Received February 27, 2020, accepted March 12, 2020, date of publication March 20, 2020, date of current version April 1, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.2982293

Large Scale Resource Allocation for the Internet of Things Network Based on ADMM

YANHUA HE[1], (Student Member, IEEE), SUNXUAN ZHANG[1], LIANGRUI TANG[1], AND YUN REN[2]
[1]State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China
[2]State Grid Zhejiang Electric Power Company Ningbo Bureau, Zhejiang 315000, China

Corresponding author: Liangrui Tang (tlr@ncepu.edu.cn)

This work was supported in part by the National High Technology Research and Development of China 863 Program under Grant 2014AA01A701, in part by the Beijing Natural Science Foundation under Grant 4142049, and in part by the Fundamental Research Funds for the Central Universities under Grant 2018QN003.

ABSTRACT Large scale deployment of Internet of Things (IoT) devices poses challenges in resource allocation. In this paper, alternating direction method of multipliers (ADMM) is adopted to solve such large scale resource allocation problems. Based on this, three optimization problems are investigated in a hierarchical IoT network. Considering ADMM could not solve a non-convex optimization problem directly, a non-convex fractional programming problem i.e., energy efficiency maximization problem for IoT region server, is formulated. Faced with this problem, we introduce the Dinkelbach algorithm to transfer the energy efficiency maximization problem into an equivalent convex optimization problem. Then the classic ADMM with two blocks is employed to solve the equivalent convex optimization problem. On the other hand, the classic ADMM with two blocks could not satisfy the convergence speed demands of the high-dimensional convex optimization problems any more. Thus, the network latency minimization problem for controller is designed and then solved by the Jacobian-ADMM algorithm in parallel. It is hard to satisfy controller and IoT region servers' objectives at the same time. Given this, an incentive mechanism on the basis of Stackelberg game is designed. Thus a game-based resource allocation problem is proposed to deal with the contradiction between the centralized objective of the controller and the individual objectives from the IoT region servers. Based on the Dinkelbach algorithm and Jacobian-ADMM algorithm, a two-layer iterative resource allocation algorithm is posed to solve the game-based resource allocation problem. Last but not least, the convergence of the proposed algorithms are analyzed with numerous simulation results.

INDEX TERMS IoT network, large scale resource allocation, ADMM, convex optimization.

I. INTRODUCTION

The Internet of Things (IoT) is an emerging technology that proffers to connect massive smart devices together and to the Internet [1]. IoT technology has been widely used in the construction of smart city and smart grid, due to the advantages of ubiquitous sensing, universal networking, intelligent information processing, and real-time control. For example, tremendous amount of devices are deployed to monitor the physical world in people's daily lives in real time by collecting and uploading their local sensed contents such as images, videos, and textual data. With the ongoing worldwide development of IoT, more than 50 billion IoT devices are predicted to be connected with the expanding IoT by 2020 [2]. The IoT network is thirsty for an efficient resource allocation scheme which could adapt the increasing expansion of IoT network [3]–[5].

Convex optimization is an indispensable tool for resource allocation due to the ability of flexible formulations and efficient global optimization. In addition, convex relaxation approach is powerful to deal with the problems with complicated variables, such as the joint channel selection and resource allocation problems which include both discrete and continuous variables [6]. Importantly, convex relaxation approach provides a principled way of developing polynomial-time algorithms for non-convex or NP-hard

The associate editor coordinating the review of this manuscript and approving it for publication was Takuro Sato.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

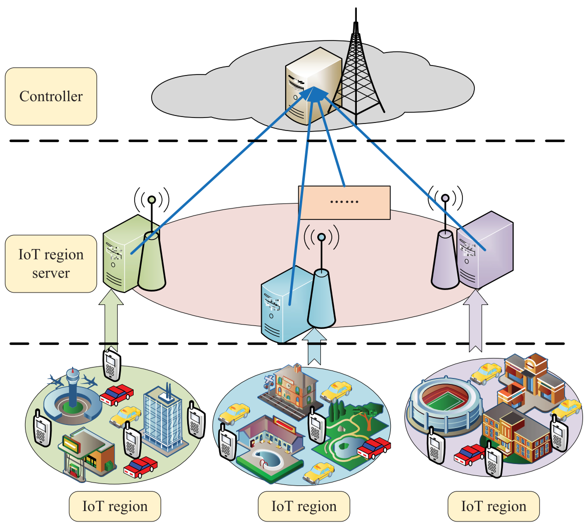


FIGURE 1. The hierarchical architecture of IoT network.

problems [7]. For example, group-sparsity penalty relaxation is utilized for the NP-hard mixed-integer nonlinear programming problems in [8] and [9]. Furthermore, [10] applies the general iterative successive convex approximation (SCA) approach to solve a mixed-integer non-convex resource allocation problem encountering high computational complexity.

