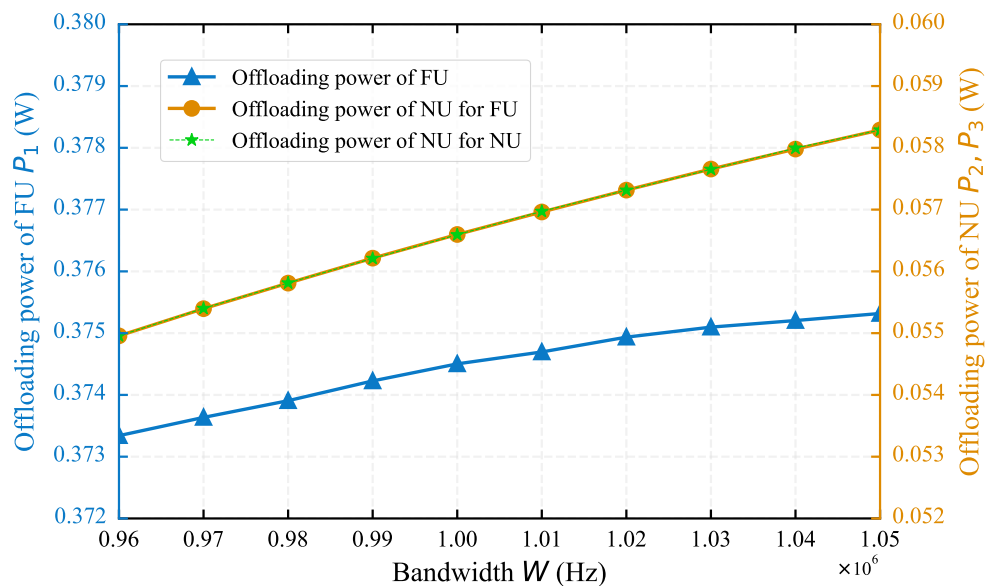


Figure 8. Offloading power of FU and NU with different Bandwidth W .

4.2. Comparison with Baseline Algorithms

To evaluate the performance of our propose algorithm, we choose the the following three representative benchmarks as the baseline algorithms.

(1) All offloading scheme: Both FU and NU do not perform local computing and consumes all the energy for task offloading.

(2) No cooperation scheme: FU offloads tasks directly to HAP without soliciting assistance from NU, similar to the method in [49].

(3) No lyapunov scheme: Disregarding the dynamics of task queues and the energy queue, this scheme focuses solely on maximizing the average task computation rate, similar to the Myopic method in [30]. To ensure a fair comparison, we constrain the energy consumption of each time slot by solving.

$$\begin{aligned} \max \quad & D_{tot}(t) \\ \text{s.t.} \quad & (22b) - (22d), (22g) - (22k) \\ & P_0^t \tau_0^t \leq \gamma \end{aligned} \quad (57)$$

Figure 9 shows the average task computation rates under four schemes over a 5000 time-slots period, with the control parameter V set to 500. All schemes converge after 1000 time slots. Our proposed algorithm achieves the best task computation rate after convergence, outperforming the other three by 0.8%, 3.9%, and 4.1% respectively. Our algorithm's key strengths lie not only in achieving the highest data processing rate but also in ensuring the stability of the system queues. This prevents excessively long queues that could lead to prolonged response times and a negative user experience. The no-Lyapunov scheme, while achieving the second-highest computation rate, neglects queue stability in its pursuit of maximizing computation speed. This oversight can lead to system instability, prolonged user service times, and potential system failure. The all-offloading scheme, relying solely on edge computing, consumes more energy and thus underperforms in energy-limited scenarios. In the no-cooperation scheme, the system initially benefits from a high computation rate due to the NU's lack of assistance to the FU. But as the NU's tasks are completed and its resources are no longer available, the average computation rate falls sharply. The FU's communication with the HAP is further impeded by the dual proximity effect, causing a notable decline in the system's long-term computation performance.

