

Article

Resource Allocation for Edge Computing without Using Cloud Center in Smart Home Environment: A Pricing Approach

Huan Liu, Shiyong Li * and Wei Sun

School of Economics and Management, Yanshan University, Qinhuangdao 066004, China; liuhuan5011@foxmail.com (H.L.); wsun@ysu.edu.cn (W.S.)

* Correspondence: shiyongli@ysu.edu.cn; Tel.: +86-335-805-7025

Received: 26 September 2020; Accepted: 14 November 2020; Published: 16 November 2020



Abstract: Recently, more and more smart homes have become one of important parts of home infrastructure. However, most of the smart home applications are not interconnected and remain isolated. They use the cloud center as the control platform, which increases the risk of link congestion and data security. Thus, in the future, smart homes based on edge computing without using cloud center become an important research area. In this paper, we assume that all applications in a smart home environment are composed of edge nodes and users. In order to maximize the utility of users, we assume that all users and edge nodes are placed in a market and formulate a pricing resource allocation model with utility maximization. We apply the Lagrangian method to analyze the model, so an edge node (provider in the market) allocates its resources to a user (customer in the market) based on the prices of resources and the utility related to the preference of users. To obtain the optimal resource allocation, we propose a pricing-based resource allocation algorithm by using low-pass filtering scheme and conform that the proposed algorithm can achieve an optimum within reasonable convergence times through some numerical examples.

Keywords: edge computing; smart homes; resource pricing; resource allocation; utility optimization

1. Introduction

With the emergence and rapid development of the Internet of Things (IoT), Internet technology is moving towards the direction of “intelligence of everything” on the basis of “Internet of everything (IOE)” [1]. People’s quality of life is steadily improving and they are also looking for more convenient and comfortable services due to technological progress. Currently, smart home applications represented by smart speakers, sweeping robots and smart air conditioners are becoming an indispensable part in many users’ lives [2]. In China, the scale of smart home market is growing at a rate of 20% to 30% per year. According to Prospective Industrial Research Institute, China’s smart home market is expected to reach ¥436.9 million in 2021 [3]. A mushrooming number of smart home applications are entering into millions of homes, from smart washing machines to smart vacuum cleaners, from smart door locks to telemedicine, all of them reflect the charm and potential of technology. The new technology of smart homes and Internet of Things would meet people’s pursuit of “livable, comfortable, convenient and safe” living environment [4].

The Smart Grid (SG) is considered as an imminent future power network due to its fault identification and self-healing capabilities. As one of the most significant terminal units in the smart grid, in the future, it is inevitable that the deep integration of smart home and SG becomes a development trend whose direction is mainly reflected in three aspects. Firstly, in terms of smart home energy production and utilization, the openness of SG determines that clients have the dual

attributes of consumers and producers of electric energy. With the development of SG, more and more smart homes will begin to install distributed energy equipment and use microgrid technology for flexible control, forming an effective supplement to the grid and achieving a win-win for the grid and clients. Secondly, in aspects of energy consumption, the simultaneous management of household energy and information can be realized through refined demand-side management. Finally, when it comes to services, by realizing the real-time two-way interaction between the grid and home, it can guide clients to actively participate in the construction and management of SG and provide clients with better services [5].

However, when a wide variety of smart home applications with diametrically different functional segmentation are connected and controlled over network, there are many problems such as data privacy, security, limited bandwidth in trunk links, serious energy consumption, link congestion, etc., especially data security [6]. Generally smart home applications are intelligent independently, such as a smart security system and a surveillance camera typically, they rely on the cloud platform to achieve remote control, in the event of a network malfunction, the clients may lose control of the applications [7]. Moreover, the smart home does not enable entirely coordinated work of multiple smart devices in conjunction with each other [8]. However, the emergence of edge computing technology makes it possible for intelligent connectivity and stable operation among smart home applications.

Edge computing is also known as “Fog Computing”, which refers to inserting an intermediate node layer between user and cloud computing data center [9]. The intermediate node layer is closer to users in the network topology and located in the edge of the whole network [10]. As a supplement and continuation of cloud computing, edge computing has the characteristics of strong scalability, low latency, high mobility, bandwidth saving, low energy consumption, system security and high performance [11]. In recent years, edge computing has been applied into vehicle networking, HD (High Definition) video transmission, VR (Virtual Reality), AR (Augmented Reality) and other fields [12,13]. The edge nodes deployed in the client’s home collect information through wired/wireless sensors. The information includes indoor temperature, brightness, air humidity and image information collected by outdoor/indoor cameras for comprehensive analysis. Then the edge nodes issue specific commands to smart home applications to complete the intelligent linkage control of the client’s home lighting, temperature system and alarm mechanism.

According to network topology, the conventional structure of edge computing contains three layers: cloud layer, edge layer and user layer [14]. As shown in Figure 1, cloud layer refers to the remote large-scale cloud center, which provides necessary data processing support for smooth operation of fog layer. Due to its high energy consumption, large volume, cooling process and optimization of transmission nodes, cloud center is subject to many conditions in layout location. Edge layer, as the name suggests, is closer to the “edge” (user) than the “core” (cloud center). The edge layer contains a great number of edge nodes, which are located around users and act as small computing centers with the ability of processing tasks, storing data and connecting to the network [15]. The user layer refers to the terminal facilities that directly serve clients. It not only includes mobile phones, computers and other electronic devices, but also includes the terminals that continue to be endowed with wisdom under the general trend of IoT, such as smart air conditioners, sweeping robots, precision medical devices, et al. However, in a smart home environment, it is necessary for smart home applications to operate independently from cloud center due to link congestion, data security, time delay and privacy [16,17]. Thus, we design an edge computing resource allocation structure without using cloud center for this scenario. By allocating and provisioning resources among smart home applications, smart homes can get rid of the reliance on resource from cloud center. In fact, this structure of resource allocation has been applied in many fields [18]. For instance, the Unmanned Aerial Vehicle (UAV) flight data collection system stores the collected data in the Edge Data Center (EDC) in real time and immediately executes the data analysis algorithm without communicating with cloud center [19]. Then the control platform can flexibly adjust the task flow according to the analysis results.

At present, there are two mainstream resource allocation methods for edge computing without using cloud center [20]. One is from the perspective of users, in order to obtain the optimal resource allocation, the most important issue is to select the appropriate edge nodes to perform the calculation migration. This method assumes that all users are in a hotspot area that covers edge computing networks and can access multiple edge nodes nearby. Thus, the users take advantage of the game theory and operational research to select the best node and form optimal resource scheduling strategies [21]. The other is from the perspective of edge nodes. The key consideration is how to reasonably form edge node clusters to achieve the optimization of the overall performance of the system. This method believes that edge nodes need to complete resources through the collaborative work [22,23]. This paper attempts to take the participants, edge nodes and users (all smart home applications) of resource allocation into a market, and use the rules of market equilibrium to achieve optimal resource allocation, which can overcome the shortcomings of only considering one party in the process of resource allocation in the above research.

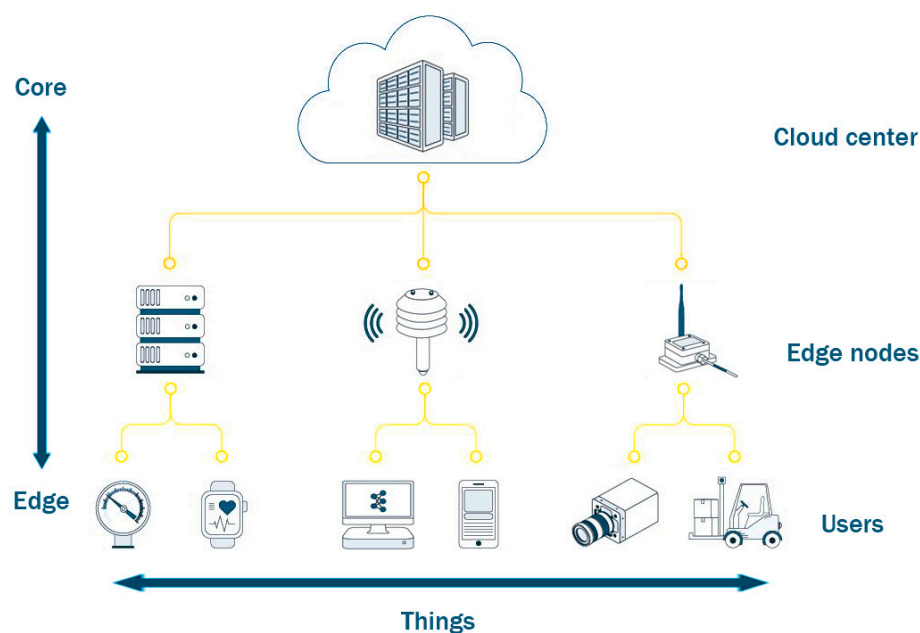


Figure 1. The conventional structure of edge computing.

In this paper, we apply the resource pricing and the utility theory in economics to investigate the resource allocation and scheduling of edge computing without using cloud center for smart home environment. In order to make the modeling process simpler, we take bandwidth resources as an example to investigate the resource allocation of edge computing for smart homes. Although it is at present that the available bandwidth exceeds significantly the local application demands, we think that it is worth discussing the bandwidth allocation in smart home environment in the future. The reasons why we believe it are as follows. On the one hand, at the supply side, the last decade has witnessed an explosion of data traffic over the network attributed to the development of mobile communication. With the advent and extension of fifth-generation (5G) mobile networks, it is inevitable to emerge the situation of the exponential rise in data-demand and increasing demand in bandwidth. On the other hand, at the demand side, with the improvement of people's quality of life and soaring demand for more convenient and comfortable services, it may appear in our life that a variety of IoT applications and smart homes including 4K/8K Ultra High Definition (UHD) video and virtual/augmented reality (VR/AR) require low-latency and high-bandwidth. We also divide the tasks generated by smart home applications into two types: the tasks with low requirement on latency and high requirement on processing capacity, and the tasks with high requirement on latency and low requirement on processing capacity. For the first type of tasks, we introduce the utility function

aiming at optimizing resource transmission delay, while for the second type, we introduce the utility function aiming at optimizing resource transmission volume. Different utility functions are chosen to describe and maximize the characteristics of users when dealing with these two types of tasks. Based on the opinion of maximizing utility and the rules of market equilibrium, we propose the utility maximization model of resource allocation for different tasks of edge computing without using cloud center. Subsequently, we analyze the optimal resource allocation for the two types of tasks and obtain the optimum for each one. Then, we design the resource allocation algorithm by using the Lagrangian method and low-pass filtering theory which is assisted to eliminate oscillations and increase convergence speed. Finally, some numerical examples are given to verify the performance of the algorithm for two types of tasks.

2. Related Work

With the economic boom and technical advancement, an increasing number of smart devices find their way into people's daily life, which have also received a lot of research attention from various communities like researchers, business and government. Lee et al. [24] designed a two-level Deep Reinforcement Learning (DRL) framework for optimal energy management of smart homes. In the proposed simulation, two agents (an air conditioner and a washing machine) interact with each other to schedule the optimal home energy consumption efficiently. Al-Ali et al. [25] presented an energy management system for smart homes and better meeting the needs of clients through big data analytics and business intelligence. Wang et al. [26] proposed a new task scheduling approach to fulfil the requirements of smart homes and healthcare regarding tight set situations of the elderly or patients, using edge computing-themed processing schemes with a focus on real-time issues. Procopiou et al. [27] proposed a lightweight detection algorithm for IoT devices based on chaos prediction and chaos theory–Chaos Algorithm (CA) which is used for the identification of Flooding and Distributed Denial of Service (DDoS) attacks to provide a secure network environment for IoT-based smart home systems. Yang et al. [28] proposed an IoT-based smart home security monitoring system that can enhance the security performance of system by improving the False Positive Rate (FPR) and reducing the network latency over the traditional security system.

At present, for the problem of resource allocation for edge computing without using cloud center, there are two main types of research approach. The first one is from the user's perspective, Guo et al. [22] illustrated that it is an important method for achieving optimal resource allocation to filter out appropriate edge nodes to perform computational migration. The authors assumed that all users are within range of edge computing hotspots and have access to multiple edge servers. They proposed an optimal task scheduling strategy based on a discrete Markov decision process (MDP) framework to achieve optimal allocation of tasks and resources. Meanwhile, they also presented an index-based allocation strategy to reduce the computation complexity and communication overheads associated with the implementation of this policy. Finally, each user can find the most suitable edge server to minimize energy consumption and latency. The second one is from the edge node's perspective, Queis et al. [23] analyzed the impact of cluster size (i.e., the number of edge nodes performing computational tasks) on service latency and energy consumption of edge nodes. Their research showed that increasing the number of edge nodes does not always reduce execution latency; If the transmission latency is longer than the computational latency at the edge nodes, the overall service latency may increase. Therefore, proper construction methods for edge node cluster and node selection methods play a crucial role in system performance. Queis et al. [29] proposed three different clustering strategies regarding optimization of service latency, overall energy consumption of the clusters and energy consumption of the nodes of the clusters in order to validate the impact of different clustering strategies on edge node cluster characteristics (size, latency, and energy consumption). Li et al. [30] proposed a self-similarity-based load balancing (SSLB) mechanism for large-scale fog computing which focuses on the applications processed on fog infrastructure. They also presented an adaptive threshold policy and corresponding scheduling algorithm, which guarantees the efficiency of SSLB. Queis et al. [31]

designed a two-step method for implementing user task scheduling and resource allocation of nodes. The first step is local resource allocation, where each edge node allocates its resources to nearby users according to specific scheduling rules; The second step is the creation of a cluster of edge nodes for users that were not allocated compute resources in the first step. Sun et al. [32] proposed a two-level resource scheduling model, which includes resource scheduling among various fog clusters and resource scheduling among fog nodes in the same fog cluster. Besides, they believed that the edge layer and the terminal layer partially intersect because the mobile terminal devices in these intersections are not only fog resource requesters but also resource providers.

