sensors

MDPI

Article

Energy-Optimal Latency-Constrained Application Offloading in Mobile-Edge Computing

Xiaohui Gu, Chen Ji * and Guoan Zhang

School of information science and technology, Nantong University, Nantong 226019, China; 17110013@yjs.ntu.edu.cn (X.G.); gzhang@ntu.edu.cn (G.Z.)

* Correspondence: gwidjin@ntu.edu.cn

Received: 21 April 2020; Accepted: 27 May 2020; Published: 28 May 2020

Abstract: Mobile-edge computation offloading (MECO) is a promising emerging technology for battery savings in mobile devices (MD) and/or in latency reduction in the execution of applications by (either total or partial) offloading highly demanding applications from MDs to nearby servers such as base stations. In this paper, we provide an offloading strategy for the joint optimization of the communication and computational resources by considering the blue trade-off between energy consumption and latency. The strategy is formulated as the solution to an optimization problem that minimizes the total energy consumption while satisfying the execution delay limit (or deadline). In the solution, the optimal transmission power and rate and the optimal fraction of the task to be offloaded are analytically derived to meet the optimization objective. We further establish the conditions under which the binary decisions (full-offloading and no offloading) are optimal. We also explore how such system parameters as the latency constraint, task complexity, and local computing power affect the offloading strategy. Finally, the simulation results demonstrate the behavior of the proposed strategy and verify its energy efficiency.

Keywords: mobile-edge computing; mobile application offloading; partial offloading; channel condition; energy-latency trade-off

1. Introduction

For the past several years, mobile devices (MDs), such as smartphones, handheld game consoles, and vehicle multi-media computers, have become virtually ubiquitous and an increasing number of new mobile applications, such as augmented reality (AR), image processing, natural language processing, face recognition, and interactive gaming, have emerged and been the focus of considerable attention [1,2]. These mobile applications are typically latency-sensitive, computation-intensive, and have high energy consumption characteristics. However, under the constraint of physical size, MDs have limited resources, which restricts their battery life and computational capacities [3,4]. Recent studies have shown that mobile-edge computation offloading (MECO) technology provides a promising opportunity for effectively overcoming the hardware limitations and energy consumption problems of MDs, by offloading computing-intensive tasks to adjacent clouds at the edges of mobile networks.

In particular, mobile-edge computing (MEC) offers cloud computing capabilities at the very edge of the mobile networks by deploying MEC servers with sufficient computational resources at base stations (BSs), which thus improves the computational efficiency and reduces the latency, and it has drawn significant attention from both academia and industry [5]. The paradigm shifts from cloud computing to MEC can effectively reduce the backhaul latency and energy consumption, as well as support a more flexible infrastructure in a more cost-effective way. For example, the work in [6] used unmanned aerial vehicles (UAVs) to help device-to-device (D2D) wireless networks [7]. Furthermore, MEC together with

virtual machine (VM) migration can effectively increase the scalability while reducing service delay [8,9]. Owing to these advantages, MEC has attracted extensive research attention from various aspects.

Because computational performance and energy consumption are competing for resources and are critical for mobile users [10], effective computation offloading schemes have become prominent for MEC systems. Basically, a decision on computation offloading may result in binary offloading and partial offloading, which are closely related to the application model/type. It determines whether full or partial offloading is applicable, what could be offloaded, and how [3]. A highly integrated or relatively simple task cannot be partitioned and has to be executed as a whole either locally at the mobile device or offloaded to the MEC server, called binary offloading. In practice, many mobile applications are composed of multiple procedures/components, making it possible to partition the program into two parts with one executed at the mobile device and the other offloaded for edge execution, called partial offloading. Although the former is easier to implement, for a very large dataset, by offloading critical subtasks to the MEC servers, partial offloading can help to reduce the latency and energy consumption on the local devices more flexibly and effectively [3]. However, partial offloading is a very complex process affected by different factors, i.e., whether the applications can be divided into offloadable parts or not; the offloadable part may differ in the amount of data and required computation; how to decide which parts could be offloaded to the MEC; the components of applications that need input from some others; and parallel offloading may not be applicable. Fortunately, these factors have been discussed and studied in many works [11–19].

For the offloading strategy, Bi et al. [20] studied the binary scheme in the wireless powered MEC network that consisted of one server and several users, where the binary policy was adopted for maximizing the weighted sum computational rate. Zhang et al., in [21], used the auction theory to model the matching relationship between the MEC server and the MDs, so as to offload tasks to the optimal MEC server. Al-Shuwaili et al., in [22], jointly allocated communication and computational resources to minimize the total MD energy consumption under latency constraints by successive convex approximation. In [23], by considering the data arrival time instants and computational deadlines, You et al. proposed an energy-efficient resource management policy for MECO systems and formulated an optimization strategy that minimized the total mobile energy consumption. For partial offloading, the work in [24] investigated MEC systems with one energy harvesting (EH) device and proposed an effective dynamic computation offloading algorithm to minimize the execution cost. In [25], You et al. studied the energy saving partial computation offloading problem, for a multiuser MECO system based on time-division multiple access (TDMA) and orthogonal frequency-division multiple access (OFDMA). In [26], Wang et al. also considered the energy consumption minimization problem in a scenario with one MEC server and multiple users. They also employed wireless power transfer to further alleviate the burden on battery usage. Furthermore, the work in [27] implemented a cooperative communication system that had three nodes, in which one of them acted as the helper for relaying and computing. Lastly, the work in [28] considered both the fronthaul and backhaul link transmission and offloading in a small-cell architecture.

However, some of above literature only focused on the performance indicator of latency, while this paper aims at minimizing the energy consumption, which is critical for power-limited devices, especially for a device with lower battery energy. Furthermore, in the problem formulation, the previously mentioned works [25,26] only considered the local computing time and transmission time, but the server computing time was neglected; thus, they fell short of accuracy for scenarios in which the MDs needed to offload computation-intensive jobs to MEC servers with limited computing capacity. Moreover, most of the works performed so far performed the management aspects, the experimental evaluation of energy savings associated with offloading, and/or the definition of an offloading criterion that considered the energy cost of the radio interface (e.g., LTE or WiFi), but without optimizing the energy cost of the data transfer according to the current channel conditions. Notice, however, that depending not only on the application, but also on the current channel conditions, the best strategy concerning the offloading process may be different. Considering minimizing the energy consumption

at the MD for delay-constrained offloading jobs, an optimization problem can be formulated to allocate the computation and communication resources jointly in the described MECO system.

In this paper, we pursue an energy-efficient MEC offloading design considering the partial and binary strategies, aiming to minimize the energy consumption of the mobile application while ensuring that the computational task can be successfully finished with a predefined hard limit of execution time. We derive the analytical solution for the single-MD to the MEC case, which can be generalized to the multi-MD case in later research. Our work presents some main differences from previous works. First, we introduce an offloading model that includes energy consumption at MD and processing time at the server side, while these offloading parameters were neglected in previous works such as [24,27]. Second, instead of considering that the application is run either totally in the cloud or totally in the MD, we include an optimization variable as the quantity of data to be processed on each side to allow a partial trade-off between local computing and offloading. Finally, in contrast with previous works, our approach allows for adapting the transmission strategy to the current channel state as perceived by the MD in the uplink.

The main technical contributions of this paper are as follows.

1. Based on the offloading model above, we formulate an offloading-optimization problem that minimizes the MD energy usage while ensuring the task is completed within a prescribed deadline, by jointly optimizing the transmitting time and offloading ratio.
2. We transform the latency-constrained problem into a two-stage optimization problem, which can be analytically solved using standard convex optimization techniques. In the solution, a channel condition threshold is derived above which full offloading is the optimal decision, whereas below the threshold, partial offloading is performed to trade off between the time and energy cost of offloading. For the partial offloading policy, the optimal transmission time and offloading ratio are derived in closed form expression.
3. This paper also discusses in detail various practical aspects of the offloading strategy, including the conditions under which total offloading or non offloading is optimal, the minimum admissible latency constraint that renders the problem feasible, and how the system parameters affect the offloading decision, including the energy efficiency of MD, the task complexity, and the computing capacity of local devices.

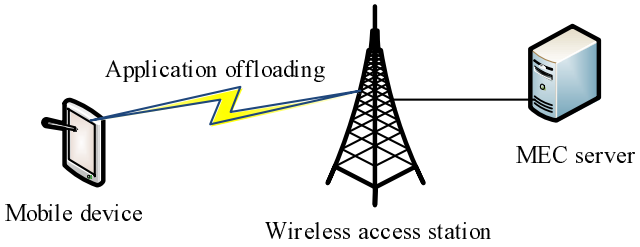
The remainder of this paper is organized as follows. In Section 2, the communication and computational model in mobile execution and MEC execution are presented, respectively. In Section 3, the optimization problem is formulated and solved for optimal partial offloading, while binary offloading decisions are also discussed under different channel conditions. The effects on the offloading strategy of a number of system parameters are analyzed in detail in Section 4. Finally, some simulation results and conclusions are provided in Sections 5 and 6, respectively.

2. System Model

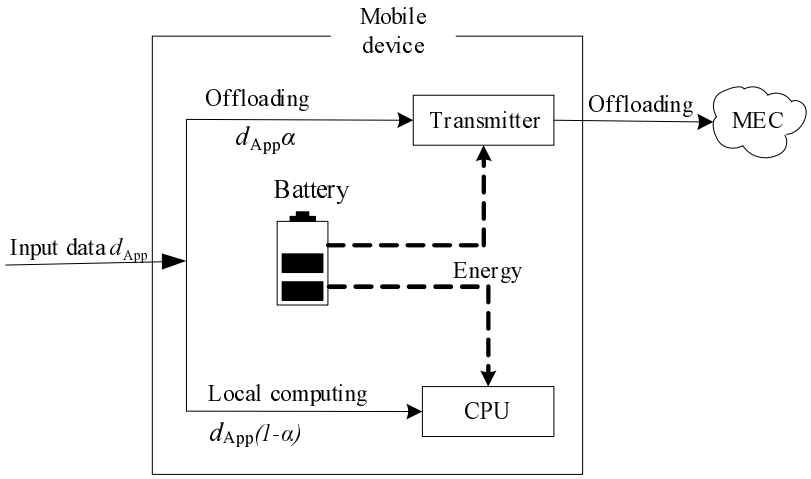
As shown in Figure 1a, the MEC server is a computing device installed at a wireless access station. The MD is provided with wireless access to the computational resources located in proximal servers. By assigning the computing-intensive applications to the BS, it can help mobile users improve the computing performance. The pioneering literature has extensively studied mobile networking and mobile-edge computing models (e.g., [23,29–31]), which provide useful insights and make performance analysis tractable. Based on these results, we adopted a canonical model that captured the essentials of a typical mobile application. Although it is possible to build a system model to depict the various aspects of MEC in detail, such a model could be too complex to be treated analytically and is difficult for practical systems. Specifically, in this paper, the mobile application is abstracted into a profile with three parameters [22,23], including:

- Input data size d_{App}: the number of data bits as the input to the application;
- Required CPU cycles c_{App}: the number of CPU cycles required to complete the application;

- Application completion deadline T_{Max}: the maximum latency, before which the application should be completed.