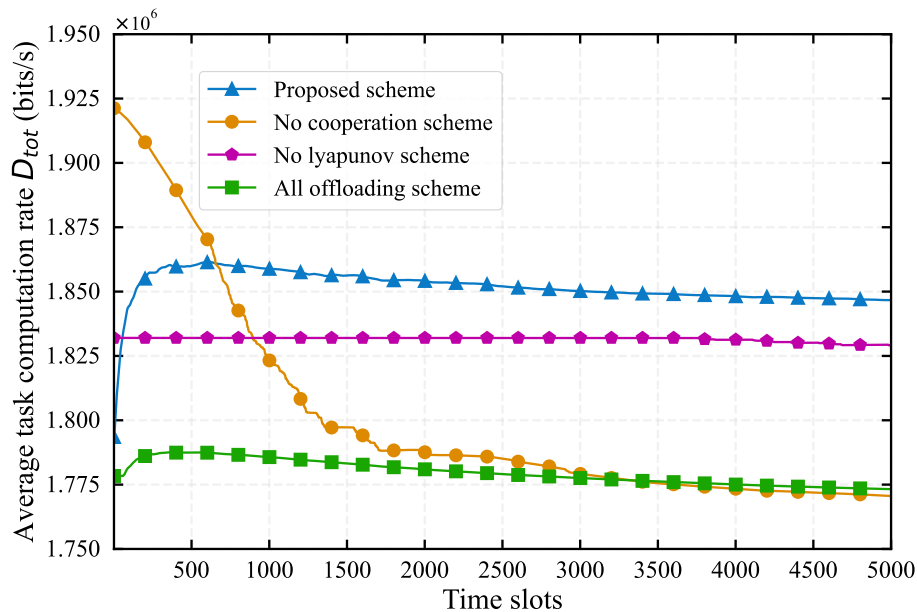


Figure 9. Average task computation rates in different schemes over time slots.

Figure 10 show the impact of varying network bandwidth W from 0.96×10^6 Hz to 1.05×10^6 Hz on the performance of different schemes. As W increase, task computation rates for all schemes rise, reflecting improved transmission efficiency for wireless power transfer and task offloading. This allows HAP to handle more offloaded tasks, highlighting the critical role of bandwidth in system

performance. Notably, our proposed scheme consistently outperforms others across all bandwidth levels, showcasing its adaptability and robustness in varying network conditions.

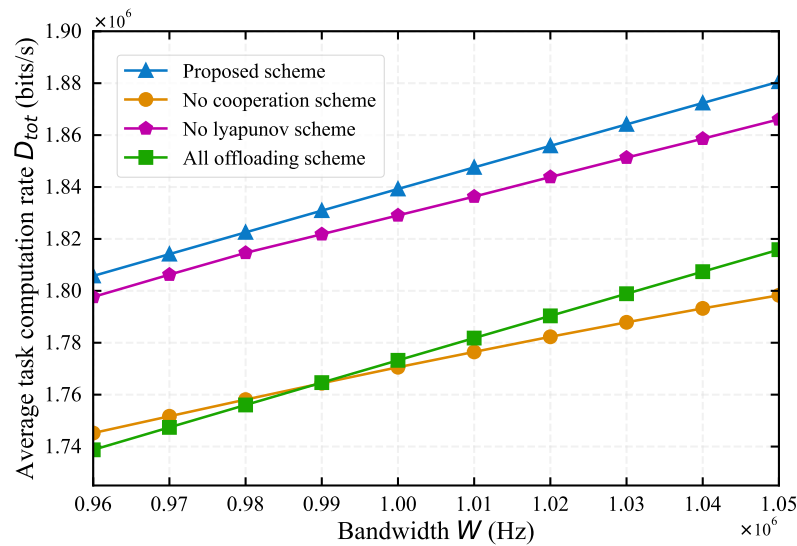


Figure 10. Average computation rates in different schemes with different bandwidth W .