Zhou et al. [33] proposed a cloud resource scheduling algorithm based on Markov prediction model which is put forward to solve the problem of task scheduling and load balancing in cloud service node failure situation, including the judgment of node load degree, the selection of migrated task and nodes, and the decision of migration routing. The goal is to achieve rapid cloud service recovery and to improve the reliability of cloud services. Zheng et al. [34] proposed an energy consumption optimization problem that considers latency performance. The model includes energy consumption and latency in the process of execution and transmission of local devices, fog nodes and cloud centers. Liu et al. [11] presented a joint model including consumption, latency, and payment costs in fog computing heterogeneous networks to solve a multi-objective optimization problem between nodes and users by using queueing theory and operations research. Wang et al. [35] designed an unloading system in real-time traffic management based on fog node-based IoV (Internet of Vehicle) system, which expands cloud computing by using parking and moving vehicles as fog nodes.

In the research of resource allocation in the network, Li et al. [36] applied the hybrid cloud resource optimization model to the mobile client, and used a two-stage optimization process to provide a scheme for resource allocation of mobile client in mobile cloud environment. The first stage emphasizes that mobile cloud users achieve utility optimization under cost and energy constraints, and the second stage is that mobile cloud providers run multiple servers within the energy loss limit to perform the work of mobile cloud users. Li et al. [37] proposed a resource allocation model for elastic services based on utility optimization with the objective of maximizing network utility from the perspective of users' utility optimization. In allocating bandwidth resources for migrating enterprise applications to the cloud, Li et al. [38] set the objective function to the problem of transmission time optimization, which makes the utility and satisfaction of enterprise users increase by minimizing cloud migration time. Nguyen et al. [39] elaborated a price-based edge computing method of resource allocation, which mainly expounds the multiple competitive service processes from heterogeneous edge nodes with limited ability to network edge under the efficient resource allocation framework of the market. In the process of explanation, the framework proved that the balanced allocation of resources realized Pareto optimization and met the fairness, proportionality and sharing incentives of allocation expectations.

3. Resource Allocation Model of Edge Computing without Using Cloud Center for Smart Homes

In a smart home environment, we assume that all smart home applications are composed of edge nodes and users, which mainly depends on whether their capacities meet task requirements. The tasks generated by the users in edge computing without using cloud center can only be solved at the edge nodes. Although the edge nodes are closer to the users in the topological position, when it comes to process different types of tasks, especially the tasks with large amount of calculation and storage, this paradigm has a massive number of limitations. In fact, there are large amounts of tasks that may appear in edge computing. We assume that these tasks of smart homes environment can be divided into two types: one is corresponding to the tasks with low requirement on latency and high requirement on processing capacity, and the other is associated with the tasks with high requirement on latency and low requirement on processing capacity. In this section, inspired by the concept of utility in economics, we formulate the resource allocation model of edge computing for smart home environment without using cloud center and adopt various utility strategies to process properly different types of tasks.

3.1. Model Description

In the traditional edge computing structure, the remote cloud center, edge nodes and users form a close relationship through the networks. However, considering the data security and privacy of smart homes, avoiding the hosting costs incurred by uploading tasks to the cloud center and reducing the latency of smart home applications, we design a new edge computing structure for the smart home that does not require the involvement of cloud center. As shown in Figure 2, the user represents a smart home application whose capacity does not meet the task requirements and the edge node is a smart home application at the bottom of network structure that has spare computing power and provides the compute and storage resources for its surrounding users. The arrow direction represents the tasks generated by users that are processed by nearby edge nodes. Their spare computing power eliminates the need to transmit data to the cloud center and decentralizes processing power to ensure real-time processing with reduced latency, while diminishing bandwidth and network storage requirements. In fact, many smart home applications both act as users sending task requests to surrounding edge nodes and as edge nodes receiving and resolving tasks from surrounding users in a smart home environment. For example, a mobile phone can act as an edge node between a smart bracelet and a cloud center, processing service requests from the bracelet about exercise and health monitoring, and as a user generating tasks about games, videos, social networks, et al. However, the tasks generated by users can only be processed by the adjacent edge nodes in a smart home environment due to the lack of cloud center. For the purpose of processing various tasks in different strategies properly, the tasks generated by users or received by edge nodes are divided into two types. As shown in Figure 2, the red tasks represent the tasks with high requirement on latency and low requirement on processing capacity (For example, game commands in phone, shopping online by Taobao VR) and the green tasks represent the tasks with low requirement on latency and high requirement on processing capacity (For example, watching HD video on Smart TV). By adopting different processing strategies for the two kinds of tasks, the model can take advantage of the relationship of edge nodes and users to solve resource allocation in edge computing without the participation of cloud center.

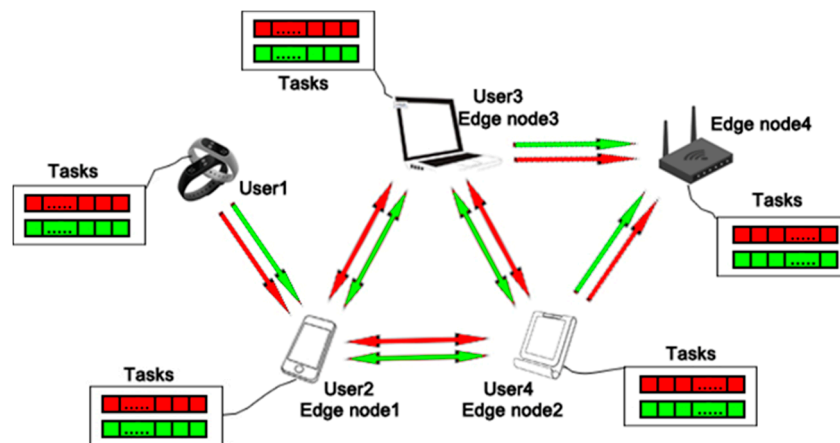


Figure 2. The structure of edge computing without using cloud center in smart home environment.

3.2. Model Formulation

Due to security and privacy requirements, it is better to divide the entire network into multiple non-connected regions. In this paper, we focus on a smart home environment which should be regarded as an independent region to avoid the network intrusion and attack. Thus, we assume that all smart home applications are in a home (namely an independent region) and are not connected with cloud center, then they are considered in a resource allocation market. In the market, all nodes (smart home applications) are divided into customers (users) and resource providers (edge nodes) according to the contents of the previous section. In the following description, we use “resource provider/provider”

instead of edge node and “resource consumer/consumer” instead of user. We take bandwidth resources as an example to intuitively analyze the resource allocation of edge computing in a smart home environment. Each customer sends a task processing request to one or more resource providers. Each resource provider receives requests from one or several resource customers and provides them with an appropriate amount of resources. In order to differentiate between resource providing nodes and resource requesting nodes in the resource allocation model and distinguish resource allocation requirements for different types of tasks, we introduce the set P of resource providers, the set R of resource customers for the tasks with high requirement on latency and low requirement on processing capacity and the set S of resource customers for the tasks with low requirement on latency and high requirement on processing capacity. A node is a member of P , R or S if it provides or requests at least one content of the two kinds of tasks, respectively. Of course, one node can be a member of all sets. Let $P(r)$ or $P(s)$ be the set of providers offering resources for customers $r \in R$ or $s \in S$. Thus, if a provider $p \in P$ offers resources for customer $r \in R$ or $s \in S$, then $p \in P(r)$ or $p \in P(s)$. Also, let $S(p)$ or $R(p)$ be the set of customers that request resources from provider p for the two types of tasks, Thus, if customer s or r requests resources from provider p , then $s \in S(p)$ or $r \in R(p)$. Note that only when $s \in S(p)$, $r \in R(p)$, then $p \in P(s)$, $p \in P(r)$.

Assume that the resource allocation between providers and customers in a smart home environment can be viewed as offering a certain amount of requested bandwidth. Each provider receives requests from customers and needs to serve the customers by providing with a certain amount of bandwidth. When processing a task with low requirement on latency and high requirement on processing capacity, denote the resources granted by provider p to customer s as x_{ps} , thus the aggregate resources granted by providers to customer s is $y_s = \sum_{p:p \in P(s)} x_{ps}$ and the resources provided by provider p to all its customers is $z_p^L = \sum_{s:s \in S(p)} x_{ps}$, the utility generated by the customer s is $U_s(y_s)$. Similarly, when processing a task with high requirement on latency and low requirement on processing capacity, denote the resources granted by provider p to customer r as x_{pr} , thus the aggregate resources granted by providers to customer r is $y_r = \sum_{p:p \in P(r)} x_{pr}$ and the resources provided by provider p to all its customers is $z_p^H = \sum_{r:r \in R(p)} x_{pr}$, the utility generated by the customer r is $U_r(y_r)$. Meanwhile, the resource allocation between providers and customers is subject to their capacity. Denote C_r^d , C_s^d as the capacity of bandwidth resources of the downstream links of customers r and s , respectively. Denote C_p^{uH} , C_p^{uL} as the capacity of reserved bandwidth resources of the upstream link of provider p for the tasks with high requirement on latency and the tasks with low requirement on latency, which are discussed in previous part. Therefore, for each provider p , we can obtain that $z_p^L \leq C_p^{uL}$, which means that the total granted resources for customers requesting the tasks with low requirement on latency and high requirement on processing capacity do not exceed the reserved upstream link capacity C_p^{uL} . For each provider p , we also obtain that $z_p^H \leq C_p^{uH}$, which means that the total granted resource for customers requesting the tasks with high requirement on latency and low requirement on processing capacity do not exceed the reserved upstream link capacity C_p^{uH} . For each customer s , $y_s \leq C_s^d$, for each customer r , $y_r \leq C_r^d$, which mean that the total obtained bandwidth resources do not exceed the downstream link capacity of each customer.

This paper considers the aggregated utility of all resource customers, which can avoid deviations caused by different resource satisfaction utility of individual customers. Then the optimization problem of the edge computing resource allocation model without using cloud center can be modelled as following.

$$\begin{aligned}
& \text{Max } \sum_{s:s \in S} U_s(y_s) + \sum_{r:r \in R} U_r(y_r) \\
& \text{s.t. } \sum_{p:p \in P(s)} x_{ps} = y_s, \forall s \in S \\
& \quad \sum_{p:p \in P(r)} x_{pr} = y_r, \forall r \in R \\
& \quad \sum_{p:p \in P(s)} x_{ps} \leq C_s^d, \forall s \in S \\
& \quad \sum_{p:p \in P(r)} x_{pr} \leq C_r^d, \forall r \in R \\
& \quad \sum_{s:s \in S(p)} x_{ps} \leq C_p^{uL} \\
& \quad \sum_{r:r \in R(p)} x_{pr} \leq C_p^{uH} \\
& \text{Over } x_{ps} \geq 0, x_{pr} \geq 0, C_s^d, C_r^d, C_p^{uL}, C_p^{uH} \geq 0, s \in S, r \in R, p \in P
\end{aligned} \tag{1}$$

The following utility functions we use in this paper were described in the previous literatures. The Equation (2) is used to achieve a distinguished variant of utility-based fairness, which has drawn a great deal of attention and interest in recent years [40–43]. The Equation (3) is a variant of fairness, which can be reduced to the minimization of time delay and has also been receiving increasing attention [44,45].

$$U_s(y_s) = \omega_s \log(y_s + 1) \tag{2}$$

$$U_r(y_r) = \left(T_r - \frac{M_r}{y_r}\right) \times \frac{\text{sgn}(y_r - B_r) + 1}{2} \tag{3}$$

In the above Equation (2), ω_s refers to the willingness to pay (WTP) of customer s for processing the tasks with low requirement on latency and high requirement on processing capacity [46,47]. In the above Equation (3), M_r refers to the amount of task for customer r . T_r refers to the time limit for customer r . B_r refers to the minimum bandwidth requirement for customer r during resource allocation. When $y_r \leq B_r$, $\left(T_r - \frac{M_r}{y_r}\right) \times \frac{\text{sgn}(y_r - B_r) + 1}{2} = 0$, and when $y_r > B_r$, $\left(T_r - \frac{M_r}{y_r}\right) \times \frac{\text{sgn}(y_r - B_r) + 1}{2} \neq 0$ [48].

The notations used in this paper are summarized in Table 1.

Table 1. Notation list.

Notations	Meanings
S	the set of resource customers (users) for the tasks with low requirement on latency and high requirement on processing capacity, each element is customer s
R	the set of resource customers (users) for the tasks with high requirement on latency and low requirement on processing capacity, each element is customer r
P	the set of resource providers (edge nodes), each element is provider p
$P(s)$	the set of providers offering resources for customers $s \in S$
$P(r)$	the set of providers offering resources for customers $r \in R$
$S(p)$	the set of customers S that request resource from provider p
$R(p)$	the set of customers R that request resource from provider p
x_{ps}	the resources granted by provider p to customer s
x_{pr}	the resources granted by provider p to customer r
y_s	the aggregate resources granted by providers to customer s $y_s = \sum_{p:p \in P(s)} x_{ps}$
y_r	the aggregate resources granted by providers to customer r $y_r = \sum_{p:p \in P(r)} x_{pr}$
z_p^L	the resources provided by provider p to all its customers, $z_p^L = \sum_{s:s \in S(p)} x_{ps}$
z_p^H	the resources provided by provider p to all its customers, $z_p^H = \sum_{r:r \in R(p)} x_{pr}$

Table 1. Cont.