(a)



(b)

Figure 1. System model. (**a**) Mobile-edge computing (MEC) platform. (**b**) The offloading workflow.

For the partial offloading policy, let α represent the ratio of the offloaded data to the full data size, and the offloaded data size of the mobile application can be denoted as $l_{\text{Off}} = d_{\text{App}}\alpha$. We consider the application scenario to be quasi-static, where the environment at MD will not change during the computation offloading period.

Notice that the input data size d_{App}, the required CPU cycles c_{App}, and the application completion deadline T_{Max} may have an impact on the energy consumption of mobile applications. With more data input, more CPU cycles, and/or a more stringent completion deadline, the energy consumption will be higher. Therefore, it will be beneficial to offload computation-intensive tasks to the MEC servers.

2.1. Communication Model

We first introduce the communication model between the MD and the MEC platform. The computation offloading policy was carried out based on energy consumption and completion time. The transmission power for the MD is denoted as p_{Tr}, and g_s represents the channel gain of the BS. The uplink data rate r_{Tr} for computation offloading of the MD is given by:

$$r_{\text{Tr}} = B \log_2 \left(1 + \frac{p_{\text{Tr}} g_s^2}{(N_0 + I)B}\right), \tag{1}$$

where N_0 and I represent the power spectrum density of additive white Gaussian noise and the interference signal, respectively. Letting B be the bandwidth of the channel, at the MEC, the received signal power can be denoted by a function of data rate r:

$$h(r) = (N_0 + I)B\left(2^{\frac{r}{B}} - 1\right), \tag{2}$$

which is monotonically increasing and convex for $r > 0$. The offloading transmission rate can be denoted as:

$$r_{\mathrm{Tr}} = \frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}, \tag{3}$$

where t_{Tr} is the transmission time of the MD for offloading the input data size d_{App}. Then, the transmission power p_{Tr} can be calculated by combining (2) and (3):

$$p_{\mathrm{Tr}} = \frac{1}{g_s^2} h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right). \tag{4}$$

2.2. Computation Model

(1) Local computing: With the local computing approach, the MD executes the application locally on the MD. Let h_{MD} denote the computational capability (i.e., CPU cycles per second) of the MD, and accordingly, the completion time for local computing is defined as:

$$t_{\mathrm{Lo}} = \frac{c_{\mathrm{App}}}{h_{\mathrm{MD}}}, \tag{5}$$

For the local computation energy, we have:

$$e_{\mathrm{Lo}} = f_{\mathrm{MD}}\, c_{\mathrm{App}}, \tag{6}$$

where f_{MD} is the consumed energy per CPU cycle for the MD. Besides, f_{MD} indicates the energy efficiency of the MD.

(2) Edge computing: With the edge computing approach, the MD is allowed to offload its computational load to a nearby MEC server. Subsequently, the server performs the offloaded task and feeds the result back to the MD. Therefore, in the overall offloading, the following three consecutive phases are included: (i) transmitting phase, (ii) computing phase, and (iii) receiving phase. According to the communication model, the transmission time and energy consumption of the MD sending its computational load to the MEC server are calculated by:

$$t_{\mathrm{Tr}} = \frac{d_{\mathrm{App}}}{r_{\mathrm{Tr}}}, \tag{7}$$

and:

$$e_{\mathrm{Tr}} = p_{\mathrm{Tr}}\, t_{\mathrm{Tr}}. \tag{8}$$

The execution time for the computing load d_{App} on the MEC server is computed by:

$$t_{\mathrm{Ser}} = \frac{c_{\mathrm{App}}}{h_{\mathrm{MEC}}}, \tag{9}$$

where h_{MEC} denotes the computational capacity of the MEC server.

For the edge computing approach, the completion time and energy consumption of the mobile application are formulated as:

$$t_{\mathrm{Off}} = t_{\mathrm{Tr}} + t_{\mathrm{Ser}} + t_{\mathrm{Rec}}, \tag{10}$$

and:

$$e_{\mathrm{Off}} = e_{\mathrm{Tr}} + e_{\mathrm{Rec}}, \tag{11}$$

where t_{Rec} and e_{Rec} denote, respectively, the time and energy required at the MD side when receiving the computational outcome from the MEC server. As in the existing works [32,33], the receiving

time t_{Rec} and the receiving energy e_{Rec} can be ignored because for many applications, such as face recognition and car barrier detection, the data size of the result is relatively much smaller than the size of the input. In this paper, we consider the energy consumption on the MD side, while our future work will consider the execution energy consumption of the MEC server.

Generally, we can assume that the server has higher computational capacity than the mobile device, i.e., $h_{\text{MEC}} > h_{\text{MD}}$. Therefore, if a job is fully offloaded to the MEC server, the time saved in the computation phase can be expressed as:

$$T_{\text{Sav}} = c_{\text{App}} \left(\frac{1}{h_{\text{MD}}} - \frac{1}{h_{\text{MEC}}} \right). \tag{12}$$

2.3. Partial Offloading Model

Next, we introduce the partial offloading model. As in [22,33], we apply a linear model in this paper and the MD enables programs to process computing tasks sequentially. Here, α is the offloading ratio, which is the proportion of the task data offloaded to the MEC server, as shown in Figure 1b. The offloading data rate is $r_{\text{Tr}} = \frac{d_{\text{App}}}{t_{\text{Tr}}}$, the offloading transmit power p_{Tr}, and the partial offloading time $t_{\text{Off}}\alpha$. Therefore, the total energy consumption of the MD includes both local computing consumption and partial offloading consumption, which is expressed as:

$$e_{\text{Tot}} = e_{\text{Off}}\alpha + e_{\text{Lo}}(1-\alpha), \tag{13}$$

$$= p_{\text{Tr}} t_{\text{Tr}} \alpha + f_{\text{MD}} c_{\text{App}} (1-\alpha), \tag{14}$$

Because we consider the computational performance as well, the completion time of the MD can be denoted as:

$$t_{\text{Tot}} = t_{\text{Off}}\alpha + t_{\text{Lo}}(1-\alpha). \tag{15}$$

The completion time above also includes the partial offloading time and the local computing time, which are proportional to the original full offloading time and local-computing time, respectively.

By the definitions of e_{Tot} and t_{Tot}, the overall energy consumption of the MD is e_{Tot}, for the task completed duration t_{Tot}.

3. Joint Optimization of the Communication and Computation Resources with Partial Offloading

Based on the offloading model established in the previous section, we are now ready to optimize an offloading policy. Our goal was to find a policy that satisfies the hard delay constraints with minimal energy consumption. The steps were carried out by analytical means, and solutions were formulated under different channel conditions and computing profiles.

3.1. Optimization Formulation

Now, we present the optimization problem of minimizing the energy consumption at the MD while satisfying the hard delay constraint. As a first step, the total energy is a monotonically decreasing function of transmit time, as shown in Lemma 1:

Lemma 1. *For any fixed* $0 \leq \alpha \leq 1$, *the total energy* e_{Tot} *is monotonically decreasing with* t_{Tr}.

Proof. Please see Appendix A. □

Lemma 1 proves that energy can be saved by extending the full offloading time under a fixed offloading ratio, till reaching the time constraint.

The optimization objective is to find the offloading policy to minimize the total energy consumption for the MD. It can be noted that e_{Tot} in (13) and t_{Tot} in (15) are affine functions of α. To simplify the

problem, we let α and t_{Tr} be the optimization variables, and the optimal p^*_{Tr} can be calculated from t^*_{Tr} by Equation (4). Hence, the energy consumption minimization problem for the MD is formulated as:

$$\begin{aligned} &\min_{\{t_{\mathrm{Tr}},\alpha\}} \; p_{\mathrm{Tr}}\, t_{\mathrm{Tr}}\, \alpha + f_{\mathrm{MD}} c_{\mathrm{App}} (1-\alpha) \\ &s.t. \; \left(t_{\mathrm{Tr}} + \frac{c_{\mathrm{App}}}{h_{\mathrm{MEC}}} \right) \alpha + \frac{c_{\mathrm{App}}}{h_{\mathrm{MD}}} (1-\alpha) \leq T_{\mathrm{Max}}. \end{aligned} \tag{16}$$

The constraint in (16) specifies that the total completion time of the application bits is bounded by the maximum affordable latency T_{Max}.

It is shown from (16) that the primal optimization problem is affected by multiple parameters from both the communication and computational aspects. Of the parameters, the channel gain g_s is crucial because it determines the quality of the wireless channel and, hence, the cost of application offloading in terms of energy.

3.2. Offloading Policy in Good Channel Conditions

In good channel conditions, a high-data-rate link can be established and maintained at a low energy cost. In addition, the MEC servers have greater computing capacity compared to the local MDs. Therefore, it is of interest to determine analytically when to perform full offloading to take advantage of high-performance servers and good channel conditions. We have the following assertion:

Assertion 1. *There exists channel gain threshold g_{Th} that, for a channel gain better than threshold g_{Th}, full offloading with $\alpha^* = 1$ is energy-optimal under any latency constraint, and:*

$$g_{\mathrm{Th}} = \sqrt{\frac{(N_0 + I) B T_{\mathrm{Sav}} \left(2^{\frac{d_{\mathrm{App}}}{B T_{\mathrm{Sav}}}} - 1 \right)}{f_{\mathrm{MD}} c_{\mathrm{App}}}}. \tag{17}$$

Proof. Please see Appendix B. □

To satisfy the hard latency constraint at minimum energy cost, one can substitute $\alpha^* = 1$ into (15) and apply Lemma 1. The optimal transmission time t^*_{Tr} for $g_s > g_{\mathrm{Th}}$ is given by:

$$t^*_{\mathrm{Tr}} = T_{\mathrm{Max}} - \frac{c_{\mathrm{App}}}{h_{\mathrm{MEC}}}, \tag{18}$$

and p^*_{Tr} can be computed using (4).