Figure 11 evaluates how the distance between FU and NU affects the performance of all four schemes, with distances varying from 120 m to 160 m. We observe that as the distance increases, the computation rates for both our proposed scheme and the all-offloading scheme decrease. This suggests that proximity plays a crucial role in task offloading efficiency. In contrast, the no-cooperation scheme shows a stable computation rate, consistent with its design that excludes task offloading between FU and NU. Interestingly, the no-Lyapunov scheme performs best at a distance of about 140 meters. However, its performance drops as the distance decreases, contrary to the expectation that a shorter distance would enhance task offloading from FU to NU. This unexpected trend is likely due to instances where the FU's task queue depletes faster than new tasks arrive, leading to lower computation rates for the no-Lyapunov scheme. This highlights the importance of balancing task computation rates with queue stability in system design.

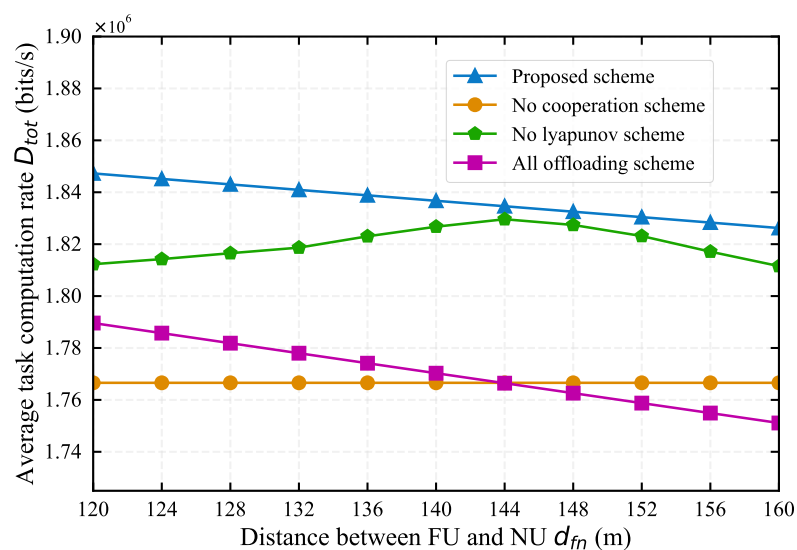


Figure 11. Average computation rates in different schemes with different distances between FU and NU.

In Figure 12, we evaluate the performance of four schemes as the task arrival rate of NU varies from 1.85×10^6 bits/s to 2.35×10^6 bits/s. Our proposed scheme's task computation rate demonstrates a modest increase and maintains the highest computation rate as tasks arrive more rapidly. This trend underscores the scheme's robustness across diverse scenarios. Correspondingly, the no cooperation scheme exhibits a more pronounced increase in task computation rate. This is attributable to its vigorous task processing capacity at the NU, which allows it to capitalize on the higher task arrival rates effectively.