Notations	Meanings
$U_s(y_s)$	the utility generated by the customer s
$U_r(y_r)$	the utility generated by the customer r
C_s^d	the capacity of bandwidth resource of the downstream links of customers s
C_r^d	the capacity of bandwidth resource of the downstream links of customers r
C_p^{uL}	the capacity of reserved bandwidth resource of the upstream link of provider p for customers s
C_p^{uH}	the capacity of reserved bandwidth resource of the upstream link of provider p for customer r
ω_s	the willingness to pay (WTP) of customer s for processing the tasks
M_r	the amount of task for customer r
T_r	the time limit for customer r
B_r	the minimum bandwidth requirement for customer r during resource allocation

3.3. Model Analysis

In the previous section, we have established mathematical model of edge computing without using cloud center for resource allocation in a smart home environment. In this section, we analyze the resource allocation model and regard it as the original problem of the resource allocation model, then we give the Lagrangian function of this nonlinear optimization problem.

$$\begin{aligned}
L(x, y; \lambda, \mu, v, \delta^2) &= \sum_{s:s \in S} \omega_s \log(y_s + 1) + \sum_{r:r \in R} \left[\left(T_r - \frac{M_r}{y_r} \right) \times \left(\frac{\text{sgn}(y_r - B_r) + 1}{2} \right) \right] \\
&+ \sum_{s:s \in S} \lambda_s \left(\sum_{p:p \in P(s)} x_{ps} - y_s \right) + \sum_{s:s \in S} v_s \left(C_s^d - \sum_{p:p \in P(s)} x_{ps} - \delta_s^2 \right) \\
&+ \sum_{r:r \in R} \lambda_r \left(\sum_{p:p \in P(r)} x_{pr} - y_r \right) \\
&+ \sum_{r:r \in R} v_r \left(C_r^d - \sum_{p:p \in P(r)} x_{pr} - \delta_r^2 \right) \\
&+ \sum_{p:p \in P} \mu_p^L \left(C_p^{uL} - \sum_{p:p \in P(s)} x_{ps} - \delta_{ps}^2 \right) \\
&+ \sum_{p:p \in P} \mu_p^H \left(C_p^{uH} - \sum_{r:r \in R(p)} x_{pr} - \delta_{pr}^2 \right)
\end{aligned} \tag{4}$$

where λ is the price vector of elements $\lambda_s, \lambda_r \geq 0$, which can be considered as the price per unit bandwidth paid by customer s and r , respectively; μ is the price vector of elements $\mu_p^L, \mu_p^H \geq 0$, which can be considered as the price per unit bandwidth charged by the upload link of provider p for the two types of tasks; v is the price vector of elements $v_s, v_r \geq 0$, which can be considered as the price paid by the download link of customer s and r , respectively. δ^2 is regarded as the slack variable.

$$\begin{aligned}
L(x, y; \lambda, \mu, v, \delta^2) &= \sum_{s:s \in S} [\omega_s \log(y_s + 1) - \lambda_s y_s] \\
&+ \sum_{r:r \in R} \left[\left(T_r - \frac{M_r}{y_r} \right) \times \left(\frac{\text{sgn}(y_r - B_r) + 1}{2} \right) - \lambda_r y_r \right] \\
&+ \sum_{s:s \in S} \sum_{p:p \in P(s)} x_{ps} (\lambda_s - v_s - \mu_{ps}) + \sum_{r:r \in R} \sum_{p:p \in P(r)} x_{pr} (\lambda_r - v_r - \mu_{pr}) \\
&+ \sum_{s:s \in S} v_s (C_s^d - \delta_s^2) + \sum_{r:r \in R} v_r (C_r^d - \delta_r^2) + \sum_{p:p \in P(s)} \mu_p^L (C_p^{uL} - \delta_{ps}^2) \\
&+ \sum_{p:p \in P(r)} \mu_p^H (C_p^{uH} - \delta_{pr}^2)
\end{aligned} \tag{5}$$

The first two terms in the above equation take y_s and y_r as independent variables, and the third and fourth terms take x_{ps} and x_{pr} as independent variables. The objective function of the dual problem can be written as

$$\begin{aligned}
 D(\lambda, \mu, \nu, \delta^2) &= \max_{x, y} L(x, y; \lambda, \mu, \nu, \delta^2) \\
 &= \sum_{s: s \in S} \varphi_s(\lambda_s) + \sum_{r: r \in R} \varphi_r(\lambda_r) + \sum_{s: s \in S} \sum_{p: p \in P(s)} \varsigma_{ps}(\lambda_s, \nu_s, \mu_p^L) \\
 &+ \sum_{r: r \in R} \sum_{p: p \in P(r)} \varsigma_{pr}(\lambda_r, \nu_r, \mu_p^H) + \sum_{s: s \in S} \nu_s (C_s^d - \delta_s^2) + \sum_{r: r \in R} \nu_r (C_r^d - \delta_r^2) \\
 &+ \sum_{p: p \in P(s)} \mu_p^L (C_p^{uL} - \delta_{ps}^2) + \sum_{p: p \in P(r)} \mu_p^H (C_p^{uH} - \delta_{pr}^2)
 \end{aligned} \quad (6)$$

From the above Equation (6) we can derive

$$\varphi_s(\lambda_s) = \max_{y_s} U_s(y_s) - \lambda_s y_s \quad (7)$$

$$\varphi_r(\lambda_r) = \max_{y_r} U_r(y_r) - \lambda_r y_r \quad (8)$$

$$\varsigma_{ps}(\lambda_s, \nu_s, \mu_p^L) = \max_{x_{ps}} x_{ps} (\lambda_s - \nu_s - \mu_p^L) \quad (9)$$

$$\varsigma_{pr}(\lambda_r, \nu_r, \mu_p^H) = \max_{x_{pr}} x_{pr} (\lambda_r - \nu_r - \mu_p^H) \quad (10)$$

Then the optimal resource allocation for tasks with low requirement on latency and high requirement on processing capacity can be expressed as

$$y_s^*(\lambda_s) = \operatorname{argmax} U_s(y_s) - \lambda_s y_s \quad (11)$$

$$x_{ps}^*(\nu_s, \mu_p) = \operatorname{argmax} x_{ps} (\lambda_s) (\lambda_s - \nu_s - \mu_p^L) \quad (12)$$

where $\sum_{p: p \in P(s)} x_{ps}(\lambda_s) = y_s^*(\lambda_s)$.

The optimal resource allocation for tasks with high requirement on latency and low requirement on processing capacity can be expressed as

$$y_r^*(\lambda_r) = \operatorname{argmax} U_r(y_r) - \lambda_r y_r \quad (13)$$

$$x_{pr}^*(\nu_r, \mu_p) = \operatorname{argmax} x_{pr}(\lambda_r) (\lambda_r - \nu_r - \mu_p^H) \quad (14)$$

where $\sum_{p: p \in P(r)} x_{pr}(\lambda_r) = y_r^*(\lambda_r)$.

From the perspective of resource customer, customers s and r are trying to maximize their own utilities, which are determined by the allocation bandwidth resource y_s, y_r they obtain. The customers s and r must pay for bandwidth resource they use, $\lambda_s y_s, \lambda_r y_r$ are the cost that customers s and r are willing to pay, respectively. ν_s, ν_r are the prices charged by the download links of customers s and r , so $x_{ps} \nu_s, x_{pr} \nu_r$ are the costs charged by the download links of customers s and r . From the perspective of resource provider, each provider p is trying to maximize its revenue. μ_p^L, μ_p^H are the prices per unit of bandwidth charged by the upload link of provider p for the two types of different tasks, so $x_{ps} \mu_p^L, x_{pr} \mu_p^H$ are the costs charged by the upload link of provider p for the two types of tasks. Thus, the dual problem of resource allocation model is

$$\begin{aligned}
 \min \quad & D(\lambda, \nu, \mu) \\
 \text{over} \quad & \lambda_s \geq 0, \lambda_r \geq 0, \nu_s \geq 0, \nu_r \geq 0, \mu_p^L \geq 0, \mu_p^H \geq 0, s \in S, r \in R, p \in P
 \end{aligned} \quad (15)$$

The objective of the dual problem is to minimize the total cost of transmission for all nodes while ensuring a certain level of satisfaction for customers.

Theorem 1. *The utility function is concave with respect to the variables for both types of tasks, so the optimal resource allocation for each customer and the optimal objective value of the original problem can be obtained, and the optimal objective value of the original problem equals to the optimal objective value of the dual problem, i.e., $U^* = D^*$, but the optimal resource allocation provided by each provider, i.e., x_{ps}^*, x_{pr}^* , is not unique.*

Proof of Theorem 1. From the convex optimization theory [49], since the objective function $\sum_{s:s \in S} U_s(y_s) + \sum_{r:r \in R} U_r(y_r)$ for both types of tasks is concave with respect to its variables and the set of constraints in the resource allocation model is convex, thus the resource allocation model is a convex optimization problem. Then the optimal resource allocation can be obtained by applying the convex optimization approach. The optimal total resource allocation for each customer, i.e., y_s, y_r , is unique. But the optimal resource provided by each provider, i.e., x_{ps}^*, x_{pr}^* may not be unique. The result is obtained. \square

Theorem 2. *When the optimum of the resource allocation model is achieved, if two providers simultaneously provide resource to a customer, then the prices charged by these two providers are equal. This result also holds when considering resource customers for other different types of tasks.*

Proof of Theorem 2. 1. If customers receive tasks with low requirement on latency and high requirement on processing capacity, in the case where the optimum of the resource allocation model is achieved, we can take advantage of the Karush-Kuhn-Tucker (KKT) conditions to obtain that

$$\begin{aligned} \frac{\omega_s}{y_s^*+1} - \lambda_s^* &= 0, y_s^* > 0, \forall s : s \in S \\ \lambda_s^* - v_s^* - \mu_p^{L*} &= 0, x_{ps}^* > 0, \forall s : s \in S, \forall p : p \in P \end{aligned}$$

The above equation is a necessary condition for the existence of an optimal solution to the resource allocation problem. For two providers that offer resources to the same customer, e.g., $p_1, p_2 \in P(s)$, the optimal prices charged by these two providers are equal, i.e.,

$$\mu_{p_1}^{L*} = \mu_{p_2}^{L*} = \lambda_s^* - v_s^*$$

2. If the customers receive tasks with high requirement on latency and low requirement on processing capacity, in the case where the optimum of the resource allocation model is achieved, we can take advantage of the KKT conditions to obtain that

$$\begin{aligned} \frac{M_r}{(y_r^*)^2} - \lambda_r^* &= 0, y_r^* > 0, \forall r : r \in R \\ \lambda_r^* - v_r^* - \mu_p^{H*} &= 0, x_{pr}^* > 0, \forall r : r \in R, \forall p : p \in P \end{aligned}$$

The above equation is a necessary condition for the existence of an optimal solution to the resource allocation problem. For two providers that provide resources to the same customer, e.g., $p_1, p_2 \in P(r)$, the optimal prices charged by these two providers are equal, i.e.,

$$\mu_{p_1}^{H*} = \mu_{p_2}^{H*} = \lambda_r^* - v_r^*$$

The result is obtained. \square

4. Optimal Resource Allocation for Smart Homes

To maximize the aggregated utility of the resource allocation model in the smart home environment, we need to maximize the aggregated utility of the model for processing the tasks with high requirement on latency and low requirement on processing capacity and the aggregated utility of the model for processing the tasks with low requirement on latency and high requirement on processing capacity, respectively. The upload link of resource provider is often regarded as scarce resource and the

corresponding constraints in the model are always active constraints. In this section, we assume that only the constraints of providers when uploading resources are considered in the process of analyzing optimal resource allocation. However, it is also critical that the amount of bandwidth resource received does not exceed the limit of the download link. In the Lagrangian function, δ^2 refers to the remaining bandwidth resource of the participant in the resource allocation. Obviously, according to the KKT conditions when $\delta^2 = 0$, the constraint on the bandwidth resource of participant becomes active. When $\delta^2 > 0$, the constraint on the bandwidth resource of participant becomes non-active. In the following analysis, we assume that $\delta^2 = 0$, which means the remaining resources of provider and the spare capacity of download link are gone. Otherwise, the non-active constraints can be omitted and only active constraints are considered [50].