3.3. Offloading Policy for the Channel Condition below the Threshold

On the other hand, let us consider the case $g_s \leq g_{\mathrm{Th}}$. In that situation, it takes more energy to overcome the propagation loss to offload a task to the MEC server ($e_{\mathrm{Off}} \geq e_{\mathrm{Lo}}$). The objective function of (16) has the following features:

$$\begin{aligned} e_{\mathrm{Tot}} &= e_{\mathrm{Off}} \alpha + e_{\mathrm{Lo}} (1-\alpha) && (19) \\ &= p_{\mathrm{Tr}} t_{\mathrm{Tr}} \alpha + f_{\mathrm{MD}} c_{\mathrm{App}} (1-\alpha) && (20) \\ &= \frac{(N_0 + I)\, B\, t_{\mathrm{Tr}}}{g_s^2} \left(2^{\frac{d_{\mathrm{App}}}{B t_{\mathrm{Tr}}}} - 1 \right) \alpha \\ &\quad + f_{\mathrm{MD}} c_{\mathrm{App}} (1-\alpha), && (21) \end{aligned}$$

and:

$$\frac{\partial^2 e_{\mathrm{Tot}}}{\partial t_{\mathrm{Tr}}^2} = \frac{\alpha(N_0+I)2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}} d_{\mathrm{App}}^2 r_{\mathrm{Tr}}^2 (\ln 2)^2}{B g_s^2 t_{\mathrm{Tr}}^3} > 0. \tag{22}$$

It can be observed that Problem (16) is non-convex and hence is challenging to solve directly.

Assertion 2. *Problem (16) can be transformed into a two stage optimization problem, by iteratively finding the optimal α^* and t_{Tr}^*.*

Proof. Since $\min\limits_{\{t_{\mathrm{Tr}},\alpha\}} \; p_{\mathrm{Tr}}\, t_{\mathrm{Tr}}\, \alpha + f_{\mathrm{MD}} c_{\mathrm{App}}(1-\alpha)$ is equivalent to $\min\limits_{t_{\mathrm{Tr}}} \min\limits_{\alpha} \; p_{\mathrm{Tr}}\, t_{\mathrm{Tr}}\, \alpha + f_{\mathrm{MD}} c_{\mathrm{App}}(1-\alpha)$, one can first solve α for any specific value of t_{Tr} and, next, substitute the optimal α^* into (16) to find the optimal t_{Tr}^*. Therefore, in the first sub-problem (23), we optimize the offloading ratio α for a given t_{Tr}. In the second sub-problem (29), we aim to find the optimal t_{Tr}^*, by substituting α^* into the original problem. □

The first sub-problem is given by:

$$\begin{aligned} &\min_{\alpha} \; p_{\mathrm{Tr}}\, t_{\mathrm{Tr}}\, \alpha + f_{\mathrm{MD}} c_{\mathrm{App}}(1-\alpha) \\ &s.t. \; \left(t_{\mathrm{Tr}} + \frac{c_{\mathrm{App}}}{h_{\mathrm{MEC}}}\right)\alpha + \frac{c_{\mathrm{App}}}{h_{\mathrm{MD}}}(1-\alpha) \le T_{\mathrm{Max}}. \end{aligned} \tag{23}$$

Therefore, we define function $h(\alpha)$:

$$h(\alpha) \triangleq e_{\mathrm{Tot}}. \tag{24}$$

It is obvious that $h(\alpha)$ is convex and non-decreasing monotonically with respect to α, since $e_{\mathrm{Off}} \ge e_{\mathrm{Lo}}$ under the condition where $g_s \le g_{\mathrm{Th}}$. Furthermore, from the latency constraint, we can find $\alpha \ge \frac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t_{\mathrm{Tr}}}$, under the default condition $t_{\mathrm{Lo}} \ge T_{\mathrm{Max}}$ and $T_{\mathrm{Sav}} \ge t_{\mathrm{Tr}}$ (When $g_s \le g_{\mathrm{Th}}$, it costs more energy offloading data to remote servers than local computing. Therefore, if $t_{\mathrm{Lo}} < T_{\mathrm{Max}}$, the local computing is optimal; otherwise, partial offloading is energy-optimal.).

Hence, for a given t_{Tr}, the optimal α^* is given by:

$$\alpha^* = \frac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t_{\mathrm{Tr}}}. \tag{25}$$

Next, we substitute (25) into the main problem and solve t_{Tr}. We define:

$$g(t_{\mathrm{Tr}}) \triangleq \alpha^* = \tfrac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t_{\mathrm{Tr}}}, \tag{26}$$

and its second-order derivative is:

$$\frac{\partial^2 g(t_{\mathrm{Tr}})}{\partial t_{\mathrm{Tr}}^2} = \frac{2(t_{\mathrm{Lo}} - T_{\mathrm{Max}})}{(T_{\mathrm{Sav}} - t_{\mathrm{Tr}})^3} \ge 0 \tag{27}$$

Therefore, $g(t_{\mathrm{Tr}})$ is convex with respect to t_{Tr}. From all the above, the composite function $f(t_{\mathrm{Tr}}) = h(g(t_{\mathrm{Tr}}))$ is convex [34] (chapter 3). By substituting (25) into (16),

$$f(t_{\mathrm{Tr}}) = \frac{(T_{\mathrm{Max}} - t_{\mathrm{Lo}})t_{\mathrm{Tr}}}{g_s^2(t_{\mathrm{Tr}} - T_{\mathrm{Sav}})} h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right) + f_{\mathrm{MD}} c_{\mathrm{App}} \left(1 - \frac{T_{\mathrm{Max}} - t\mathrm{Lo}}{t_{\mathrm{Tr}} - T_{\mathrm{Sav}}}\right). \tag{28}$$

we can now solve the convex sub-problem that minimizes $f(t_{\mathrm{Tr}})$, thanks to the fact that $f(t_{\mathrm{Tr}})$ is convex. The optimal t_{Tr}^* is to be found by solving the second sub-problem:

$$\min_{t_{\mathrm{Tr}} > 0} \; f(t_{\mathrm{Tr}}) \tag{29}$$

Solution to the Sub-Problem

Next we solve the convex problem (29) using standard convex optimization techniques, as shown in the following.

Assertion 3. *The optimal t_{Tr}^{*} is computed by the expression below:*

$$t_{\mathrm{Tr}}^{*}=\frac{d_{\mathrm{App}}}{\frac{B}{\ln 2}W_0[(-\frac{g_s^2 f_{\mathrm{MD}} c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}}-\frac{1}{e})2^{-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}}]+1+\frac{d_{\mathrm{App}}}{T_{\mathrm{Sav}}}}. \tag{30}$$

where $W_0(\cdot)$ is the Lambert Wfunction [35].

Proof. First, recall that T_{Sav} in (12) is the difference between the time that the task is executed locally and remotely. By letting the first-order derivative of the objective function of (29) be zero, we have:

$$\frac{T_{\mathrm{Max}}-t_{\mathrm{Lo}}}{g_s^2}\left\{\left[\frac{1}{t_{\mathrm{Tr}}-T_{\mathrm{Sav}}}+\frac{t_{\mathrm{Tr}}}{(t_{\mathrm{Tr}}-T_{\mathrm{Sav}})^2}\right]h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)+\frac{t_{\mathrm{Tr}}}{t_{\mathrm{Tr}}-T_{\mathrm{Sav}}}h'\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)\right\}$$
$$+f_{\mathrm{MD}}c_{\mathrm{App}}\frac{T_{\mathrm{Max}}-t_{\mathrm{Lo}}}{(t_{\mathrm{Tr}}-T_{\mathrm{Sav}})^2}=0. \tag{31}$$

It can be simplified as:

$$T_{\mathrm{Sav}}h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)-t_{\mathrm{Tr}}(t_{\mathrm{Tr}}-T_{\mathrm{Sav}})h'\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)-g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}=0, \tag{32}$$

$$t_{\mathrm{Tr}}h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)-g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}=(t_{\mathrm{Tr}}-T_{\mathrm{Sav}})\left[h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)+t_{\mathrm{Tr}}h'\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)\right], \tag{33}$$

$$(N_0+I)Bt_{\mathrm{Tr}}(2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}}-1)-g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}$$
$$=(t_{\mathrm{Tr}}-T_{\mathrm{Sav}})\left[(N_0+I)B\left(2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}}-1\right)-\frac{(N_0+I)d_{\mathrm{App}}\ln 2\,2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}}}{t_{\mathrm{Tr}}}\right]. \tag{34}$$

By denoting $u\triangleq\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}$ and based on Equation (34), u satisfies:

$$(N_0+I)\left[d_{\mathrm{App}}\ln 2\,2^u+BT_{\mathrm{Sav}}2^u-BT_{\mathrm{Sav}}\ln 2\,u\,2^u\right]=g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}+(N_0+I)BT_{\mathrm{Sav}}, \tag{35}$$

$$2^u\ln 2\left(\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}-\frac{1}{\ln 2}-u\right)=\frac{g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}}+1, \tag{36}$$

$$2^{\left(u-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}-\frac{1}{\ln 2}\right)}\ln 2\left(u-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}-\frac{1}{\ln 2}\right)=\left(-\frac{g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}e}-\frac{1}{e}\right)2^{-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}}, \tag{37}$$

We can finally further derive that:

$$e^{\ln 2\left(u-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}-\frac{1}{\ln 2}\right)}\ln 2\left(u-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}-\frac{1}{\ln 2}\right)=\left(-\frac{g_s^2 f_{\mathrm{MD}}c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}e}-\frac{1}{e}\right)2^{-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}}. \tag{38}$$

However, recall that the condition $g_s > g_{\mathrm{Th}}$ has already been discussed in Section 3.2, resulting in full offloading.