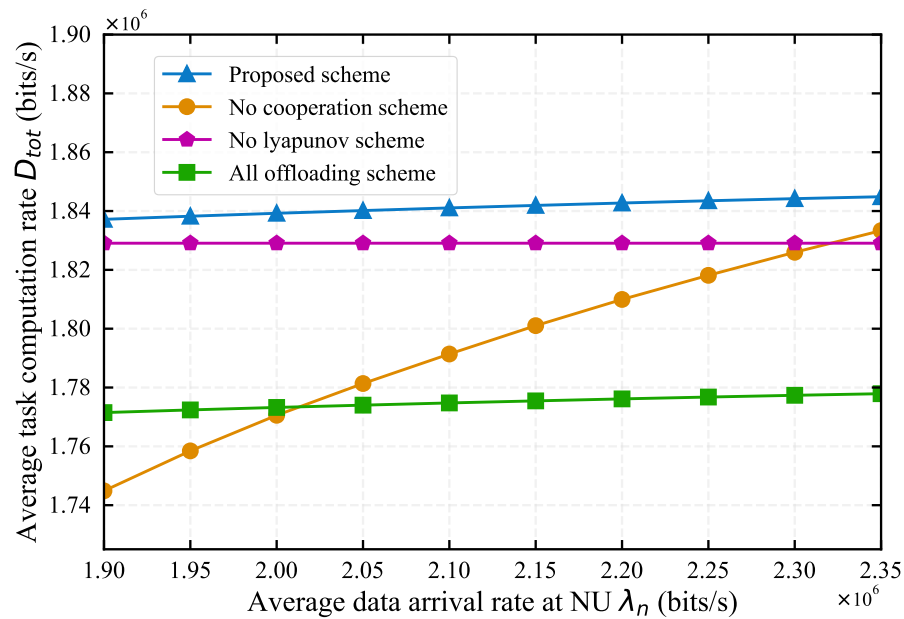


Figure 12. Average computation rates in different schemes with different task arrival rates of FU

5. Conclusions

The joint optimization of computation offloading and resource allocation in WPMEC systems poses a significant challenge due to the time-varying network environments and the time-coupling nature of energy consumption constraint. In this study, we aim at maximizing the average task computation rate in an MEC system with WPT and user collaboration. A task computation maximization problem was formulated considering uncertainty of load dynamics alongside energy consumption constraint. The complexity of this problem is amplified by the interplay of multiple coupled parameters. To surmount these difficulties, we introduce an innovative online algorithm, named UAORA. This algorithm leverages Lyapunov optimization theory to manage long-term task queue lengths and energy consumption, effectively transforming the sequential decision-making dilemma into a series of deterministic optimization problems for each time slot. Extensive simulation results substantiate the effectiveness of the proposed UAORA algorithm, demonstrating a significant enhancement in average task computation performance when compared with benchmark methods. Simulations also underscores the advantages of jointly consider the task computation rate and the task queues in our algorithm. An enticing avenue for future research is to account for dynamically changing local computation rates. Incorporating such variability could potentially augment the system's flexibility and applicability within the complexities of real-world scenarios.