The utility functions are substituted into the resource allocation model for analyzing respectively. The model for processing the tasks with low requirement on latency and high requirement on processing capacity is

$$\begin{aligned}
 & \max \sum_{s:s \in S} \omega_s \log(y_s + 1) \\
 \text{s.t.} & \sum_{p:p \in P(s)} x_{ps} = y_s, \forall s \in S \\
 & \sum_{p:p \in P(s)} x_{ps} \leq C_s^d, \forall s \in S \\
 & \sum_{s:s \in S(p)} x_{ps} \leq C_p^{uL}, \forall p \in P \\
 & x_{ps} \geq 0, s \in S, r \in R, p \in P
 \end{aligned} \tag{16}$$

The model for processing the tasks with high requirement on latency and low requirement on processing capacity is

$$\begin{aligned}
 & \max \sum_{r:r \in R} (T_r - \frac{M_r}{y_r}) \times \left[\frac{\text{sgn}(y_r - B_r) + 1}{2} \right] \\
 \text{s.t.} & \sum_{p:p \in P(r)} x_{pr} = y_r, \forall r \in R \\
 & \sum_{p:p \in P(r)} x_{pr} \leq C_r^d, \forall r \in R \\
 & \sum_{r:r \in R(p)} x_{pr} \leq C_p^{uH}, \forall p \in P \\
 & x_{pr} \geq 0, s \in S, r \in R, p \in P
 \end{aligned} \tag{17}$$

4.1. The Optimal Resource Allocation for Customer s

We first don't consider the download bandwidth constraint (i.e., the first inequality constraint) in the resource allocation model (16), and obtain the following Lagrangian function:

$$\begin{aligned}
 L_1(x, y; \lambda, \mu) &= \sum_{s:s \in S} \omega_s \log(y_s + 1) + \sum_{s:s \in S} \lambda_s \left(\sum_{p:p \in P(s)} x_{ps} - y_s \right) \\
 &+ \sum_{p:p \in P(s)} \mu_p^L \left(C_p^{uL} - \sum_{s:s \in S(p)} x_{ps} \right)
 \end{aligned} \tag{18}$$

The Lagrangian function can be rewritten as

$$L_1(x, y; \lambda, \mu) = \sum_{s:s \in S} [\omega_s \log(y_s + 1) - \lambda_s y_s] + \sum_{s:s \in S} \sum_{p:p \in P(s)} x_{ps} (\lambda_s - \mu_p^L) + \sum_{p:p \in P(s)} \mu_p^L C_p^{uL} \tag{19}$$

According to Equation (7), there is an optimal allocation

$$y_s^* = U_s^{-1}(\lambda_s) = \frac{\omega_s}{\lambda_s^*} - 1 \tag{20}$$

Then, substituting (20) into (19), we obtain

$$\bar{L}_1(x; \lambda, \mu) = \sum_{s:s \in S} \left[\omega_s \log\left(\frac{\omega_s}{\lambda_s^*}\right) - \omega_s + \lambda_s^* + \lambda_s^* \sum_{p:p \in P(s)} x_{ps} \right] - \sum_{s:s \in S} \sum_{p:p \in P(s)} x_{ps} \mu_p^L + \sum_{p:p \in P(s)} \mu_p^L C_p^{uL} \tag{21}$$

Let $\partial \bar{L}_1(x; \lambda, \mu) / \partial \lambda_s = 0$, then we obtain the optimal price paid by customer s

$$-\frac{\omega_s}{\lambda_s} + 1 + \sum_{p:p \in P(s)} x_{ps} = 0$$

$$\lambda_s^* = \frac{\omega_s}{1 + \sum_{p:p \in P(s)} x_{ps}} \tag{22}$$

Substitute the obtained result (22) into function (19) again

$$\begin{aligned} \bar{L}_1(x; \nu) = & \sum_{s:s \in S} \left[\omega_s \log\left(1 + \sum_{p:p \in P(s)} x_{ps}\right) - \omega_s + \left(\frac{\omega_s}{1 + \sum_{p:p \in P(s)} x_{ps}} + 1\right) \right. \\ & \left. + \left(\frac{\omega_s}{1 + \sum_{p:p \in P(s)} x_{ps}} + 1\right) \times \sum_{p:p \in P(s)} x_{ps} \right] - \sum_{s:s \in S} \sum_{p:p \in P(s)} x_{ps} \mu_p^L \\ & + \sum_{p:p \in P(s)} \mu_p^L C_p^{uL} \end{aligned} \tag{23}$$

Let $\partial \bar{L}_1(x; \nu) / \partial x_{ps} = 0$, then we obtain the optimal price charged by provider p

$$\frac{\omega_s}{1 + \sum_{p:p \in P(s)} x_{ps}} - \mu_{ps} = 0$$

$$\mu_p^{L*} = \frac{\omega_s}{1 + \sum_{p:p \in P(s)} x_{ps}} \tag{24}$$

Therefore, when the download bandwidth constraint of each customer is not considered, we obtain $\mu_{p_1}^{L*} = \mu_{p_2}^{L*} = \lambda_s^*$, $p_1, p_2 \in P(s)$, $s : s \in S$. In this case, the price paid by the resource customer s for per unit bandwidth is equal to the price charged by the provider p for per unit bandwidth.

We assume that the entire network is divided into τ non-connected regions, where each region is composed of a sub-set of all nodes. According to Theorem 2, the optimal prices charged by resource providers in each region are equal, and they are all equal to the price paid by any customer s , $\mu_p^{L*} = \mu_\tau$, $\lambda_s^* = \lambda_\tau$, $\mu_\tau = \lambda_\tau$, where $s : s \in S_\tau$. S_τ refers to the collection of all resource consumers in the region. Particularly, when $\tau = 1$, then $P_\tau = P$, $\mu_\tau = \mu$, $S_\tau = S$.

Therefore, the transformed Lagrangian function can be expressed as

$$\begin{aligned} \bar{L}_1(x; \nu) = & \sum_{\tau} \left[\sum_{s:s \in S_\tau} \omega_s \log\left(\sum_{p:p \in P_\tau(s)} x_{ps} + 1\right) + \sum_{p:p \in P_\tau} \mu_{ps} \left(C_p^{uL} - \sum_{s:s \in S_\tau(p)} x_{ps}\right) \right] \\ = & \sum_{\tau} \left[\sum_{s:s \in S_\tau} \left(\omega_s \log \frac{\omega_s}{\nu_\tau} - \omega_s + \mu_\tau\right) + \sum_{p:p \in P_\tau} \mu_\tau C_p^{uL} \right] \end{aligned} \tag{25}$$

where $\mu_p^{L*} = \mu_\tau, \forall p \in P_\tau$.

Let $d\bar{L}_1(\mu_\tau)/d\mu_\tau = 0$, then

$$\sum_{\tau} \left\{ \sum_{s:s \in S_{\tau}} \left[\omega_s \times \frac{1}{\frac{\omega_s}{\mu_{\tau}}} \times \frac{\omega_s}{-(\mu_{\tau}^2)} + 1 \right] + \sum_{p:p \in P_{\tau}} C_p^{uL} \right\} = 0$$

$$\mu_{\tau} = \frac{\sum_{s:s \in S_{\tau}} \omega_s}{|S_{\tau}| + \sum_{p:p \in P_{\tau}} C_p^{uL}} \tag{26}$$

Substituting the above result (26) into (24)

$$y_s^* = \frac{\omega_s(|S_{\tau}| + \sum_{p:p \in P_{\tau}} C_p^{uL})}{\sum_{s:s \in S_{\tau}} \omega_s} - 1 \tag{27}$$

$|S_{\tau}|$ indicates the number of customers who need resources in sub-region τ .

Recall that the optimal resource allocation of customer s is also subject to its download bandwidth capacity C_s^d , thus the optimal resource allocation y_s^* has the following value

$$y_s^* = \min \left\{ C_s^d, \frac{\omega_s(|S_{\tau}| + \sum_{p:p \in P_{\tau}} C_p^{uL})}{\sum_{s:s \in S_{\tau}} \omega_s} - 1 \right\} \tag{28}$$

4.2. The Optimal Resource Allocation for Customer r

Following the analysis method for resource allocation (16), we obtain the Lagrangian function for resource allocation (17) as follows.

$$L_2(x, y; \lambda, \mu) = \sum_{r:r \in R} \left\{ \left(T_r - \frac{M_r}{y_r} \right) \times \left[\frac{\text{sgn}(y_r - B_r) + 1}{2} \right] \right\} + \sum_{r:r \in R} \lambda_r \left(\sum_{p:p \in P(r)} x_{pr} - y_r \right)$$

$$+ \sum_{p:p \in P} \mu_p^H \left(C_p^{uH} - \sum_{r:r \in R(p)} x_{pr} \right) \tag{29}$$

The Lagrangian function can be rewritten as

$$L_2(x, y; \lambda, \mu) = \sum_{r:r \in R} \left\{ \left(T_r - \frac{M_r}{y_r} \right) \times \left[\frac{\text{sgn}(y_r - B_r) + 1}{2} \right] - \lambda_r y_r \right\}$$

$$+ \sum_{r:r \in R} \sum_{p:p \in P(r)} x_{pr} (\lambda_r - \mu_p^H) + \sum_{p:p \in P} \mu_p^H C_p^{uH} \tag{30}$$

According to Equation (8), there is an optimal allocation

$$y_r^* = U_r^{-1}(\lambda_r) = \left(\frac{M_r}{\lambda_r^*} \right)^{\frac{1}{2}} \tag{31}$$

Then

$$\bar{L}_2(x; \lambda, \mu) = \sum_{r:r \in R} \left(T_r - 2(\lambda_r M_r)^{\frac{1}{2}} \right) + \sum_{r:r \in R} \sum_{p:p \in P(r)} x_{pr} (\lambda_r - \mu_p^H) + \sum_{p:p \in P} \mu_p^H C_p^{uH} \tag{32}$$

Let $\partial \bar{L}_2(x; \lambda, \mu) / \partial \lambda_r = 0$, then we obtain the optimal price paid by user r .

$$-\frac{M_r}{(\lambda_r M_r)^{\frac{1}{2}}} + \sum_{p:p \in P(r)} x_{pr} = 0$$

$$\lambda_r^* = \frac{M_r}{\left(\sum_{p:p \in P(r)} x_{pr}\right)^2} \quad (33)$$

Substitute the obtained result into function (31) again

$$\begin{aligned} \bar{L}_2(x; \mu) = & \sum_{r:r \in R} \left[T_r - 2 \left(\frac{M_r}{\left(\sum_{p:p \in P(r)} x_{pr}\right)^2} M_r \right)^{\frac{1}{2}} \right] \\ & + \sum_{r:r \in R} \sum_{p:p \in P(r)} x_{pr} \left[\frac{M_r}{\left(\sum_{p:p \in P(r)} x_{pr}\right)^2} - \mu_p^H \right] + \sum_{p:p \in P} \mu_p^H C_p^{uH} \end{aligned} \quad (34)$$

$$\bar{L}_2(x; \mu) = \sum_{r:r \in R} \left(T_r - 2 \frac{M_r}{\sum_{p:p \in P(r)} x_{pr}} \right) + \sum_{r:r \in R} \left(\frac{M_r}{\sum_{p:p \in P(r)} x_{pr}} - \sum_{p:p \in P(r)} x_{pr} \mu_p^H \right) + \sum_{p:p \in P} \mu_p^H C_p^{uH} \quad (35)$$

Let $\partial \bar{L}_2(x; \mu) / \partial x_{pr} = 0$, then we obtain the optimal price charged by edge node p

$$\mu_p^{H*} = \frac{M_r}{\left(\sum_{p:p \in P(r)} x_{pr}\right)^2} \quad (36)$$

On the basis of the above discussion, it can be concluded that when the optimal resource allocation is achieved, the following condition holds: $\mu_{p_1}^{H*} = \mu_{p_2}^{H*} = \lambda_r^*$, $p_1, p_2 \in P(s)$, $r : r \in R$.

We assume that the entire network is divided into γ non-connected regions, where each region corresponds to a part of nodes. According to Theorem 2, the optimal prices charged by resource providers in each region are equal, and they are all equal to the price paid by any customer r , $\mu_p^{H*} = \mu_\gamma$, $\lambda_r^* = \lambda_\gamma$, $\mu_\gamma = \lambda_\gamma$, where $r : r \in R_\gamma$, $p : p \in P_\gamma$. R_γ refers to the collection of all resource consumers r in the region. Particularly, when $\gamma = 1$, then $R_\gamma = R$, $\mu_\gamma = \mu_p^H$, $P_\gamma = P$.

Therefore, the transformed Lagrangian function can be expressed as

$$\begin{aligned} \bar{L}_2(x; \mu) = & \sum_{\gamma} \left[\sum_{r:r \in R_\gamma} \left(T_r - \frac{M_r}{\sum_{p:p \in P(r)} x_{pr}} \right) - \sum_{r:r \in R(r)} \sum_{p:p \in P(r)} x_{pr} \mu_\gamma + \sum_{p:p \in P} \mu_\gamma C_p^{uH} \right] \\ = & \sum_{\gamma} \left\{ \sum_{r:r \in R_\gamma} \left[T_r - \frac{M_r}{\left(\frac{M_r}{\mu_\gamma}\right)^{\frac{1}{2}}} - \mu_\gamma \times \left(\frac{M_r}{\mu_\gamma}\right)^{\frac{1}{2}} \right] + \sum_{p:p \in P} \mu_\gamma C_p^{uH} \right\} \\ = & \sum_{\gamma} \left\{ \sum_{r:r \in R_\gamma} \left[T_r - 2 \left(M_r \mu_\gamma \right)^{\frac{1}{2}} \right] + \sum_{p:p \in P} \mu_\gamma C_p^{uH} \right\} \end{aligned} \quad (37)$$

where $\mu_p^{H*} = \mu_\gamma$, $r : r \in S_\gamma$.