For $g_s \leq g_{\mathrm{Th}}$, by referring to the definition of the Lambert W function [35], u in Equation (38) is obtained as follows:

$$u = \frac{1}{\ln 2}\left\{W_0\left[\left(-\frac{g_s^2 f_{\mathrm{MD}} c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}e} - \frac{1}{e}\right)2^{-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}}\right] + 1\right\} + \frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}},$$

Therefore, the optimal time t_{Tr}^* readily follows. □

The optimal transmitting power p_{Tr}^* can be derived from Equation (4) accordingly. Finally, α^* in (25) can be redescribed as:

$$\alpha^* = \frac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t_{\mathrm{Tr}}^*}. \tag{39}$$

Overall, the solution in (30) and (39) indicates that if the channel is not good enough, offloading the computing tasks to the remote MEC will have a high energy cost; thus, the partial offloading strategy is the preferred decision.

Combining the above subsections, the optimal solution of the primal problem (16) is finally presented as (40). For channel conditions better than the threshold g_{Th} defined in (17), full offloading is carried out; for channel $g_s < g_{\mathrm{Th}}$, partial offloading is performed, and the offloading ratio α^* and transmission time t_{Tr}^* are optimally chosen to minimize the total energy cost while satisfying the hard time constraint T_{Max}. Finally, the transmission power p_{Tr}^* can be derived from t_{Tr}^* by (4).

$$\begin{cases} t_{\mathrm{Tr}}^* = T_{\mathrm{Max}} - \frac{c_{\mathrm{App}}}{h_{\mathrm{MEC}}}, & \alpha^* = 1, \quad \text{if } g_s > g_{\mathrm{Th}} \\ t_{\mathrm{Tr}}^* = \dfrac{d_{\mathrm{App}}}{\frac{B}{\ln 2} W_0[(-\frac{g_s^2 f_{\mathrm{MD}} c_{\mathrm{App}}}{(N_0+I)BT_{\mathrm{Sav}}} - \frac{1}{e})2^{-\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}}] + 1 + \frac{d_{\mathrm{App}}}{T_{\mathrm{Sav}}}}, & \alpha^* = \frac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t_{\mathrm{Tr}}^*}, \quad \text{if } g_s \leq g_{\mathrm{Th}} \end{cases} \tag{40}$$

3.4. Special Cases of Full Offloading and Non Offloading

Here, we proceed to analyze the special cases of the optimization problem, to determine under what conditions the binary decisions of full offloading or non offloading can be performed.

3.4.1. Optimality Condition of Total Offloading

Here, we investigate the conditions under which the optimum is to process all the application bits at the MEC server, i.e., $\alpha^* = 1$. The condition is twofold: (1) $\alpha^* = 1$ should be feasible, and (2) $\frac{\partial e_{\mathrm{Tot}^*}}{\partial \alpha^*} \leq 0, \forall\, 0 \leq \alpha^* \leq 1$.

According to the completion time constraint, the first condition (1) holds when $T_{\mathrm{Max}} \geq t_{\mathrm{Off}}^*$, and $t_{\mathrm{Off}}^* = t_{\mathrm{Tr}}^* + t_{\mathrm{Ser}}$, i.e., the time required to transmit the task to the MEC server plus the time for remote processing should not be larger than the maximum latency constraint.

Finally, the sufficient condition (2) can be expanded as:

$$\frac{\partial e_{\mathrm{Tot}^*}}{\partial \alpha^*} = \frac{t_{\mathrm{Tr}}^*}{g_s^2} h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}^*}\right) - f_{\mathrm{MD}} c_{\mathrm{App}} \leq 0 \Rightarrow e_{\mathrm{Off}}^* \leq e_{\mathrm{Lo}}. \tag{41}$$

This condition holds when $g_s \geq g_{\mathrm{Th}}$ and $t_{\mathrm{Tr}}^* \geq T_{\mathrm{Sav}}$, which is equivalent to the situation where it will take less energy to offload the application, as discussed in Section 3.2.

3.4.2. Optimality of Non Offloading

Here, we provide the necessary conditions under which the optimal solution is to process all task bits locally at the MD, i.e., $\alpha^* = 0$. These conditions are twofold: (1) $\alpha^* = 0$ should be feasible, and (2) $\frac{\partial e_{\mathrm{Tot}}^*}{\partial \alpha^*} \mid_{\alpha^*=0} \geq 0$.

According to the completion time constraint, the first condition (1) holds only if $T_{\text{Max}} \geq t_{\text{Lo}}$, i.e., the time for executing all the application bits locally at the MD should not violate the latency constraint.

On the other hand, we see from Equation (19) that $e^*_{\text{Tot}}(\alpha^*)$ is equal to $f_{\text{MD}} c_{\text{App}}$ at $\alpha^* = 0$ and is continuous within a certain interval containing $\alpha^* = 0$. These two characteristics allow us to state that $e^*_{\text{Tot}}(\alpha^*)$ is constant (i.e., not dependent on α^*) within a certain interval containing $\alpha^* = 0$; therefore, the second condition (2) holds only when:

$$\frac{\partial e^*_{\text{Tot}}}{\partial \alpha^*} = \frac{t^*_{\text{Tr}}}{g_s^2} h \left(\frac{d_{\text{App}}}{t^*_{\text{Tr}}} \right) - f_{\text{MD}} c_{\text{App}} \geq 0 \Rightarrow e^*_{\text{Off}} \geq e_{\text{Lo}}. \tag{42}$$

This condition indicates that if offloading within a time constraint costs more energy than local computing, the MD should rely on its own resources. This condition holds in the situation where $g_s \leq g_{\text{Th}}$, and the proof is left for Appendix B.

To summarize the results of the above subsections, the offloading policy decisions are listed in Table 1, under different channel conditions and latency constraints.

Table 1. Offloading policy decisions.

Channel \ Latency	$t_{\text{Max}} < t_{\text{Lo}}$	$t_{\text{Max}} \geq t_{\text{Lo}}$
$g_s > g_{\text{Th}}$	Full offloading	Full offloading
$g_s \leq g_{\text{Th}}$	Partial offloading	Non offloading

4. Analysis of the System Parameters

In this section, we investigate how the system parameters affect the optimization problem, including the time constraint, task complexity, and processing capacity at MD. This not only enables us to understand how the offloading strategy operates under different system settings, but also helps the system-level investigations of practical MEC systems in further research.

4.1. The Latency Constraint T_{Max} and Feasibility of α^*

The completion time constraint of (16) is $t_{\text{Off}}\alpha + t_{\text{Lo}}(1-\alpha) \leq T_{\text{Max}}$, which is related to the available time budget (and formulated in terms of maximum allowed latency), and can be rewritten as a set of two detailed constraints as follows:

$$t_{\text{Off}}\alpha \leq T_{\text{Max}} \Rightarrow \alpha \leq \frac{T_{\text{Max}}}{t_{\text{Off}}}, \tag{43a}$$

$$t_{\text{Lo}}(1-\alpha) \leq T_{\text{Max}} \Rightarrow \alpha \geq 1 - \frac{T_{\text{Max}}}{t_{\text{Lo}}}. \tag{43b}$$

Therefore, the minimum and maximum values of variable α can be defined as follows:

$$\alpha \geq \alpha_{\text{Min}}, \alpha_{\text{Min}} = \max\left\{0, 1 - \frac{T_{\text{Max}}}{t_{\text{Lo}}}\right\}, \tag{44a}$$

$$\alpha \leq \alpha_{\text{Max}}, \alpha_{\text{Max}} = \min\left\{1, \frac{T_{\text{Max}}}{t_{\text{Off}}}\right\}. \tag{44b}$$

The primal problem (16) is feasible only if $\alpha_{\text{Max}} \geq \alpha_{\text{Min}}$. As shown in Equation (44a) and Equation (44b), α_{Max} and α_{Min} can be viewed as linear functions of T_{Max}, and they are upper and lower bounded by one and zero, respectively. Next, we investigate how T_{Max} impacts the offloading policy in different channel conditions.

4.1.1. Channel Gain $g_s > g_{\mathrm{Th}}$

In Figure 2, it can be clearly seen that it obtains the lowest value of T_{Max}, under which a full offloading strategy will save energy. Let us denote such a lowest value by T_1. According to the closed form expressions of $t^*_{\mathrm{Tr}}, e_{\mathrm{Off}}$ and e_{Lo}, T_1 can be calculated as follows:

$$\begin{cases} t^*_{\mathrm{Tr}} = T_{\mathrm{Max}} - t_{\mathrm{Ser}} \\ \frac{t^*_{\mathrm{Tr}}}{g_s^2} h\left(\frac{d_{\mathrm{App}}}{t^*_{\mathrm{Tr}}}\right) = f_{\mathrm{MD}} c_{\mathrm{App}} \end{cases} \Rightarrow T_{\mathrm{Max}} = T_1 = 2^{\frac{d_{\mathrm{App}}}{B}} - \frac{g_s^2 f_{\mathrm{MD}} c_{\mathrm{App}}}{(N_0 + I)B} + t_{\mathrm{Ser}}. \tag{45}$$

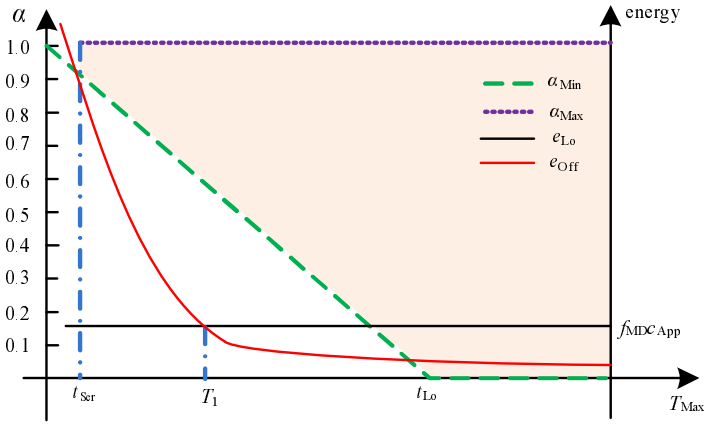


Figure 2. Dependence of α_{Max}, α_{Min}, and e^*_{Tot} versus T_{Max} for $g_s > g_{\mathrm{Th}}$.