Author Contributions: Methodology, H.H.; Validation, Z.C. and H.F.; Formal analysis, H.H.; Investigation, H.H. and H.F.; Resources, H.H.; Data curation, H.H. and Y.Y.; Writing—original draft, H.H. and H.F.; Writing—review and editing, H.H.; Supervision, Y.Y. 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. Wu, W.; Zhou, F.; Hu, R.Q.; Wang, B. Energy-efficient resource allocation for secure NOMA-enabled mobile edge computing networks. *IEEE Transactions on Communications* **2019**, *68*, 493–505.
2. Na, Z.; Liu, Y.; Shi, J.; Liu, C.; Gao, Z. UAV-supported clustered NOMA for 6G-enabled Internet of Things: Trajectory planning and resource allocation. *IEEE Internet of Things Journal* **2020**, *8*, 15041–15048.
3. Zhao, R.; Zhu, F.; Tang, M.; He, L. Profit maximization in cache-aided intelligent computing networks. *Physical Communication* **2023**, *58*, 102065.
4. Liu, X.; Sun, Q.; Lu, W.; Wu, C.; Ding, H. Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G. *IEEE Wireless Communications* **2020**, *27*, 67–73.
5. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 2322–2358. doi:10.1109/COMST.2017.2745201.
6. Chen, M.; Hao, Y. Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network. *IEEE Journal on Selected Areas in Communications* **2018**, *36*, 587–597. doi:10.1109/JSAC.2018.2815360.
7. Sun, C.; Zhou, J.; Liuliang, J.; Zhang, J.; Zhang, X.; Wang, W. Computation Offloading with Virtual Resources Management in Mobile Edge Networks. 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), 2018, pp. 1–5. doi:10.1109/VTCSpring.2018.8417681.
8. Guo, H.; Zhang, J.; Liu, J.; Zhang, H. Energy-Aware Computation Offloading and Transmit Power Allocation in Ultradense IoT Networks. *IEEE Internet of Things Journal* **2019**, *6*, 4317–4329. doi:10.1109/JIOT.2018.2875535.
9. Sun, H.; Zhou, F.; Hu, R.Q. Joint Offloading and Computation Energy Efficiency Maximization in a Mobile Edge Computing System. *IEEE Transactions on Vehicular Technology* **2019**, *68*, 3052–3056. doi:10.1109/TVT.2019.2893094.
10. Anajemba, J.H.; Yue, T.; Iwendi, C.; Alenezi, M.; Mittal, M. Optimal Cooperative Offloading Scheme for Energy Efficient Multi-Access Edge Computation. *IEEE Access* **2020**, *8*, 53931–53941. doi:10.1109/ACCESS.2020.2980196.
11. Zhu, X.; Luo, Y.; Liu, A.; Bhuiyan, M.Z.A.; Zhang, S. Multiagent Deep Reinforcement Learning for Vehicular Computation Offloading in IoT. *IEEE Internet of Things Journal* **2021**, *8*, 9763–9773. doi:10.1109/JIOT.2020.3040768.
12. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. *Proceedings of the IEEE* **2019**, *107*, 1738–1762. doi:10.1109/JPROC.2019.2918951.
13. 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* **2020**, *19*, 1855–1867. doi:10.1109/TWC.2019.2959300.
14. Mao, Y.; Zhang, J.; Letaief, K.B. Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices. *IEEE Journal on Selected Areas in Communications* **2016**, *34*, 3590–3605. doi:10.1109/JSAC.2016.2611964.
15. Huang, K.; Lau, V.K.N. Enabling Wireless Power Transfer in Cellular Networks: Architecture, Modeling and Deployment. *IEEE Transactions on Wireless Communications* **2014**, *13*, 902–912. doi:10.1109/TWC.2013.122313.130727.
16. 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* **2022**, *16*, 287–298. doi:10.1109/JSYST.2020.3020474.
17. Margolies, R.; Gorlatova, M.; Sarik, J.; Stanje, G.; Zhu, J.; Miller, P.; Szczodrak, M.; Vignraham, B.; Carloni, L.; Kinget, P.; others. Energy-harvesting active networked tags (enhants) prototyping and experimentation. *ACM Transactions on Sensor Networks (TOSN)* **2015**, *11*, 1–27.
18. Tataria, H.; Shafi, M.; Molisch, A.F.; Dohler, M.; Sjöland, H.; Tufvesson, F. 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. *Proceedings of the IEEE* **2021**, *109*, 1166–1199. doi:10.1109/JPROC.2021.3061701.