Let $d\bar{L}_2(x; \mu_{pr}) / d\mu_\gamma = 0$, then

$$\mu_\gamma = \left[\frac{\sum_{r:r \in R_\gamma} (M_r)^{\frac{1}{2}}}{\sum_{p:p \in P} C_p^{uH}} \right]^2 \quad (38)$$

Comparing (33) and (36), and substituting μ_γ into the above Equation (33)

$$y_r^* = U_r^{-1}(\lambda_r) = \frac{(M_r)^{\frac{1}{2}} \sum_{p:p \in P} C_p^{uH}}{\sum_{r:r \in R_\gamma} (M_r)^{\frac{1}{2}}} \quad (39)$$

Recall that the optimal resource allocation of customer r is also subject to its download bandwidth capacity C_r^d , thus the optimal resource allocation y_r^* has the following value

$$y_r^* = \min \left\{ C_r^d, \frac{(M_r)^{\frac{1}{2}} \sum_{p:p \in P} C_p^{uH}}{\sum_{r:r \in R_y} (M_r)^{\frac{1}{2}}} \right\} \quad (40)$$

5. Resource Allocation Algorithm for Smart Homes

5.1. Algorithm Introduction

In order to obtain the optimal resource allocation of edge computing without using cloud center in a smart home environment, this section will introduce a distributed algorithm that only depends on local information, which is in line with smart home. Since the objective of the resource allocation model is concave but not strictly concave, which results in the optimal resource allocation is usually not unique, so that the proposed algorithm may be oscillation. Thus, we introduce and apply the low-pass filtering method to eliminate the possible oscillation and improve the convergence speed [51]. We regard $\widetilde{x}_{ps}(t), \widetilde{x}_{pr}(t)$ as the optimal estimation of the $x_{ps}(t), x_{pr}(t)$ and summarize the detailed algorithm steps as follows

1. Each provider p updates the resources $x_{ps}(t), x_{pr}(t)$ for its customers s and r . The update rule for customer s is

$$x_{ps}(t+1) = \left\{ (1 - \theta_{ps})x_{ps}(t) + \theta_{ps}\widetilde{x}_{ps}(t) + \theta_{ps}\kappa_{ps}x_{ps}(t) [\lambda_s(t) - v_s(t) - \mu_p^L(t)] \right\}_{x_{ps}(t)}^+ \quad (41)$$

$$\widetilde{x}_{ps}(t+1) = (1 - \theta_{ps})\widetilde{x}_{ps}(t) + \theta_{ps}x_{ps}(t) \quad (42)$$

$$\lambda_s(t) = U'_s(y_s(t)) \quad (43)$$

$$y_s(t) = \sum_{p:p \in P(s)} x_{ps}(t) \quad (44)$$

The update rule for customer r is

$$x_{pr}(t+1) = \left\{ (1 - \theta_{pr})x_{pr}(t) + \theta_{pr}\widetilde{x}_{pr}(t) + \theta_{pr}\kappa_{pr}x_{pr}(t) [\lambda_r(t) - v_r(t) - \mu_p^H(t)] \right\}_{x_{pr}(t)}^+ \quad (45)$$

$$\widetilde{x}_{pr}(t+1) = (1 - \theta_{pr})\widetilde{x}_{pr}(t) + \theta_{pr}x_{pr}(t) \quad (46)$$

$$\lambda_r(t) = U'_r(y_r(t)) \quad (47)$$

$$y_r(t) = \sum_{p:p \in P(r)} x_{pr}(t) \quad (48)$$

where $\kappa_{ps}, \kappa_{pr} > 0$ are the small step sizes, $\theta_{ps}, \theta_{pr} > 0$ are the parameters for low-pass filtering method. $a = (b)_c^+$ means $c > 0, a = b; c = 0, a = \max\{0, b\}$.

2. Resource customers r, s update their prices $v_s(t), v_r(t)$ as the following law

$$v_s(t+1) = \left[v_s(t) + \alpha_s \frac{y_s(t) - C_s^d}{C_s^d} \right]_{v_s(t)}^+ \quad (49)$$

$$y_s(t) = \sum_{p:p \in P(s)} x_{ps}(t) \quad (50)$$

$$v_r(t+1) = \left[v_r(t) + \alpha_r \frac{y_r(t) - C_r^d}{C_r^d} \right]_{v_r(t)}^+ \quad (51)$$

$$y_r(t) = \sum_{p:p \in P(r)} x_{pr}(t) \quad (52)$$

where $\alpha_s, \alpha_r > 0$ are the small step sizes.

3. Each resource provider p (edge node) updates its prices $\mu_p^L(t), \mu_p^H(t)$ by using the following method.

$$\mu_p^L(t+1) = \left[\mu_p^L(t) + \beta_s \frac{z_p^L - C_p^{uL}}{C_p^{uL}} \right]_{\mu_p^L(t)}^+ \quad (53)$$

$$z_p^L(t) = \sum_{s:s \in S(p)} x_{ps}(t) \quad (54)$$

$$\mu_p^H(t+1) = \left[\mu_p^H(t) + \beta_r \frac{z_p^H - C_p^{uH}}{C_p^{uH}} \right]_{\mu_p^H(t)}^+ \quad (55)$$

$$z_p^H(t) = \sum_{r:r \in R(p)} x_{pr}(t) \quad (56)$$

where $\beta_s, \beta_r > 0$ are the small step sizes.

5.2. Basic Steps

According to the above-mentioned low-pass filtering approach, the augmented variables $\widetilde{x}_{ps}(t), \widetilde{x}_{pr}(t)$ can only eliminate the possible oscillation of the algorithm and improve the convergence speed, but it will not change the final optimum of the algorithm. At the optimal resource allocation, we obtain that $\widetilde{x}_{ps}^*(t) = \widetilde{x}_{ps}^*(t), \widetilde{x}_{pr}^*(t) = \widetilde{x}_{pr}^*(t)$.

The iterative step size selected in the algorithm has a great influence on the convergence speed. The step size selected in the algorithm should be small enough to ensure convergence, but not too small, which causes the convergence speed to become too slow, so it is necessary to choose a suitable iteration step size. The algorithm can be iterated to the optimal resource allocation within reasonable convergence times.

The basic steps of the algorithm iteration procedure are described as following.

STEP 1: Initialize the variables and parameters.

We need to initialize the iteration step sizes $\kappa_{ps}, \kappa_{pr}, \alpha_s, \alpha_r, \beta_s, \beta_r$ and initialize the bandwidth resource allocation $x_{ps}(t), x_{pr}(t)$ from provider p to customers s and r at time t .

STEP 2: Calculate the prices paid by customers s and r at time t .

Customers s and r calculate the offered bandwidth resources $y_s(t), y_r(t)$ that they receive based on the amount of bandwidth resources provided by providers at time t , and then calculate the prices they should pay for provider p , respectively.

STEP 3: Calculate the prices charged by the upload and download links of each node at time t .

Provider p updates the amount of resources it provides for customers s and r at time t , $z_p^L(t)$ and $z_p^H(t)$. At the same time, provider p calculates the price $\mu_p(t+1)$ charged by the uploading link. Customers s and r update the prices $v_s(t+1), v_r(t+1)$ charged by their downloading links.

STEP 4: Update the bandwidth allocation of provider p at time t .

Provider p updates its resource allocation $x_{ps}(t+1), x_{pr}(t+1)$ for customers s and r at time t .

STEP 5: Set the stop condition.

The optimization problem considered in this paper is a convex optimization problem, which means that the optimal solution exists and is also the global optimal solution [52]. Therefore, when the algorithm reaches an equilibrium, $x_{ps}(t+1) = x_{ps}(t), x_{pr}(t+1) = x_{pr}(t)$, the iteration of the algorithm can be suspended and the optimal bandwidth resource allocation is obtained.

In each iteration, the two types of resource customers s and r calculate the prices they should pay to providers and the prices charged by downloading links according to the amount of resources

provided by providers. Each provider p calculates the price charged by its upload link and updates the resource allocation for its customers. Therefore, we can repeat the iterative process until the optimum is finally reached.

6. Simulation and Numerical Examples in a Smart Home Environment

In this section, we will build a small-scale smart home environment in an edge computing resource allocation scenario without using cloud center and study the performance of resource allocation schemes. As shown in Figure 3, assume that there are 9 nodes (smart home applications) in the smart home environment, where 3 nodes on the left which serve as resource providers provide bandwidth resource for the other 6 nodes on the right which serve as customers. The resource providers and customers form a resource allocation market and are free to exchange resources in the marketplace.

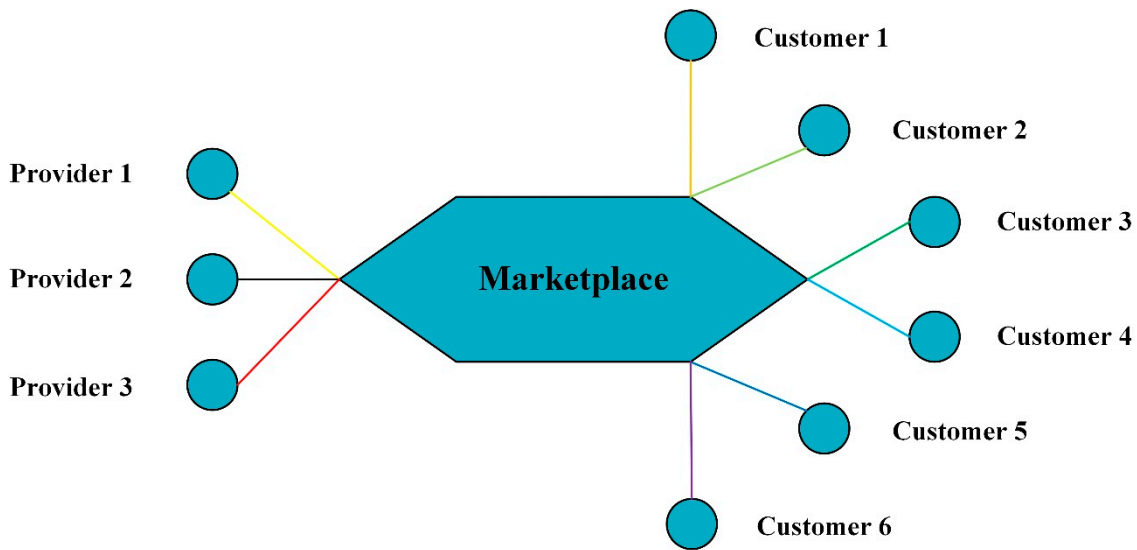


Figure 3. The simple topological structure of marketplace without using cloud center.

6.1. Processing the Tasks for Customer s

Assume that the utility functions of 6 customers are: $U_1(y_1) = 25 \log(y_1 + 1)$, $U_2(y_2) = 20 \log(y_2 + 1)$, $U_3(y_3) = 15 \log(y_3 + 1)$, $U_4(y_4) = 10 \log(y_4 + 1)$, $U_5(y_5) = 5 \log(y_5 + 1)$, $U_6(y_6) = 4 \log(y_6 + 1)$. Assume that the upstream capacity of providers for these customers is $C_p^{uL} = (C_1^{uL}, C_2^{uL}, C_3^{uL}) = (15, 12, 10) \text{ Mb/s}$ and the downstream capacity of customers is $C_s^d = (C_1^d, C_2^d, C_3^d, C_4^d, C_5^d, C_6^d) = (17, 14, 10, 8, 6, 4) \text{ Mb/s}$ and step sizes are $\kappa_{ps} = 0.5$, $\alpha_s = 0.2$, $\beta_s = 0.2$, $\theta_{ps} = 0.2$. The initial rates of the resource allocation are set to be $x_{ps} = 1 \text{ Mb/s}$.

The prices paid by the 6 resource customers are shown in Figure 4a, we find that the prices are finally driven to be an equilibrium, that is, $\lambda_1^* = \lambda_2^* = \lambda_3^* = \lambda_4^* = \lambda_5^* = \lambda_6^*$. The prices charged by the 3 providers for upload links are $\mu_1^L, \mu_2^L, \mu_3^L$, and at the optimum $\mu_1^{L*} = \mu_2^{L*} = \mu_3^{L*}$. After a certain number of iterations, the prices charged by providers for upload links and the prices paid by customers are equal, namely $\lambda_s^* = \mu_p^{L*}, p \in P(s)$.

Figure 4b shows the utility of resource customers $U_1, U_2, U_3, U_4, U_5, U_6$, as well as the aggregated utility $TU = U_1 + U_2 + U_3 + U_4 + U_5 + U_6$. Obviously, the algorithm can drive the customer's utility to reach the optimal value within a limited number of iterations.

The optimal resource allocation for each customer is also shown in Figure 5, where x_{ps} represents the amount of bandwidth resource provided by provider p to customer s , y_s is the amount of bandwidth resource received by each customer s . For example, in the "Resources for customer 1" of Figure 5, x_{11}, x_{21}, x_{31} respectively represent the amount of resource provided by providers $p = 1, p = 2, p = 3$ to customer $s = 1$ and y_1 represents the total amount of bandwidth resource obtained by the customer

$s = 1$. We can find that the optimal resource allocation can be achieved within a certain number of iterations.

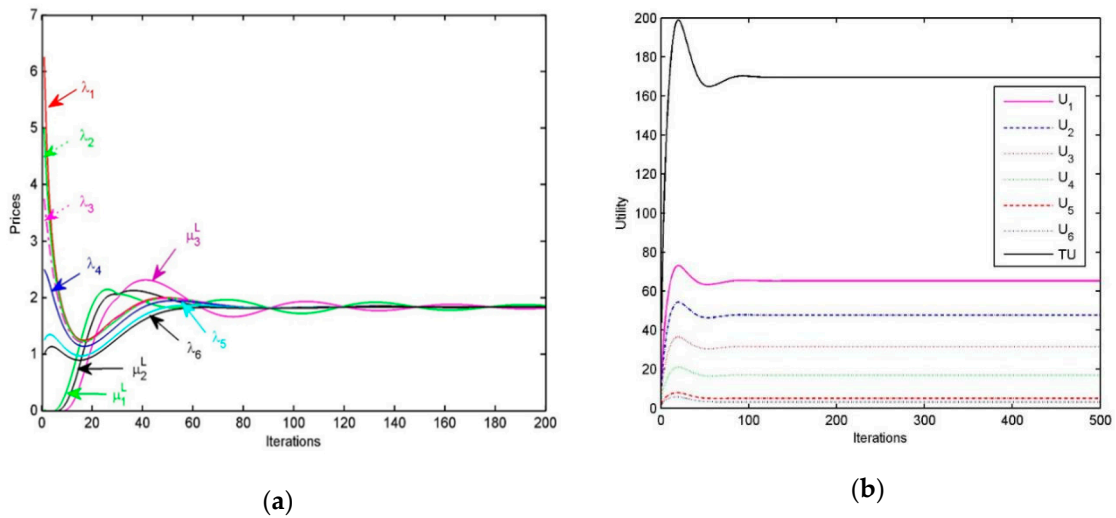


Figure 4. (a) Performance of the resource allocation algorithm for customer s : Prices; (b) Performance of the resource allocation algorithm for customer s : Utility.