As seen in Figure 2, in the situation where $t_{\mathrm{Ser}} \leq T_{\mathrm{Max}} \leq T_1$, the full offloading strategy has to be carried out to meet the tight latency constraint at the expense of high energy consumption. If $T_1 \leq T_{\mathrm{Max}} \leq t_{\mathrm{Lo}}$, the full offloading can save both time and energy. If $t_{\mathrm{Lo}} \leq T_{\mathrm{Max}}$, offloading will take advantage of the relaxed time constraint, extending the transmission time to save more energy.

4.1.2. Channel Gain $g_s < g_{\mathrm{Th}}$

In Figure 3, it can be observed that, for any t_{Off} and t_{Lo}, there exists the lowest value of T_{Max} (also called the minimum admissible latency) under which α^* in Equation (16) becomes unfeasible. Let us denote the lowest value by T_0. By referring to (44a) and (44b), T_0 has to satisfy that:

$$1 - \frac{T_0}{t_{\mathrm{Lo}}} = \frac{T_0}{t_{\mathrm{Off}}} \Rightarrow T_0 = \frac{t_{\mathrm{Off}} t_{\mathrm{Lo}}}{t_{\mathrm{Off}} + t_{\mathrm{Lo}}}. \tag{46}$$

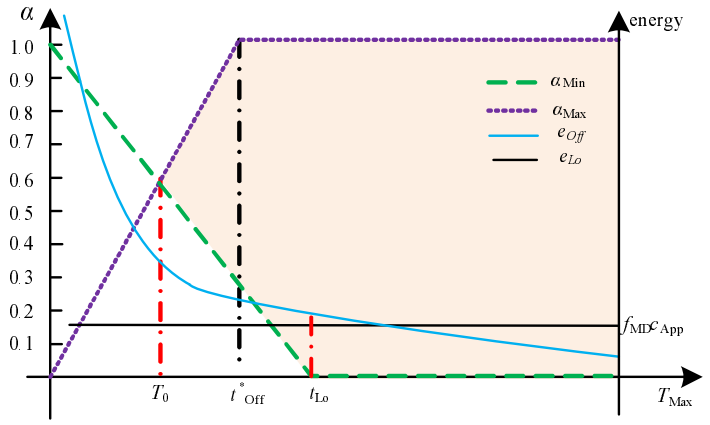


Figure 3. Dependence of α_{Max}, α_{Min}, and e^*_{Tot} versus T_{Max} for $g_s \leq g_{\mathrm{Th}}$.

In fact, as shown in Figure 3, T_0 is the x-coordinate of the intersection-point of the two constraint functions $\alpha_{\text{Max}}(T_{\text{Max}})$ and $\alpha_{\text{Min}}(T_{\text{Max}})$. Furthermore, it can be readily verified from (46) that:

- $T_0 \leq t_{\text{Lo}}$, $T_0 \leq t_{\text{Off}}$;
- $T_0 > T_{\text{Ser}}$.

where t_{Lo} is the time that is needed to do all the processing locally at the MD and t_{Ser} is the time that would be needed to do all the processing remotely at the MEC server, as defined in (5) and (9).

For the policy specified in the previous section,

$$t^*_{\text{Off}} = t^*_{\text{Tr}} + t_{\text{Ser}}, \tag{47}$$

where t^*_{Tr} is defined in (30). Now, T^*_0 can be evaluated from (46):

$$T^*_0 = \frac{t^*_{\text{Off}} t_{\text{Lo}}}{t^*_{\text{Off}} + t_{\text{Lo}}}, \tag{48}$$

which is the minimum admissible latency under this policy. For the time constraint T_{Max} satisfying $T^*_0 \leq T_{\text{Max}} \leq t_{\text{Lo}}$, the partial offloading is optimal to minimize the total energy consumption for meeting the latency constraint at the same time. If $t_{\text{Lo}} \leq T_{\text{Max}}$, the local computing strategy is optimal in terms of energy and time consumption. Note that in the situation where $t_{\text{Ser}} \leq T_{\text{Max}} \leq T^*_0$, the full offloading is optimal and can meet the tight latency constraint at the expense of high energy consumption.

Interestingly, when the time budget (i.e., the maximum allowed latency T_{Max}) is equal to the minimum affordable latency T_0, partial offloading is required, and the distribution of the task size is given by:

$$\alpha_{\text{Min}} = \alpha_{\text{Max}} \Rightarrow \frac{t_{\text{Lo}}}{t^*_{\text{Off}} + t_{\text{Lo}}}. \tag{49}$$

where the previous expressions were obtained by finding the point where functions $\alpha_{\text{Max}}(T_{\text{Max}})$ and $\alpha_{\text{Min}}(T_{\text{Max}})$ intersect.

4.2. The Channel Gain Threshold as a Function of the System Parameters

For those computation-intensive applications, it is inefficient to run it locally, because too much energy is consumed per CPU cycle at the MD or because the application spends too many CPU cycles per bit. In this situation, the full offloading strategy is more energy and time efficient for MDs. Here, we introduce the parameter bit-wise application complexity, which is defined as $C_{\text{App}} \triangleq \frac{c_{\text{App}}}{d_{\text{App}}}$, and the channel gain threshold g_{Th} can be expressed as a function of f_{MD} and C_{App} as follows:

$$g_{\text{Th}} = \sqrt{\frac{(N_0+I)B}{f_{\text{MD}}}\left(\frac{1}{h_{\text{MD}}} - \frac{1}{h_{\text{MEC}}}\right)\left[2^{\frac{1}{BC_{\text{App}}\left(\frac{1}{h_{\text{MD}}} - \frac{1}{h_{\text{MEC}}}\right)}} - 1\right]}. \tag{50}$$

Figure 4 shows the radius of the effective coverage x_0 of an MEC server installed in a picocell base station for full offloading, based on the relationship between path loss [36] and channel gain $L_T = -20 \log g_{\text{Th}}$. As shown in Figure 4, with the energy efficiency of MD becoming weaker and the application job becoming increasingly complicated, the effective coverage region decreases; thus, more MDs will benefit from MEC in the cell.

Moreover, for asymptotically large C_{App}, the channel gain threshold g_{Th} in (50) can be further approximated as follows:

$$g_{\text{Th}} \approx \sqrt{\frac{(N_0 + I)}{f_{\text{MD}} C_{\text{App}} \ln 2}}. \tag{51}$$

As shown in Figure 5, g_{Th} is largely determined by the local computing power and task complexity.

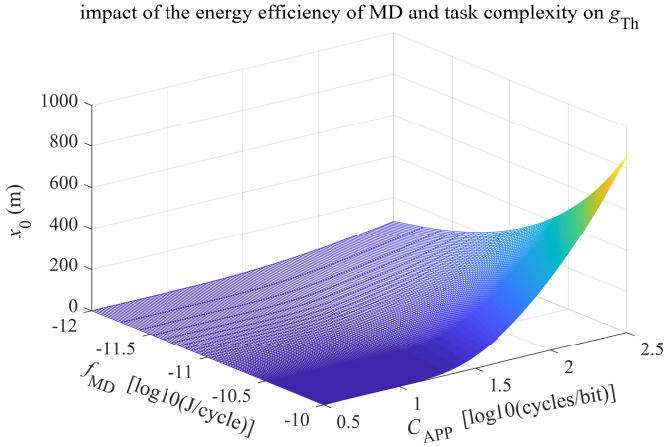


Figure 4. The effective coverage of MEC versus the energy efficiency of MD and task complexity ($h_{\mathrm{MD}} = 1$ GHz, $h_{\mathrm{MEC}} = 10$ GHz, $B = 5$ MHz, and $N_0 + I = -145$ dBm).

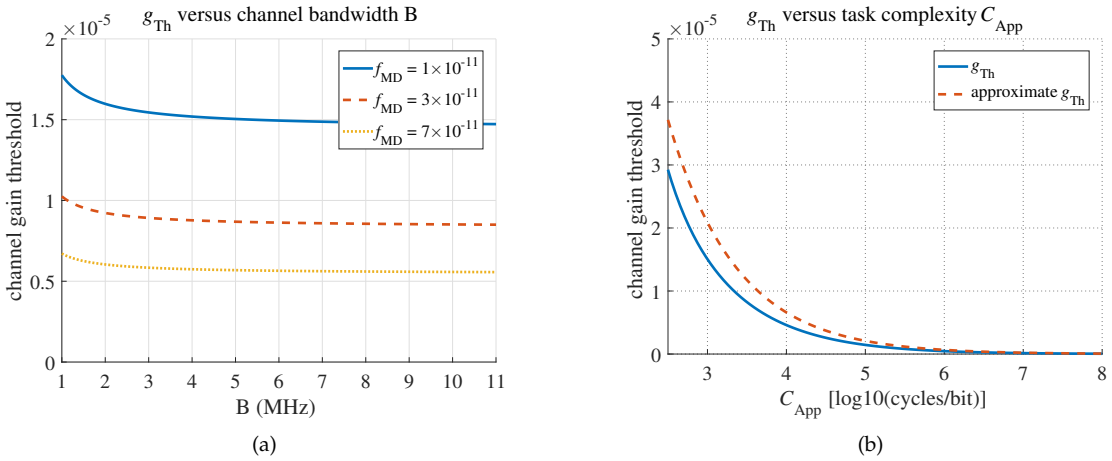


Figure 5. Channel gain threshold g_{Th}. (**a**) g_{Th} versus B. (**b**) g_{Th} versus C_{App}.

In conclusion, the proposed method is applicable to the emerging computational-intensive and/or delay-sensitive applications, i.e., novel AI applications from sectors such as the Industrial Internet of Things, intelligent robots, smart cities, and smart homes.