However, performance optimization problems are entering a new era characterized by a high dimension and a large number of constraints because of the drastically increasing size of IoT network [11]. The general convex programming will no longer apply to the scaled up resource allocation problems. At this opportunity, alternating direction method of multipliers (ADMM) algorithm has attracted wide attentions for its distributed and parallel implementation, as well as the capability of scaling to large scale problems. ADMM has also been modified to solve many other mathematical optimization problems in more complicated and specific forms, such as multiagent ADMM with Gaussian back substitution for more than two variables [12], proximal Jacobian ADMM on parallel programming [13], and the asynchronous distributed ADMM [14]. ADMM has provided a powerful methodology to solve large-scale high-dimensional data processing and control optimization problems. Thus, ADMM has already been successfully applied in distributed energy management, machine learning, and image recognition [15], [16].

On the other hand, a generic hierarchical IoT architecture shown in Figure 1 is taken into account in this paper [17]. Such hierarchical architecture plays an irreplaceable role in cloud computing [18], access control [19] and data scheduling [20] for IoT network. In such a system, controller and IoT region servers usually have their own processing resource objectives [21]. In most cases, there are conflicts for the optimization objectives between controller and IoT region servers. Thus, an incentive mechanism which could balance the relationship between controller and IoT region servers is expected.

Last but not least, once the IoT network expands rapidly, the coupling relationship between the controller and IoT region servers will become complicated and changeable. On this occasion, ADMM is expected to be employed to deal with the increasing number of optimization variables and constraints. However, how to deal with the relationship between ADMM and incentive mechanism remains an open issue.

Motivated by this, three optimization problems are formulated. They are the energy efficiency maximization problems for IoT region servers, the network latency minimization problem for controller and the game-based resource allocation problem. Correspondingly, ADMM-and-Dinkelbach-based resource allocation algorithm, Jacobian-ADMM-based resource allocation parallel algorithm and game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm are proposed to solve these formulated three problems. The major contributions of this paper are summarized as follow:

- Three optimization problems are formulated in a large scale hierarchical IoT system. Taking the increasing energy consumption caused by the seamless deployment of IoT devices into consideration, a non-convex energy efficiency maximization problem is proposed for each IoT region server. For the controller, a network latency minimization problem is formulated to satisfy the demand of real-time transmission. Importantly, an incentive mechanism based on the Stackelberg game is introduced. Then a game-based resource allocation problem is designed.
- The energy efficiency maximization problem for each IoT region server is formulated as a non-convex fractional optimization problem, which could not be solved by ADMM algorithm directly. Faced with this, the ADMM-and-Dinkelbach-based resource allocation algorithm is proposed. First, the Dinkelbach algorithm is employed to transform the formulated energy efficiency maximization problem into an equivalent convex optimization problem. Then, the new equivalent convex optimization problem is solved by the classic ADMM with two blocks.
- The network latency minimization problem for controller is formulated as a high dimension convex optimization problem with a large number of variables constraints. The classic ADMM with two blocks is no longer applied to such a large scale resource allocation problem. Based on this, we propose a distributed parallel algorithm which is named as Jacobian-ADMM-based resource allocation parallel algorithm. Under this algorithm, controller's computing tasks are offloaded to IoT region servers.
- It is considered that the controller only focuses on its own objective, and the IoT region servers are selfish to achieve their own objectives. To address this issue, the Stackelberg game is introduced. Under the Stackelberg game architecture, the controller acts as

the leader and the IoT region servers play the roles of followers. Then the leader will incentivize the followers with payments to compensate their losses. The detailed works include

  - First, in order to encourage the IoT region servers to schedule the resource to accomplish the controller's objective, an incentive function is designed for IoT region servers. The incentive function includes the controller's objective information, the IoT region server's objective part, and the incentive part.
  - Second, the incentive function will be optimized instead of the original energy efficiency maximization problem. Then, a game-based resource allocation problem is designed.
  - In order to solve the game-based resource allocation problem with a tractable method, we propose a game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm. In the inter loop, the followrs' problems are solved based on Dinkelbach algorithm and Jacobian ADMM. More importantly, after the inter loop iteration ends, the optimized resource allocation results will be sent to the leader. In the outer loop, the leader obtains the response information from inter loop and then updates the incentive parameter. The incentive parameter will feed back to the followers. The iterative process will not stop until the Stackelberg equilibrium is achieved.