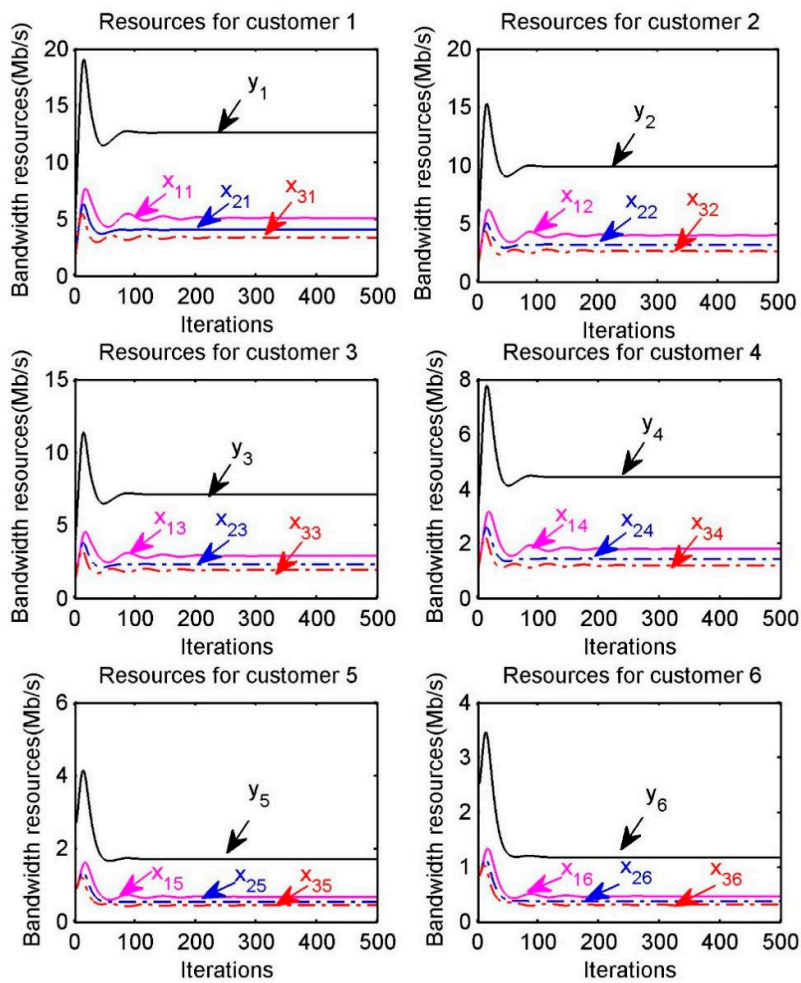


Figure 5. Optimal bandwidth resource allocation for customer s .

Linear Interactive and General Optimizer (LINGO) is a nonlinear programming software which can be used to solve nonlinear programming problems, some linear and nonlinear equations, and has become one of the best choices for solving optimization models with its powerful functions. As shown in Table 2, we list the simulation results of the proposed algorithm and the optimal solution of the optimization problem by applying LINGO. It is not difficult to find that the total amount of optimal bandwidth resource received by each resource customer is unique, but the optimal bandwidth resource obtained from each resource provider is not unique. This can be understood from the fact that a customer can obtain bandwidth resource from multiple providers and a provider can provide resource for multiple customers, which leads to the non-uniqueness of the optimal solution. Indeed, according to the convex optimization theory [52], the objective function is concave with respect to the variables and the constraints are linear, which means that the resource allocation model is a convex optimization problem. However, the objective function is not strict concave with respect to variables x_{ps} , thus the optimal bandwidth allocation from provider to each customer is not unique.

Table 2. The optimal bandwidth allocation for customer s .

Variable	x_{11}^*	x_{21}^*	x_{31}^*	x_{12}^*	x_{22}^*	x_{32}^*	y_1^*	y_2^*
Algorithm	5.116	4.088	3.404	4.015	3.205	2.666	12.60	9.886
LINGO	4.482	3.396	4.729	4.726	3.190	1.969	12.60	9.885
Variable	x_{13}^*	x_{23}^*	x_{33}^*	x_{14}^*	x_{24}^*	x_{34}^*	y_3^*	y_4^*
Algorithm	2.901	2.324	1.940	1.804	1.441	1.199	7.165	4.444
LINGO	3.121	2.178	1.865	1.512	1.621	1.309	7.164	4.442
Variable	x_{15}^*	x_{25}^*	x_{35}^*	x_{16}^*	x_{26}^*	x_{36}^*	y_5^*	y_6^*
Algorithm	0.696	0.559	0.467	0.475	0.382	0.320	1.722	1.177
LINGO	1.157	0.438	0.127	0	0.177	0	1.722	1.177

According to Equation (28), we can observe that optimal bandwidth resource allocation obtained by customer s not only relies on the willingness to pay (WTP) of customer, the number of customers in sub-region and the upload bandwidth capacity C_p^{ul} of provider, but is also subject to the download bandwidth capacity C_s^d of customer. In other words, when the amount of bandwidth resources provided by providers increases, a customer may not be able to pick up the entire amount of resources due to its own limited download bandwidth capacity. Therefore, we can determine whether a customer is able to pick up the entire amount of resources provided by providers through the assumption that the amount of resources provided by the providers increases, i.e., through releasing provider's upload bandwidth capacity. As shown in Figure 6, we assume that the upload bandwidth capacity of providers in diverse settings are $(C_1^{ul}, C_2^{ul}, C_3^{ul}) = (15, 12, 10) Mb/s$, $(C_1^{ul}, C_2^{ul}, C_3^{ul}) = (22.5, 18, 15) Mb/s$, $(C_1^{ul}, C_2^{ul}, C_3^{ul}) = (30, 24, 20) Mb/s$, respectively. The six sub-figures correspond to the resource allocation for the six customers and in each sub-figure the red, blue and purple lines correspond to the resource allocation under different upload bandwidth capacities. Obviously, when the upload bandwidth capacity is released, the amount of resources received by customer is affected by its download bandwidth capacity.

In Figure 7, we investigate the effect of different iterative step sizes on convergence speed. The six sub-figures correspond to the resource allocation for the six customers. The purple, red and blue lines in each sub-figure are regarded as resource allocation with different step sizes. We can generalize the effect of different iteration step sizes on convergence speed by comparing how fastly the lines of different colors finally converge in the sub-figure. For example, in the first sub-figure "Resources for customer 1", the purple, red and blue lines represent the resource allocation in the case of $\kappa_{ps} = \alpha_s = \beta_s = 0.1$, $\kappa_{ps} = \alpha_s = \beta_s = 0.3$, $\kappa_{ps} = \alpha_s = \beta_s = 0.5$, respectively. Obviously, the convergence speed is proportional to step sizes. Thus, it is essential that choosing appropriate step sizes for achieving the optimal resource allocation within reasonable iteration times.

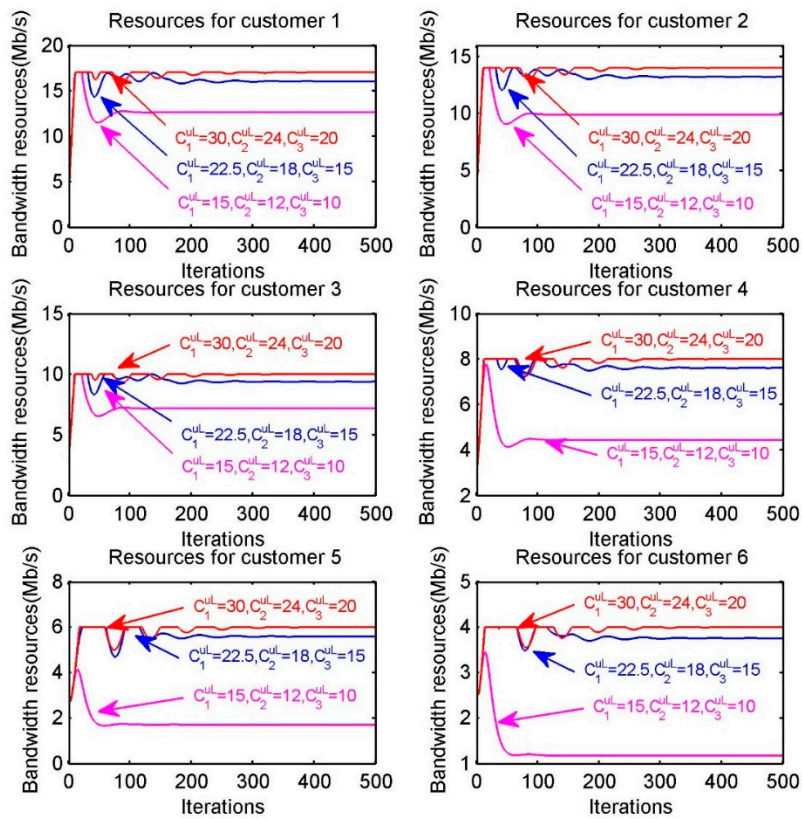


Figure 6. Optimal bandwidth resource allocation of each customer s when increasing the upload capacity of providers.

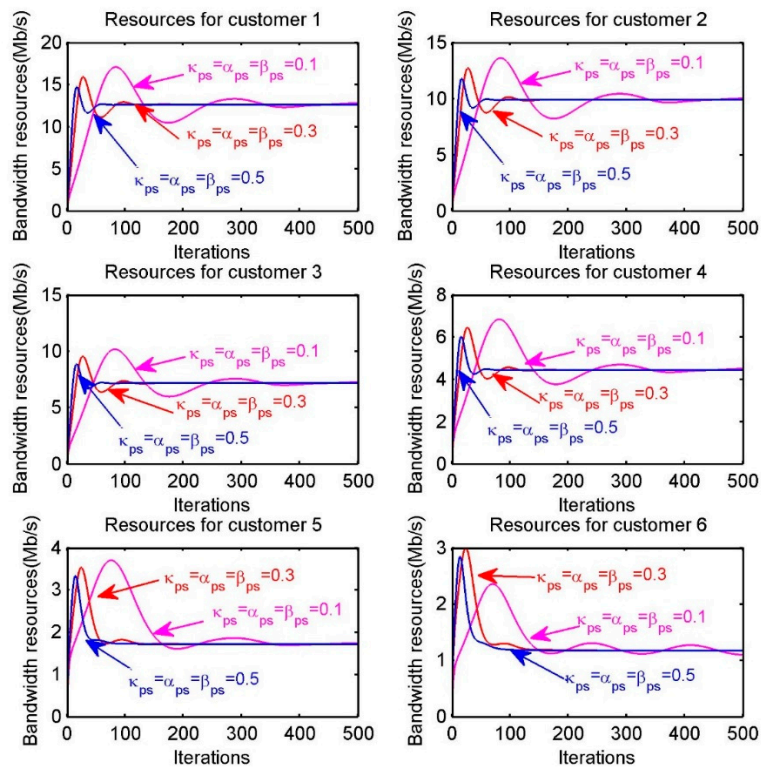


Figure 7. Resources allocation of each customer s when choosing different step sizes.

6.2. Processing the Tasks for Customer r

Assume that the utility functions of customers for the tasks with high requirement on latency and low requirement on processing capacity are: $U_1(y_1) = 100 - \frac{50}{y_1}$, $U_2(y_2) = 100 - \frac{40}{y_2}$, $U_3(y_3) = 100 - \frac{30}{y_3}$, $U_4(y_4) = 50 - \frac{50}{y_4}$, $U_5(y_5) = 50 - \frac{40}{y_5}$, $U_6(y_6) = 50 - \frac{30}{y_6}$. Assume that the upstream capacity of provider is $C_p^{uH} = (C_1^{uH}, C_2^{uH}, C_3^{uH}) = (10, 8, 5) Mb/s$, the downstream capacity of customer r is $C_r^d = (C_1^d, C_2^d, C_3^d, C_4^d, C_5^d, C_6^d) = (20, 18, 15, 12, 10, 5) Mb/s$ and the step sizes $\kappa_{pr} = 0.5$, $\alpha_r = 0.2$, $\beta_r = 0.2$, $\theta_{pr} = 0.2$. The initial rates of the resource allocation are all set to be $x_{pr} = 0.5 Mb/s$.

The prices charged by the three providers for upload links are shown in Figure 8. We find that at the equilibrium they are all equal. The prices paid by the six customers are also shown in this figure. It is obvious that at the equilibrium these prices are all equal, i.e., $\lambda_1^* = \lambda_2^* = \lambda_3^* = \lambda_4^* = \lambda_5^* = \lambda_6^*$. Furthermore, after reasonable iterations, the prices charged by the providers and the prices paid by the resource customers are all equal, namely $\lambda_r^* = \mu_p^{H*}$, $p \in P(r)$.