5. Simulation Results

The following provides simulation results to illustrate the performance of the proposed offloading strategy. We first consider the scenario where the wireless picocell base station had a coverage radius of 500 m and the base station antenna was located inside the building. According to the path loss model for the residential environment, the path loss model is expressed as:

$$L_T = 20 \log f_c[\mathrm{MHz}] + 28 \log x_{\mathrm{d}}[\mathrm{m}] + L_f(n_f) - 28. \tag{52}$$

where $f_c = 2$ GHz is the carrier frequency; x_{d} is the distance between the mobile user and the wireless base station; $L_f(n_f) = 4n_f$ (dB) is the floor penetration factor for the ITU-Rmodel (13.2) [36], and the number of floors $n_f = 1$. We considered the computational task to be generated by a face recognition application, and the task profile refers to [37,38]. The computation profile refers to [38,39]. Moreover, as the mobile devices are evolving to become smarter and the resolution of the images and videos is getting higher, we set the data size for the task to $d_{\mathrm{App}} = 5000$ KB. The simulation parameters are summarized in Table 2. These parameters were used as the default in the subsequent simulations unless explicitly specified. Beforehand, we calculated $t_{\mathrm{Ser}} = 1$ s, $t_{\mathrm{Lo}} = 10$ s, $T_{\mathrm{Sav}} = 9$ s, $e_{\mathrm{Lo}} = 0.1$ J.

Table 2. Parameter setup.

Parameter	Value
B	5 MHz
N_0	−174 dBm
I	−145 dBm
d_{App}	5000 kBytes
c_{App}	1000 Megacycles
f_{MD}	1×10^{-11} J/cycle
h_{MD}	1 GHz
h_{MEC}	10 GHz

The channel gain threshold was calculated from Table 2 where $g_{\text{Th}} = 4.0 \times 10^{-5}$, according to Equation (17). The relationship between channel gain and path loss was $L_T = -20 \log g_{\text{Th}}$; thus, the MEC's full offloading coverage radius $x_0 = 200$ m, which is denoted as the effective coverage in the rest of the paper.

Table 3. Radius of effective coverage x_0.

x_0 * / I \ C_{App} **	200	700	1300	2000	4000	6000
−145 dBm	200	433	606	759	1085	1333
−135 dBm	62	134	187	235	335	412
−125 dBm	20	42	59	74	106	130
−115 dBm	6	13	19	23	33	41
−105 dBm	2	4	6	7	10	13

* Unit of x_0: m. ** Unit of C_{App}: cycles/bit.

In addition, we also list the numerical values of MEC's full offloading coverage radius x_0 for situations of different channel interference strengths and bit-wise complexity C_{App}, as shown in Table 3. For an MEC system, developers can use these values to choose the right location of the MEC servers so as to make full use of their computing resources.

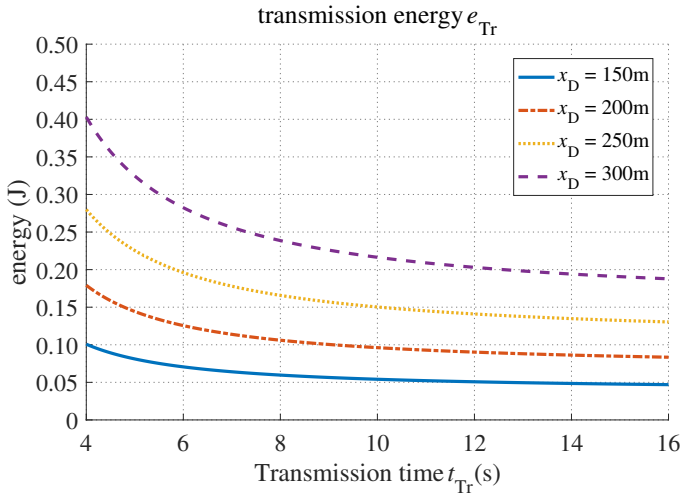


Figure 6. The transmission energy versus transmission time.

Figure 6 shows the relationship between transmission time and transmission energy consumed by the MD. Note that the energy spent was $\frac{N_0 B t_{\text{Tr}}}{g_s^2}\left(2^{\frac{d_{\text{App}}}{B t_{\text{Tr}}}} - 1\right)$. As the transmission time increased, it required less energy for the MD to transmit task data to the MEC server. In addition, when the MD got

closer to the BS, the channel condition became better, i.e., the channel gain improved, and it cost the MD less energy to offload the computing task.

Next, for $g_s > g_{\mathrm{Th}}$, we verified that the full offloading policy was optimal in terms of minimizing energy consumption, as shown in Figure 7.

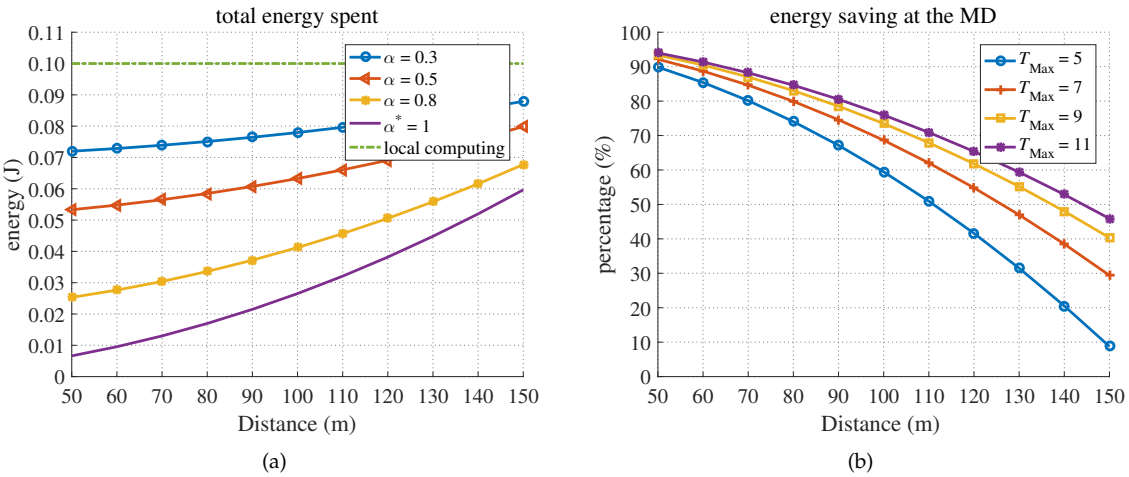


Figure 7. The energy consumption and percentage of energy savings due to offloading under different latency constraints versus the distance between the BS and mobile device (MD). (**a**) Energy consumption. (**b**) Energy savings.

Figure 7a shows the energy consumption for different distances between the BS and MD in the MEC-effective coverage of $g_s > g_{\mathrm{Th}}$. As shown in the figure, for $g_s > g_{\mathrm{Th}}$, the full offloading policy was optimal for saving energy, which corresponded to the optimum solution of (16). In such a situation and $T_{\mathrm{Max}} \geq T_1$, full offloading energy was always lower than local computing energy. Moreover, only the task data size, the task complexity, and the CPU computing capability of MDs could affect the energy consumption and latency of local computing. Therefore, the changes in the distance between the MD and BS or the task completion time requirement did not influence the performance of local computing. Figure 7b shows the energy savings as a percentage for the case of full offloading. Note that in the case of full offloading, the offloading time is $T_{\mathrm{Max}} - t_{\mathrm{Ser}}$. As shown in the figure, if the latency deadline became less hard, and the distance from the BS decreased (the channel gain increased), the percentage of energy savings would increase, as expected.

In the situation in which the MD was out of the MEC-effective coverage for which $g_s \leq g_{\mathrm{Th}}$, partial offloading was preferred, and we continued to evaluate performance of the energy-optimal policy under various settings.

Figure 8 demonstrates that the proposed partial offloading strategy was optimal for minimizing the energy consumption under the hard delay constraint, and the offloading fraction $\alpha^* = \frac{t_{\mathrm{Lo}} - T_{\mathrm{Max}}}{T_{\mathrm{Sav}} - t^*_{\mathrm{Tr}}}$ increased as the latency deadline became more stringent. As shown in the figure, the total energy consumption increased as the distance from the BS increased.

Figure 9 also shows the energy and time consumption for different distances between the BS and MD for the case of partial offloading. For comparison, full offloading and local computing are included in Figure 9. When $g_s < g_{\mathrm{Th}}$, full offloading consumed more energy than partial offloading and local computing under the same delay constraint, as shown in Section 3. Moreover, the offloading energy increased as the distance from the BS increased, as shown in Figure 9a. We can observe from Figure 9b that the full offloading time was less than the partial offloading and local computing time. In fact, to meet the time constraint of $T_{\mathrm{Max}} \leq t_{\mathrm{Lo}}$, it was necessary for the MD to distribute a part of the computing load to the MEC server; thus, the partial offloading strategy always cost more energy than local computing, as shown in Figure 9a. Additionally, to minimize the energy consumption at the MD, the proportion of the offloaded application decreased with increasing distance. Therefore, more time was consumed on local computing. To compensate for the additional time loss of local computing,

the offloading data rate r_{Tr} must be increased, yielding a shorter full offloading time t^*_{Off}, as shown in Figure 9b.