19. Ju, H.; Zhang, R. Throughput Maximization in Wireless Powered Communication Networks. *IEEE Transactions on Wireless Communications* **2014**, *13*, 418–428. doi:10.1109/TWC.2013.112513.130760.
20. Ji, L.; Guo, S. Energy-Efficient Cooperative Resource Allocation in Wireless Powered Mobile Edge Computing. *IEEE Internet of Things Journal* **2019**, *6*, 4744–4754. doi:10.1109/JIOT.2018.2880812.
21. Li, M.; Zhou, X.; Qiu, T.; Zhao, Q.; Li, K. Multi-Relay Assisted Computation Offloading for Multi-Access Edge Computing Systems With Energy Harvesting. *IEEE Transactions on Vehicular Technology* **2021**, *70*, 10941–10956. doi:10.1109/TVT.2021.3108619.
22. Mach, P.; Becvar, Z. Device-to-Device Relaying: Optimization, Performance Perspectives, and Open Challenges Towards 6G Networks. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 1336–1393. <https://doi.org/10.1109/COMST.2022.3180887>.
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. doi:10.1109/TGCN.2021.3056770.
24. Li, B.; Si, F.; Zhao, W.; Zhang, H. Wireless Powered Mobile Edge Computing With NOMA and User Cooperation. *IEEE Transactions on Vehicular Technology* **2021**, *70*, 1957–1961. doi:10.1109/TVT.2021.3051651.
25. 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* **2021**, *8*, 8005–8020. doi:10.1109/jiot.2020.3041673.
26. Wang, X.; Chen, X.; Wu, W.; An, N.; Wang, L. Cooperative application execution in mobile cloud computing: A stackelberg game approach. *IEEE Communications Letters* **2015**, *20*, 946–949.
27. You, C.; Huang, K. Exploiting Non-Causal CPU-State Information for Energy-Efficient Mobile Cooperative Computing. *IEEE Transactions on Wireless Communications* **2018**, *17*, 4104–4117. doi:10.1109/TWC.2018.2820077.
28. Hu, X.; Wong, K.K.; Yang, K. Wireless Powered Cooperation-Assisted Mobile Edge Computing. *IEEE Transactions on Wireless Communications* **2018**, *17*, 2375–2388. doi:10.1109/twc.2018.2794345.
29. Wang, F.; Xu, J.; Ding, Z. Optimized Multiuser Computation Offloading with Multi-Antenna NOMA. 2017 IEEE Globecom Workshops (GC Wkshps), 2017, pp. 1–7. doi:10.1109/GLOCOMW.2017.8269088.
30. 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.
31. Zhang, S.; Bao, S.; Chi, K.; Yu, K.; Mumtaz, S. DRL-based computation rate maximization for wireless powered multi-AP edge computing. *IEEE Transactions on Communications* **2023**.
32. 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. doi:10.1109/JIOT.2020.3048937.
33. Zheng, X.; Zhu, F.; Xia, J.; Gao, C.; Cui, T.; Lai, S. Intelligent computing for WPT-MEC-aided multi-source data stream. *EURASIP Journal on Advances in Signal Processing* **2023**, *2023*, 52.
34. Zhu, B.; Chi, K.; Liu, J.; Yu, K.; Mumtaz, S. Efficient Offloading for Minimizing Task Computation Delay of NOMA-Based Multiaccess Edge Computing. *IEEE Transactions on Communications* **2022**, *70*, 3186–3203. doi:10.1109/TCOMM.2022.3162263.
35. Wen, Y.; Zhou, X.; Fang, F.; Zhang, H.; Yuan, D. Joint time and power allocation for cooperative NOMA based MEC system. 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall). IEEE, 2020, pp. 1–5.
36. 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.
37. Peng, J.; Qiu, H.; Cai, J.; Xu, W.; Wang, J. D2D-assisted multi-user cooperative partial offloading, transmission scheduling and computation allocating for MEC. *IEEE Transactions on Wireless Communications* **2021**, *20*, 4858–4873.
38. 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.
39. Lin, X.H.; Bi, S.; Su, G.; Zhang, Y.J.A. A Lyapunov-Based Approach to Joint Optimization of Resource Allocation and 3D Trajectory for Solar-Powered UAV MEC Systems. *IEEE Internet of Things Journal* **2024**.
40. Sun, G.; Wang, Z.; Su, H.; Yu, H.; Lei, B.; Guizani, M. Profit Maximization of Independent Task Offloading in MEC-Enabled 5G Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems* **2024**.

41. Shen, S.; Xie, L.; Zhang, Y.; Wu, G.; Zhang, H.; Yu, S. Joint differential game and double deep q-networks for suppressing malware spread in industrial internet of things. *IEEE Transactions on Information Forensics and Security* **2023**.
42. Mei, J.; Dai, L.; Tong, Z.; Zhang, L.; Li, K. Lyapunov optimized energy-efficient dynamic offloading with queue length constraints. *Journal of Systems Architecture* **2023**, *143*, 102979.
43. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Transactions on Communications* **2016**, *64*, 4268–4282.
44. Neely, M. *Stochastic network optimization with application to communication and queueing systems*; Springer Nature, 2022.
45. Boyd, S.; Vandenberghe, L. *Convex optimization*; Cambridge university press, 2004.
46. Grant, M.; Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.1, 2014.
47. 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.
48. 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**.
49. Huang, Y.; Liu, Y.; Chen, F. NOMA-aided mobile edge computing via user cooperation. *IEEE Transactions on Communications* **2020**, *68*, 2221–2235.

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.