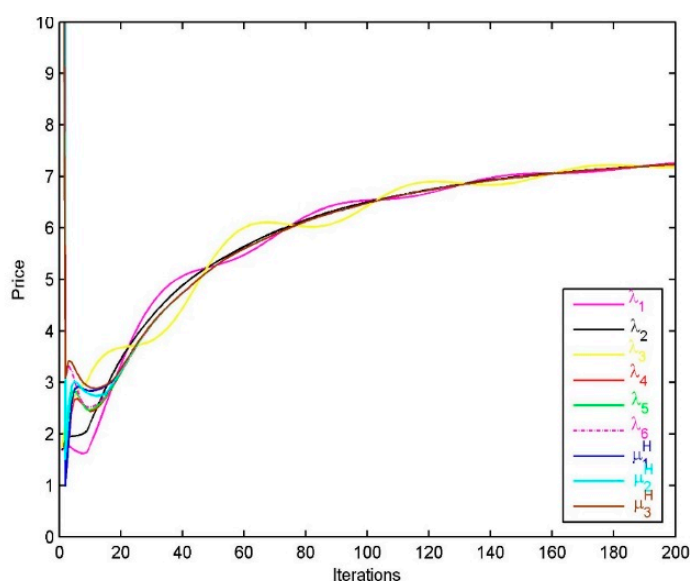


Figure 8. Performance of the resource allocation algorithm for customer r : Prices.

Figure 9 shows the utility of each resource customer $U_1, U_2, U_3, U_4, U_5, U_6$, as well as the aggregated utility $TU = U_1 + U_2 + U_3 + U_4 + U_5 + U_6$. Obviously, the algorithm can drive the customer’s utility to reach the optimal value within a limited number of iterations. The utility of each resource customer r is related to the number of tasks and time limits, from which we find that the longer the time limits are, the higher the utility of resource customer is and the larger the number of tasks for customer are, the lower the utility of customer is.

The optimal resource allocation is also shown in Figure 10, where x_{pr} represents the amount of bandwidth resource provided by provider p to customer r , y_r is the amount of bandwidth resource received by each customer r . Meanwhile, in the “Resource for customer 1” of Figure 10, x_{11}, x_{21}, x_{31} respectively represents the amount of resource provided by the three providers to customer 1 and y_1 represents the total amount of resource obtained by customer 1.

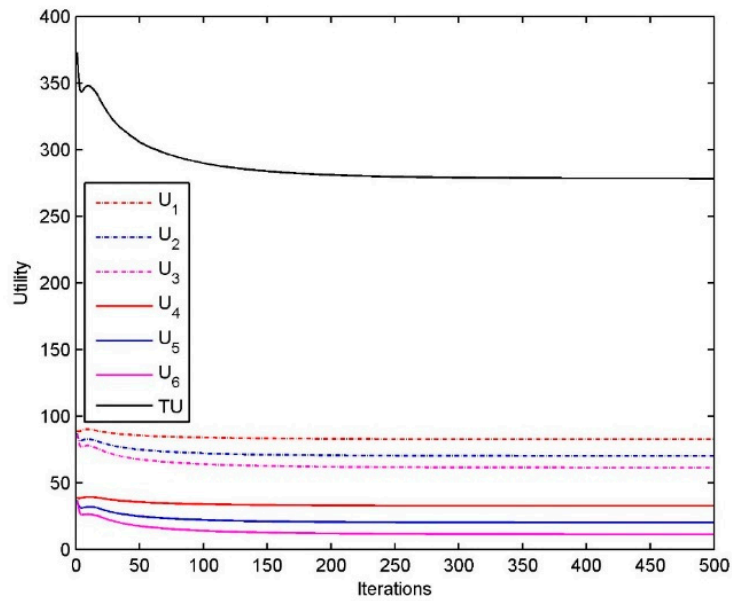


Figure 9. Performance of the resource allocation algorithm for customer r : Utility.

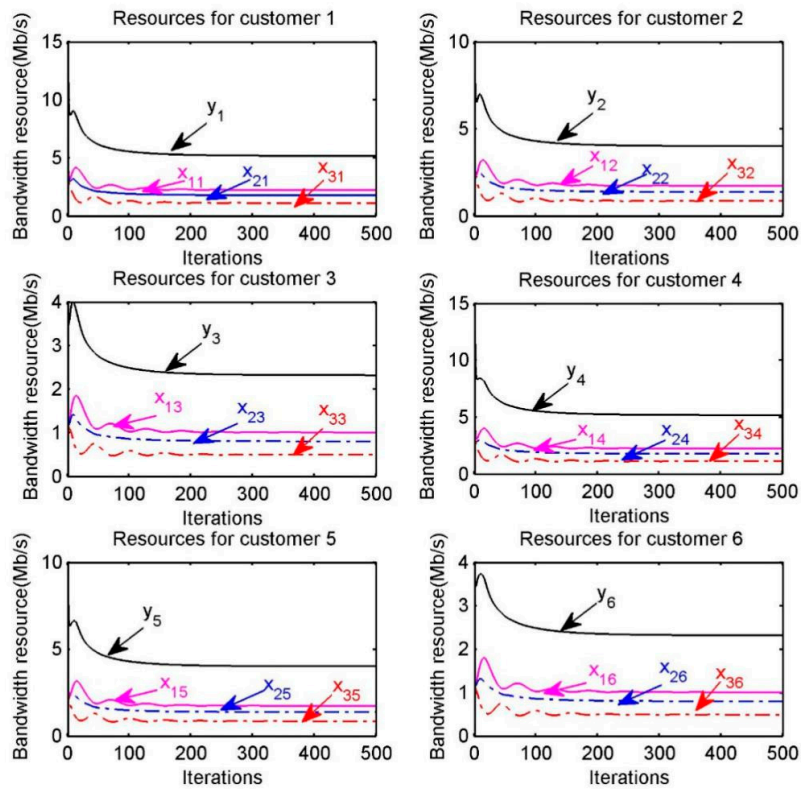


Figure 10. Optimal bandwidth resource allocation for customer r .

In Table 3, we list the optimal resource allocation obtained from the algorithm and the optimal value obtained by using the nonlinear programming software LINGO. It is not difficult to observe that the total amount of optimal bandwidth resources received by each resource customer is unique, which has been justified in Theorem 1. When the constraints with download links of customers are not active in the simple case, the optimal bandwidth allocation for each customer can also be derived from the algorithm, which is also equivalent to the optimal values from LINGO.

Table 3. The optimal allocation of the bandwidth resources for customer r .

Variable	x_{11}^*	x_{21}^*	x_{31}^*	x_{12}^*	x_{22}^*	x_{32}^*	y_1^*	y_2^*
Algorithm	2.244	1.800	1.133	1.744	1.395	0.871	5.177	4.010
LINGO	3.650	0.811	0.714	1.480	1.511	1.019	5.175	4.010
Variable	x_{13}^*	x_{23}^*	x_{33}^*	x_{14}^*	x_{24}^*	x_{34}^*	y_3^*	y_4^*
Algorithm	1.010	0.806	0.500	2.245	1.800	1.134	2.316	5.179
LINGO	0.904	0.967	0.444	1.858	1.921	1.397	2.315	5.176
Variable	x_{15}^*	x_{25}^*	x_{35}^*	x_{16}^*	x_{26}^*	x_{36}^*	y_5^*	y_6^*
Algorithm	1.746	1.395	0.869	1.011	0.806	0.498	4.010	2.315
LINGO	1.309	1.372	1.328	0.800	1.418	0.098	4.009	2.316

According to Equation (40), we can observe that optimal bandwidth resource allocation obtained by customer r not only depends on the amount of task M_r of customer r and the upload bandwidth capacity C_p^{uH} of provider p , but is also subject to the download bandwidth capacity C_r^d of customer r . Therefore, we can discuss whether a customer is able to pick up the entire amount of resources provided by providers through the assumption that the amount of resources provided by the providers improves, i.e., increasing provider’s upload bandwidth capacity gradually. As shown in Figure 11, we assume that the upload bandwidth capacity of providers in diverse settings are $(C_1^{uH}, C_2^{uH}, C_3^{uH}) = (10, 8, 5)Mb/s$, $(C_1^{uH}, C_2^{uH}, C_3^{uH}) = (30, 24, 15)Mb/s$, $(C_1^{uH}, C_2^{uH}, C_3^{uH}) = (50, 40, 25)Mb/s$, respectively.

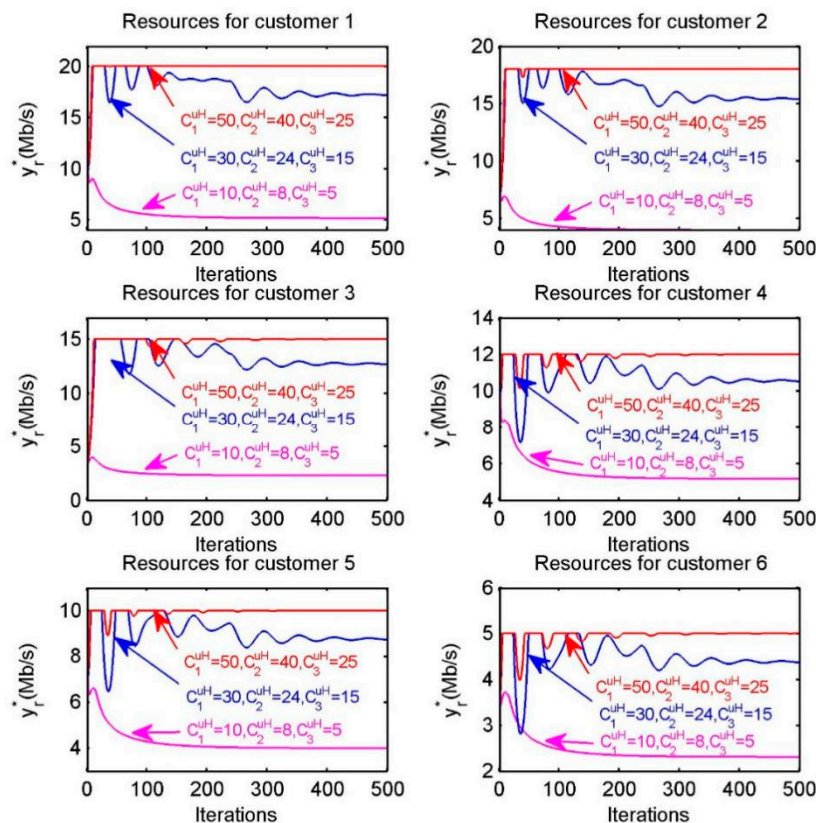


Figure 11. Resource allocation of each customer r when increasing the upload capacity of providers.

In each sub-figure of the Figure 11, the red, blue and purple lines correspond to the resource allocation under different upload bandwidth capacities. Obviously, when the upload bandwidth capacity of each provider increases, the amount of resources received by each customer is constrained

by its download bandwidth capacity. For instance, the red line in the sub-figure of “Resources for customer 1” is restricted within 20 Mb/s due to its download bandwidth capacity.

In Figure 12, we demonstrate the total bandwidth resource allocation of each customer, when different iteration step sizes are selected. For example, the first sub-figure “Resources for customer 1”, the blue curve with parameters $\kappa_{pr} = \alpha_r = \beta_r = 0.5$ converges at a significantly faster speed than the purple curve with parameters $\kappa_{pr} = \alpha_r = \beta_r = 0.1$. Based on the previous analysis and discussion on the convergence speed, we summarize that the convergence speed mainly depends on parameters such as step sizes rather than other factors. Therefore, we can get some conclusions, on the one hand, the iteration step sizes should be small enough to ensure convergence, however, it is unnecessary that step sizes are too small to slow the speed of convergence. On the other hand, the step sizes should also be not so large that the algorithm may not converge efficiently within the neighborhood of the optimum.

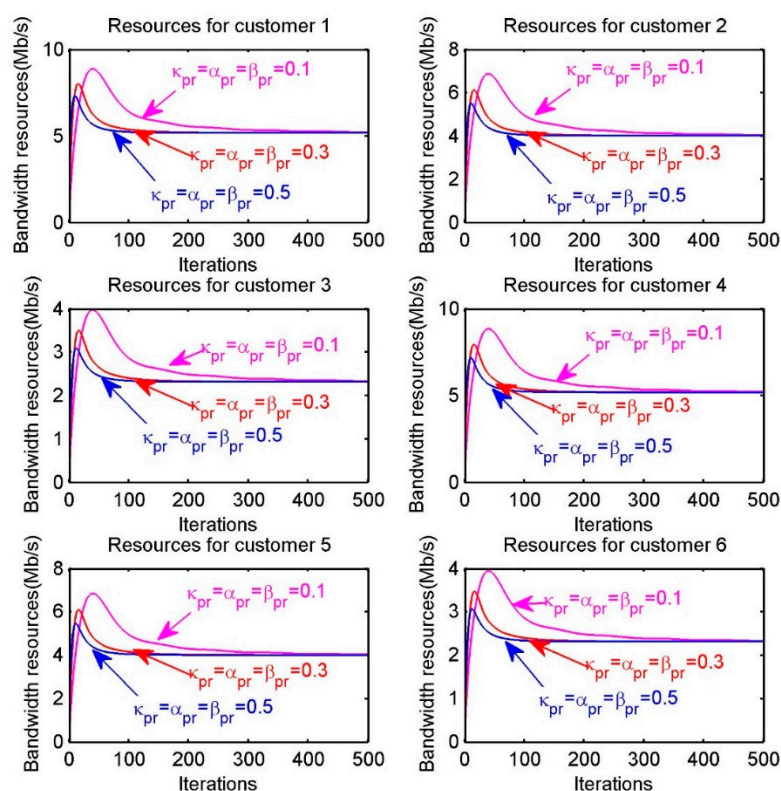


Figure 12. Resources allocation of each customer r when choosing different step sizes.