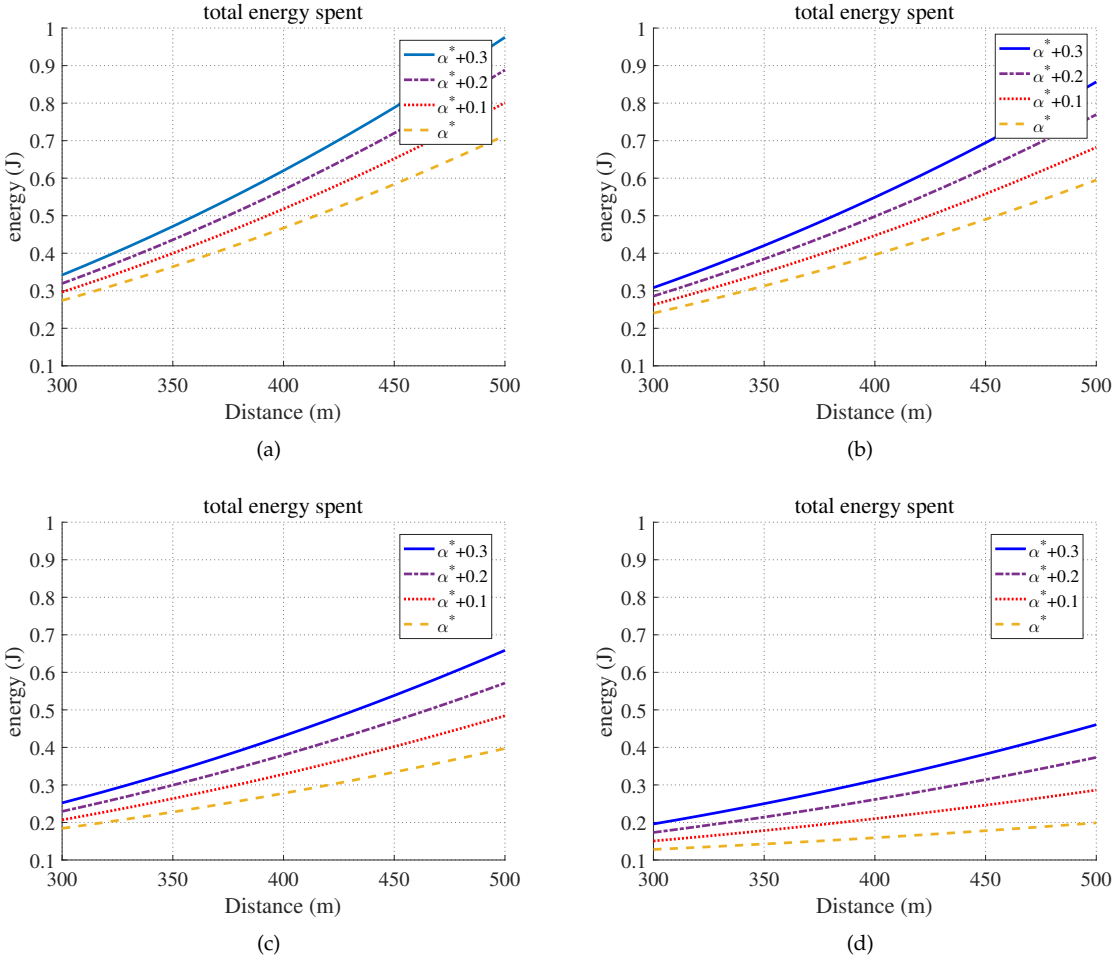


Figure 8. The completion energy consumption versus the distance from the BS under different latency constraints. (**a**) $T_{\mathrm{Max}} = 6.5$ s. (**b**) $T_{\mathrm{Max}} = 7.5$ s. (**c**) $T_{\mathrm{Max}} = 8.5$ s. (**d**) $T_{\mathrm{Max}} = 9.5$ s.

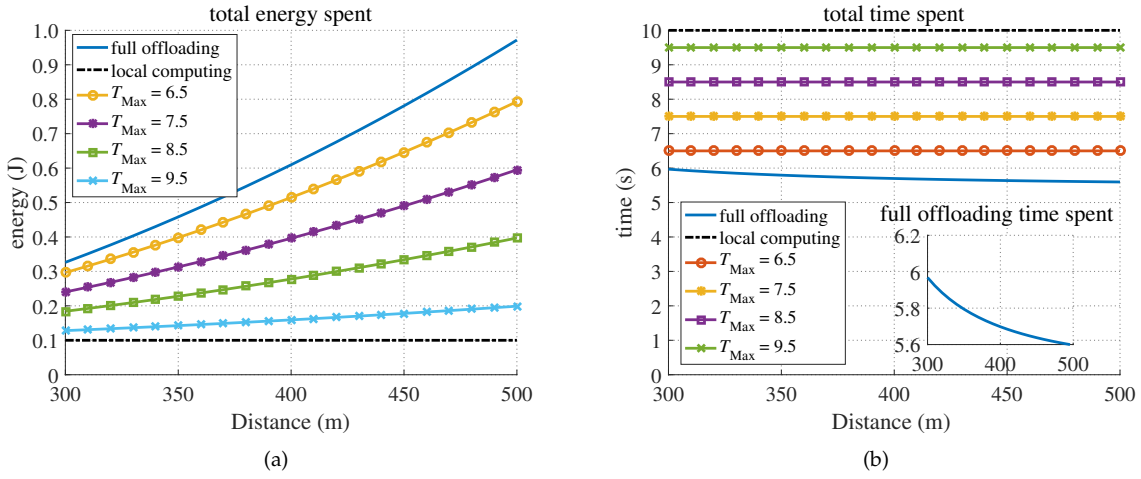


Figure 9. The completion energy and time consumption versus the distance from the BS under different latency constraints. (**a**) Energy consumption. (**b**) Time consumption.

Figure 10 shows the percentage of a task to be remotely processed at the MEC server versus the distance from the BS. The task offloading percentage decreased as the distance from the BS increased, as expected. Note that as the latency constraint became tighter, the offloading percentage increased, and the percentage was also affected by the distance from the BS, which coincided with the optimal solution in (39).

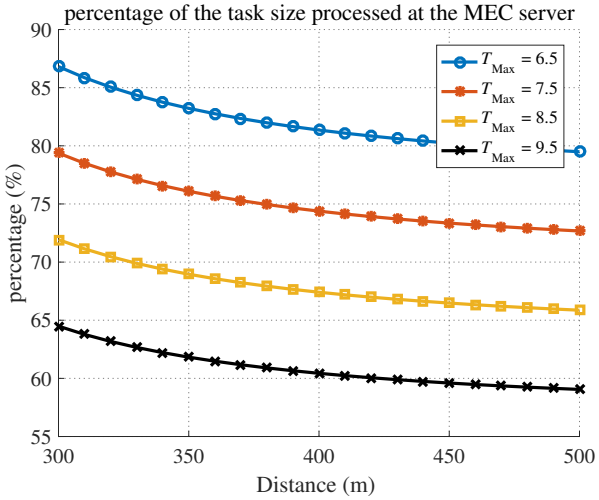


Figure 10. Percentage of the task remotely processed at the MEC server versus the distance from the BS under different latency constraints.

6. Conclusions

This paper presented an offloading strategy for optimizing communication and computation resources usage in an MEC scenario. An energy consumption minimization problem was formulated with the latency constraint. A channel gain threshold of binary offloading and partial offloading was derived, through analyzing the optimization problem. Since the problem was non-convex and difficult to solve directly, we decomposed the original non-convex problem into two sub-problems to obtain the optimal solution in closed form expression. The analytical solution also yielded new understandings of the inherent trade-off between the energy consumption and latency.

For future work, we will extend the strategy to the multi-MD offloading scenario in a multi-access wireless environment. In addition, we are interested in considering a more dynamic model in which the MD may join and leave the MEC server's coverage within a foreseeable period, in which the mobility patterns might play an important role in the problem formulation.

Author Contributions: Conceptualization, X.G. and C.J.; methodology, X.G. and G.Z.; software, X.G. and C.J.; validation, X.G., C.J., and G.Z.; formal analysis, G.Z.; investigation, C.J. and G.Z.; resources, X.G.; data curation, C.J.; writing, original draft preparation, X.G. and C.J.; writing, review and editing, X.G., C.J., and G.Z.; visualization, G.Z.; supervision, C.J. and G.Z.; project administration, G.Z.; funding acquisition, G.Z. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant Nos. 61971245 and 61801249.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Appendix A. Monotonicity of e_{Tot} as a Function of t_{Tr}

The total energy consumption of the MD includes both local computing consumption and partial offloading consumption, which is expressed as:

$$e_{\mathrm{Tot}} = p_{\mathrm{Tr}} t_{\mathrm{Tr}} \alpha + f_{\mathrm{MD}} c_{\mathrm{App}} (1-\alpha) \tag{A1}$$

$$= \frac{t_{\mathrm{Tr}}}{g_s^2} h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right) \alpha + f_{\mathrm{MD}}\, c_{\mathrm{App}} (1-\alpha) \tag{A2}$$

To prove that e_{Tot} is a monotonically decreasing function of t_{Tr}, we need to compute the partial derivative in which:

$$\frac{\partial e_{\mathrm{Tot}}}{\partial\, t_{\mathrm{Tr}}} = \alpha \left[\frac{(N_0+I)B}{g_s^2} \left(2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}} - 1 \right) - \frac{(N_0+I) d_{\mathrm{App}} \ln 2}{g_s^2 t_{\mathrm{Tr}}} 2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}} \right] \tag{A3}$$

$$= \frac{\alpha (N_0+I)B}{g_s^2} \left(2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}} - \frac{d_{\mathrm{App}} \ln 2}{Bt_{\mathrm{Tr}}} 2^{\frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}} - 1 \right). \tag{A4}$$

For notational simplicity, let $v \triangleq \frac{d_{\mathrm{App}}}{Bt_{\mathrm{Tr}}}$, and let $f(v) = 2^v - v\, 2^v \ln 2 - 1$, then:

$$\frac{\partial e_{\mathrm{Tot}}}{\partial (t_{\mathrm{Tr}})} = \frac{\alpha (N_0+I)B}{g_s^2} \left(2^v - v \ln 2\, 2^v - 1 \right). \tag{A5}$$

It can be noticed that $f(v)|_{v=0} = 0$, besides:

$$f^{'}(v) = -(\ln 2)^2\, v\, 2^v \leq 0, \quad v \geq 0. \tag{A6}$$

Thus, $f(v) \leq 0$ for $v \geq 0$. Finally, by noticing that:

$$\frac{\partial\, e_{\mathrm{Tot}}}{\partial\, t_{\mathrm{Tr}}} = \frac{\alpha (N_0+I)B}{g_s^2} f(v) \leq 0, \tag{A7}$$

the lemma is proved.