7. Conclusions

In recent years, the smart home market has developed rapidly and smart home products in various forms have begun to reach millions of households, greatly improving the convenience of people’s living. However, smart home applications are generally intelligent independently, which relies on the cloud platform to achieve remote control and realize collaborative work. Once attacked, there would be a serious threat and impact on the data privacy and security of smart homes. At the same time, as a decentralized computing framework, edge computing can provide networking, storage and computation services to IoT devices and reduce the bandwidth pressure of links. In a smart home environment, generally edge computing may face the issue of network latency and data security due to centralized cloud or data center, so edge computing without using cloud center has broad development potential as a new deployment choice for smart home. This paper discusses the bandwidth resource allocation of edge computing without using cloud center for processing tasks which are decomposed into two types, the tasks with low requirement on latency and high requirement on processing capacity,

and the tasks with high requirement on latency and low requirement on processing capacity in a smart home environment. According to the characteristics of different types of tasks, the utility maximization model of edge computing resource allocation is established and further interpreted from an economic point of view. We analyze the relationships between the prices charged by providers (fog nodes) for upload links and the prices paid by customers (users), and propose a gradient-based algorithm which can achieve optimal resource allocation for different tasks. Finally, some numerical examples are given to illustrate the effectiveness and convergence of the proposed algorithm.

Author Contributions: H.L., S.L. and W.S. have contributed equally to the development of this model and algorithm and its writing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the support from the National Natural Science Foundation of China (Nos.71671159 and 71971188), and the Natural Science Foundation of Hebei Province (Nos.G2018203302, G2020203005).

Acknowledgments: The authors would like to thank the anonymous reviewers and Associate Editor for very detailed and helpful comments and suggestions to improve this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, Q.; Li, Z.; Xie, W.; Zhang, Q. Edge Computing in Smart Homes. *J. Comput. Res. Dev.* **2020**, *9*, 1800–1809.
- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.K.; Alyasseri, Z.A.A. Optimization methods for power scheduling problems in smart home: Survey. *Renew. Sustain. Energy Rev.* **2019**, *115*, 109362. [CrossRef]
- Chinese Smart Home Market Outlook and Investment Strategy Planning Report for 2020–2025, Prospective Industrial Research Institute. Available online: <https://bg.qianzhan.com/report/detail/1609221718505703.html> (accessed on 14 November 2020).
- Hong, A.; Nam, C.; Kim, S. What will be the possible barriers to consumers' adoption of smart home services? *Telecommun. Policy* **2020**, *44*, 101867. [CrossRef]
- Al-Badi, A.H.; Ahshan, R.; Hosseinzadeh, N.; Ghorbani, R.; Hossain, E. Survey of Smart Grid Concepts and Technological Demonstrations Worldwide Emphasizing on the Oman Perspective. *Appl. Syst. Innov.* **2020**, *3*, 5. [CrossRef]
- Zhang, F.; Zhu, D.; Zang, W.; Yang, J.; Zhu, G. Edge Computing: Review and Application Research on New Computing Paradigm. *J. Front. Comput. Sci. Technol.* **2020**, *14*, 541–553.
- Challa, M.L.; Soujanya, K.L.S. Secured smart mobile app for smart home environment. *Mater. Today Proc.* **2020**, 1–5. [CrossRef]
- Tran, S.N.; Ngo, T.; Zhang, Q.; Karunanithi, M. Mixed-dependency models for multi-resident activity recognition in smart homes. *Multimed. Tools Appl.* **2020**, *79*, 1–16. [CrossRef]
- Song, F.; Ai, Z.; Zhou, Y.; You, I.; Choo, K.R.; Zhang, H. Smart collaborative automation for receive buffer control in multipath industrial networks. *IEEE T. Ind. Inform.* **2020**, *16*, 1385–1394. [CrossRef]
- Abdulkareem, K.H.; Mohammed, M.A.; Gunasekaran, S.S.; Al-Mhiqani, M.N.; Mutlag, A.A.; Mostafa, S.A.; Ali, N.S.; Ibrahim, D.A. A Review of Fog Computing and Machine Learning: Concepts, Applications, Challenges, and Open Issues. *IEEE Access* **2019**, *7*, 153123–153140. [CrossRef]
- Liu, L.; Chang, Z.; Guo, X.; Mao, S.; Ristaniemi, T. Multiobjective Optimization for Computation Offloading in Fog Computing. *IEEE Internet Things* **2018**, *5*, 283–294. [CrossRef]
- Grande, E.; Beltrán, M. Edge-centric delegation of authorization for constrained devices in the Internet of Things. *Comput. Commun.* **2020**, *160*, 464–474. [CrossRef]
- Feng, H.; Guo, S.; Zhu, A.; Wang, Q.; Liu, D. Energy-efficient User Selection and Resource Allocation in Mobile Edge Computing. *Ad. Hoc. Netw.* **2020**, *107*, 102202. [CrossRef]
- Rahimi, M.; Songhorabadi, M.; Kashani, M.H. Fog-based smart homes: A systematic review. *J. Netw. Comput. Appl.* **2020**, *153*, 102531. [CrossRef]
- Song, F.; Zhu, M.; Zhou, Y.; You, I.; Zhang, H. Smart Collaborative Tracking for Ubiquitous Power IoT in Edge-Cloud Interplay Domain. *IEEE Internet Things* **2020**, *7*, 6046–6055. [CrossRef]

16. Song, F.; Zhou, Y.; Wang, Y.; Zhao, T.; You, I.; Zhang, H. Smart collaborative distribution for privacy enhancement in moving target defense. *Inform. Sci.* **2019**, *479*, 593–606. [[CrossRef](#)]
17. Ghobaei-Arani, M.; Souri, A.; Rahmadian, A.A. Resource Management Approaches in Fog Computing: A Comprehensive Review. *J. Grid. Comput.* **2020**, *18*, 1–42. [[CrossRef](#)]
18. Zhang, K.; Gui, X.; Ren, D.; Li, J.; Wu, J.; Ren, D. Survey on computation offloading and content caching in mobile edge networks. *Ruan Jian Xue Bao/J. Softw.* **2019**, *30*, 2491–2516.
19. Song, F.; Zhou, Y.; Chang, L.; Zhang, H. Modeling Space-Terrestrial Integrated Networks with Smart Collaborative Theory. *IEEE Netw.* **2019**, *33*, 51–57. [[CrossRef](#)]
20. Chiang, W.; Li, C.; Shang, J.; Urban, T.L. Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Appl. Energy* **2019**, *242*, 1164–1175. [[CrossRef](#)]
21. Song, F.; Ai, Z.; Zhang, H.; You, I.; Li, S. Smart Collaborative Balancing for Dependable Network Components in Cyber-Physical Systems. *IEEE T. Ind. Inform.* **2020**. [[CrossRef](#)]
22. Guo, X.; Singh, R.; Zhao, T.; Niu, Z. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; IEEE: New York, NY, USA; pp. 1–7.
23. Oueis, J.; Calvanese, E.; Barbarossa, S. Small cell clustering for efficient distributed cloud computing. In Proceedings of the 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington, DC, USA, 2–5 September 2014; IEEE: New York, NY, USA; pp. 1474–1479.
24. Lee, S.; Choi, D. Energy management of smart home with home appliances, energy storage system and electric vehicle: A hierarchical deep reinforcement learning approach. *Sensors (Switzerland)* **2020**, *20*, 2157. [[CrossRef](#)] [[PubMed](#)]
25. Al-Ali, A.R.; Zualkernan, I.A.; Rashid, M.; Gupta, R.; Alikarar, M. A smart home energy management system using IoT and big data analytics approach. *IEEE T. Consum. Electr.* **2017**, *63*, 426–434. [[CrossRef](#)]
26. Wang, H.; Gong, J.; Zhuang, Y.; Shen, H.; Lach, J. HealthEdge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Shenzhen, China, 7–9 August 2017; pp. 1–2.
27. Procopiou, A.; Komninos, N.; Douligieris, C. ForChaos: Real time application DDoS detection using forecasting and chaos theory in smart home IoT network. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 1–14. [[CrossRef](#)]
28. Yang, L.; Feng, J.; Lu, X.L.; Liu, Q. Design of intelligent home security monitoring system based on Internet of Things. *Mod. Electron. Tech.* **2019**, *42*, 55–58.
29. Oueis, J.; Calvanese, E.; Domenico, A.D.; Barbarossa, S. On the impact of backhaul network on distributed cloud computing. In Proceedings of the 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Istanbul, Turkey, 6–9 April 2014; pp. 12–17.
30. Li, C.L.; Zhuang, H.; Wang, Q.F.; Zhou, X.H. SSLB: Self-Similarity-Based Load Balancing for Large-Scale Fog Computing. *Arab. J. Sci. Eng.* **2018**, *43*, 7487–7498. [[CrossRef](#)]
31. Oueis, J.; Calvanese, E.; Barbarossa, S. The Fog Balancing: Load Distribution for Small Cell Cloud Computing. In Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, Scotland, 11–14 May 2015; pp. 1–6.
32. Sun, Y.; Lin, F.; Xu, H. Multi-objective Optimization of Resource Scheduling in Fog Computing Using an Improved NSGA-II. *Wirel. Pers. Commun.* **2018**, *102*, 1369–1385. [[CrossRef](#)]
33. Zhou, P.; Yin, B.; Qiu, X.; Guo, S.; Meng, L. Service Reliability Oriented Cloud Resource Scheduling Method. *Acta Elect. Ronica Sinica* **2019**, *47*, 1036–1043.
34. Zheng, C.; Zhou, Z.; Tapani, R.; Niu, Z. Energy Efficient Optimization for Computation Offloading in Fog Computing System. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
35. Wang, X.; Ning, Z.; Wang, L. Offloading in Internet of Vehicles: A Fog-Enabled Real-Time Traffic Management System. *IEEE T. Ind. Inform.* **2018**, *14*, 4568–4578. [[CrossRef](#)]
36. Li, C.; Li, L. Cost and energy aware service provisioning for mobile client in cloud computing environment. *J. Supercomput.* **2015**, *71*, 1196–1223.
37. Li, S.; Sun, W. A mechanism for resource pricing and fairness in peer-to-peer networks. *Electron. Commer. Res.* **2016**, *16*, 425–451. [[CrossRef](#)]

38. Li, S.; Zhang, Y.; Sun, W. Optimal Resource Allocation Model and Algorithm for Elastic Enterprise Applications Migration to the Cloud. *Mathematics* **2019**, *7*, 909. [[CrossRef](#)]
39. Nguyen, D.T.; Le, L.B.; Bhargava, V. Price-based Resource Allocation for Edge Computing: A Market Equilibrium Approach. *IEEE T. Cloud. Comput.* **2018**. [[CrossRef](#)]
40. Lee, J.W.; Mazumdar, R.R.; Shroff, N.B. Non-convex optimization and rate control for multiclass services in the Internet. *IEEE/ACM Trans. Netw.* **2005**, *13*, 827–840. [[CrossRef](#)]
41. Chiang, M.; Low, S.; Calderbank, A.; Doyle, J. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE* **2007**, *95*, 255–312. [[CrossRef](#)]
42. Shakkottai, S.; Srikant, R. Network optimization and control. *Found. Trends Netw.* **2007**, *3*, 271–379. [[CrossRef](#)]
43. Li, S.; Sun, W.; Hua, C. Fair resource allocation and stability for communication networks with multipath routing. *Int. J. Syst. Sci.* **2014**, *45*, 2342–2353. [[CrossRef](#)]
44. Li, S.; Sun, W.; Li, Q. Utility maximization for bandwidth allocation in peer-to-peer file-sharing networks. *J. Ind. Manag. Optim.* **2020**, *16*, 1099–1117. [[CrossRef](#)]
45. Li, S.; Sun, W.; Tian, N. Resource allocation for multi-class services in multipath networks. *Perform. Eval.* **2015**, *92*, 1–23. [[CrossRef](#)]
46. Mo, J.; Walrand, J. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.* **2000**, *8*, 556–567. [[CrossRef](#)]
47. Li, S.; Sun, W. Utility maximisation for resource allocation of migrating enterprise applications into the cloud. *Enterp. Inf. Syst.-Uk* **2020**, 1–33. [[CrossRef](#)]
48. Tian, S.; Chen, H.; Li, P. Discussion of Step Function and Its Value at Time Zero. *J. Electr. Electron. Educ.* **2005**, *2*, 38–44.
49. Bertsekas, D.P. *Nonlinear Programming*; Athena Scientific: Belmont, MA, USA, 2003.
50. Li, M. Generalized Lagrange Multiplier Method and KKT Conditions with an Application to Distributed Optimization. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 252–256. [[CrossRef](#)]
51. Sarangi, S.K.; Panda, R.; Abraham, A. Design of optimal low-pass filter by a new Levy swallow swarm algorithm. *Soft Comput.* **2020**, *24*, 18113–18128. [[CrossRef](#)]
52. Stephen, B.; Lieven, V. *Convex Optimization*; Tsinghua University Press: Beijing, China, 2013; p. 702.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).