Appendix B. The Existence of Channel Threshold g_{Th}

Let us consider the case in which the offloading task can be finished in the same time interval as the local computing time in which $t_{\mathrm{Max}} = t_{\mathrm{Lo}}$. By referring to Equations (10) and (12), one finds that $t_{\mathrm{Tr}} = T_{\mathrm{Sav}}$. Therefore, the energy consumption in the transmitting stage can be written as a function of channel gain g_s as follows:

$$e_0(g_s) = \frac{1}{g_s^2} h\left(\frac{d_{\mathrm{App}}}{t_{\mathrm{Tr}}}\right)$$

$$= \frac{1}{g_s^2} (N_o + I) B T_{\mathrm{Sav}} \left(2^{\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}} - 1 \right). \tag{A8}$$

Next, one may compare $e_0(g_s)$ with local computing energy consumption, which is $f_{\mathrm{MD}} c_{\mathrm{App}}$. As e_0 is an inverse proportional function of g_s in (A8), there exists a threshold g_{Th}, such that for $g_s > g_{\mathrm{Th}}$, the offloading energy is less than the local $f_{\mathrm{MD}} c_{\mathrm{App}}$. Therefore,

$$g_{\mathrm{Th}} = \sqrt{\frac{(N_0+I) B T_{\mathrm{Sav}} \left(2^{\frac{d_{\mathrm{App}}}{BT_{\mathrm{Sav}}}} - 1 \right)}{f_{\mathrm{MD}} c_{\mathrm{App}}}}. \tag{A9}$$

For channel conditions better than threshold g_{Th}, full offloading with $\alpha^* = 1$ is optimal because in this case, transmitting the data to the MEC server always consumes less energy than local computing under the same latency constraint.

The result above can now be extended to prove that full offloading ($\alpha^* = 1$) is optimal for the general channel condition $g_s > g_{\mathrm{Th}}$. The constraints of $T_{\mathrm{Max}} < t_{\mathrm{Lo}}$ and $T_{\mathrm{Max}} > t_{\mathrm{Lo}}$ are discussed as follows.

1. If $T_{\mathrm{Max}} < t_{\mathrm{Lo}}$, then offloading has to be performed because the time constraint cannot be totally satisfied by full local computing. Now, assume if there exists some fraction yet to be locally processed, then it must not be the optimal choice because offloading requires less energy than local computing for $g_s > g_{\mathrm{Th}}$ under the same time constraint.
2. If $T_{\mathrm{Max}} > t_{\mathrm{Lo}}$, then $t_{\mathrm{Tr}} > T_{\mathrm{Sav}}$. By Lemma 1, offloading will further reduce energy consumption due to the longer transmission time, which is advantageous to local computing.

Appendix C. Proof the Monotonicity of e_{Tot} with Respect to α When $g_s < g_{\mathrm{Th}}$

Based on (30), it is obvious that $\frac{\partial t^*_{\mathrm{Tr}}}{\partial (g_s^2)} \geq 0$ and t^*_{Tr} is equal to T_{Sav} at the channel condition threshold $g_s = g_{\mathrm{Th}}$; thus, if $g_s \leq g_{\mathrm{Th}}$, we have $t^*_{\mathrm{Tr}} \leq T_{\mathrm{Sav}}$. In addition, e^*_{Tot} satisfies that $\frac{\partial e^*_{\mathrm{Tot}}}{\partial t^*_{\mathrm{Tr}}} \leq 0$, which was proven in Section 3.3. Therefore, if $t^*_{\mathrm{Tr}} \leq T_{\mathrm{Sav}}$, we can calculate that:

$$\frac{\partial e_{\mathrm{Tot}}}{\partial \alpha} = \frac{(N_0 + I) B t_{\mathrm{Tr}}}{g_s^2} \left(2^{\frac{d_{\mathrm{App}}}{B t_{\mathrm{Tr}}}} - 1 \right) - f_{\mathrm{MD}} c_{\mathrm{App}} \tag{A10}$$

$$\geq \frac{(N_0 + I) B T_{\mathrm{Sav}}}{g_s^2} \left(2^{\frac{d_{\mathrm{App}}}{B T_{\mathrm{Sav}}}} - 1 \right) - f_{\mathrm{MD}} c_{\mathrm{App}} \tag{A11}$$

$$\geq \frac{(N_0 + I) B T_{\mathrm{Sav}}}{g_{\mathrm{Th}}^2} \left(2^{\frac{d_{\mathrm{App}}}{B T_{\mathrm{Sav}}}} - 1 \right) - f_{\mathrm{MD}} c_{\mathrm{App}} \tag{A12}$$

$$= 0$$

References

1. Porambage, P.; Okwuibe, J.; Liyanage, M. Survey on multi-access edge computing for Internet of Things realization. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2961–2991. [CrossRef]
2. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge computing for the Internet of Things: A case study. *IEEE Internet Things J.* **2018**, *5*, 1275–1284. [CrossRef]
3. Mach, P.; Becvar, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [CrossRef]
4. Abbas, N.; Zhang, Y.; Taherkordi, A. Mobile edge computing: A survey. *IEEE Internet Things J.* **2017**, *5*, 450–465. [CrossRef]
5. Mobile-Edge Computing-Introductory Technical White Paper. Available online: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-introductory_technical_white_paper_v12018-09-14.pdf (accessed on 14 September 2018.) [CrossRef]
6. Tang, F.; Fadlullah, Z.M.; Kato, N.; Ono, F.; Miura, R. AC-POCA: Anticoordination game based partially overlapping channels assignment in combined UAV and D2D based networks. *IEEE Trans. Veh. Technol.* **2018**, *60*, 1672–1683.
7. Liu, J.; Nishiyama, H.; Kato, K.; Guo, J. On the outage probability of device-to-device communication enabledmulti-channel cellular networks: A RSS threshold-based perspective. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 163–175. [CrossRef]
8. Rodrigues, T.G.; Suto, K.; Nishiyama, H.; Kato, N.; Temma, K. Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration. *IEEE Trans. Comput.* **2018**, *67*, 1287–1300. [CrossRef]
9. Rodrigues, T.G.; Suto, K.; Nishiyama, H.; Kato, N. Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control. *IEEE Trans. Comput.* **2017**, *66*, 810–819. [CrossRef]

10. Dong, L.; Li, R. Distributed mechanism for computation offloading task routing in mobile edge cloud network. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019. [CrossRef]
11. Kuang, Z.; Li, L.; Gao, J.; Zhao, L.; Liu, A. Partial offloading scheduling and power allocation for mobile edge computing systems. *IEEE Internet Things J.* **2019**, *6*, 6774–6785.
12. Ren, J.; Yu, G.; Cai, Y.; He, Y.; Qu, F. Partial offloading for latency minimization in mobile-edge computing. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017. [CrossRef]
13. Gao, Y.; Cui, Y.; Wang, X.; Liu, Z. Optimal resource allocation for scalable mobile edge computing. *IEEE Commun. Lett.* **2019**, *23*, 1211–1214.
14. Feng, J.; Liu, Z.; Wu, C.; Ji, Y. AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10660–10675. [CrossRef]
15. Wu, H.; Knottenbelt, W.J.; Wolter, K. An efficient application partitioning algorithm in mobile environments. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 1464–1480. [CrossRef]
16. Huaming, W. Multi-objective decision-making for mobile cloud offloading: A survey. *IEEE Access* **2018**, *6*, 3962–3976. [CrossRef]
17. Ko, S.W.; Huang, K.; Kim, S.L.; Chae, H. Live prefetching for mobile computation offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3057–3071.
18. Mahmoodi, S.E.; Uma, R.N.; Subbalakshmi, K.P. Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Trans. Cloud Comput.* **2016**, *7*, 301–313. [CrossRef]
19. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [CrossRef]
20. Bi, S.; Zhang, Y. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4177–4190. [CrossRef]
21. Zhang, H.; Guo, F.; Ji, H.; Zhu, C. Combinational auction-based service provider selection in mobile edge computing networks. *IEEE Access* **2017**, *5*, 13455–13464. [CrossRef]
22. Al-Shuwaili, A.; Simeone, O. Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 398–401. [CrossRef]
23. You, C.; Zeng, Y.; Zhang, R.; Huang, K. Asynchronous mobile-edge computation offloading: Energy-efficient resource management. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 7590–7605. [CrossRef]
24. Mao, Y.; Zhang, J.; Letaief, K.B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3590–3605. [CrossRef]
25. You, C.; Huang, K.; Chae, H. Multiuser resource allocation for mobile-edge computation offloading. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 1–6 December 2016. [CrossRef]
26. Wang, F.; Xu, J.; Wang, X.; Cui, S. Joint offloading and computing optimization in wireless powered mobile-edge computing systems. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 1784–1797. [CrossRef]
27. Cao, X.; Wang, F.; Xu, J.; Zhang, R.; Cui, S. Joint computation and communication cooperation for energy-efficient mobile edge computing. *IEEE Internet Things J.* **2019**, *6*, 4188–4200.
28. Yang, L.; Zhang, H.; Li, M.; Guo, J.; Ji, H. Mobile edge computing empowered energy efficient task offloading in 5G. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6398–6409. [CrossRef]
29. Tao, X.; Ota, K.; Dong, M.; Qi, H. Performance guaranteed computation offloading for mobile-edge cloud computing. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 774–777. [CrossRef]
30. Gao, G.; Xiao, M.; Wu, J. Opportunistic mobile data offloading with deadline constraints. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 3584–3599. [CrossRef]
31. Meng, X.; Wang, W; Zhang, Z. Delay-constrained hybrid computation offloading with cloud and fog computing. *IEEE Access* **2017**, *5*, 21355–21367. [CrossRef]
32. Mao, Y.; Zhang, J; Letaief, K.B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017. [CrossRef]
33. Liu, L.; Chang, Z.; Guo, X.; Mao, S. Multiobjective optimization for computation offloading in fog computing. *IEEE Internet Things J.* **2018** *5*, 283–294. [CrossRef]

34. Boyd, S.; Vandenbergh, L. Convex functions. In *Convex Optimization*; Publishing House: London, UK, 2004, pp. 67–103.
35. Corless, R.M.; Gonnet, G.H.; Hare, D.E.G.; Jeffrey, D.J.; Knuth, D.E. On the LambertW function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [CrossRef]
36. Path Loss Models: S-72.333 Physical Layer Methods in Wireless Communication Systems. Available online: http://www.comlab.hut.fi/opetus/333/2004_2005_slides/Path_loss_models.pdf (accessed on 24 May 2020.)
37. Jeongho, K.; Yeongjin, K.; Joohyun, L.; Song, C. DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 2510–2523.
38. Changsheng, Y.; Kaibin, H.; Hyukjin, C.; Byoung-Hoon, K. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [CrossRef]
39. Songtao, G.; Jiadi, L.; Yuanyuan, L.; Bin, X.; Zhetao, L. Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing. *IEEE J. Sel. Areas Commun.* **2019**, *18*, 319–333. [CrossRef]