- At last, we analyze the performance of the three proposed algorithm from theory and simulations. The easily ignored convergence performance is verified and guaranteed. Specially, the proposed algorithms with rapid-convergence and strong-scalability could be applied in a large scale IoT network well.

The remained paper is organized as follows. Section II presents the system model. In Section III the Dinkelbach algorithm and classic ADMM are introduced to solve the energy efficiency maximization problem. In Section IV, the network latency minimization problem is presented and solved by the proposed Jacobian-ADMM-based resource allocation parallel algorithm. Then a game-based resource allocation problem is formulated to balance the controller's objective and IoT region servers' objectives in Section V, which is solved by the game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm. The simulation results are shown in Section VI and conclusions are drawn in Section VII.

II. SYSTEM MODEL

In this paper, we consider an uplink scenario in a centrally controlled system, i.e., the three hierarchical architecture of IoT network, as shown in Figure 1. The top layer is a central controller, which is responsible for system initialization, IoT device registration and so on. In the lowest layer, there are several IoT regions distributed. A number of heterogeneous mix of IoT devices, such as vehicles, human devices and smart city IoT devices, are deployed in these IoT regions. The second layer is consisted of IoT region servers, which are used to cache, compute, transmit data, and send the control information to IoT devices from itself or controller. The IoT region servers which are combined with base stations are functioned with power control. Moreover, the IoT region servers could not directly communicate with each other. The coordination between IoT region servers depends on the central controller.

In addition, each IoT region servers specially provides services to its own IoT region. Assuming that there are N IoT regions, correspondingly there are N IoT region servers which are indexed as $\mathcal{N} = \{1, 2, \cdots, n, \cdots, N\}$. Moreover, there are K_n IoT devices deployed in the $n-th$ IoT region, these IoT devices are indexed as $\mathcal{K}_n = \{1, 2, \cdots, i, \cdots, K_n\}$, $\forall n \in \mathcal{N}$.

Regarding to channel models of this system, both fast fading and slow fading that are caused by multipath propagation, shadowing and path loss are taken into consideration. The channel gain from the $i-th$ IoT device to its $n-th$ IoT region server is given by

$$g_{n,i} = \varpi \beta_{n,i} \zeta_{n,i} d_{n,i}^{-\alpha} \tag{1}$$

where ϖ is the path loss constant, $\beta_{n,i}$ is the fast-fading gain with exponential distribution, $\zeta_{n,i}$ is the slow fading gain with log-normal distribution, α is the path loss exponent, and $d_{n,i}$ is the transmission distance between the $i-th$ IoT device and its $n-th$ IoT region server.

Each device is allocated with an orthogonal link with the bandwidth of w Hz. The achievable data transmission rate from $i-th$ IoT device to its $n-th$ IoT region server is calculated as

$$R_{n,i}(p_{n,i}) = w\log_2\left(1 + p_{n,i}h_{n,i}\right) \tag{2}$$

$h_{n,i}$ is the signal-to-noise ratio (SNR) which is given by $\frac{g_{n,i}}{\sigma^2}$. σ^2 is the white Gaussian noise power and $p_{n,i}$ is the transmission power of the $i-th$ IoT device in the $n-th$ IoT region.

IoT network will be the leading energy guzzler in information and communications technology, which inspires the IoT region servers to focus on the energy efficiency in order to build a green IoT network [22]. So the energy efficiency of the $n-th$ IoT region is evaluated by

$$\eta_n = \frac{\sum_{i=1}^{K_n} R_{n,i}(p_{n,i})}{\sum_{i=1}^{K_n} p_{n,i}} \tag{3}$$

Assume that the arriving data packets with the length of $L_{n,i}$ bits of device i follow the Poisson point distribution process, so the transmission latency $\tau_{n,i}$ from $i-th$ IoT device to the

$n-th$ IoT region server is given by

$$\tau_{n,i}(p_{n,i}) = \frac{L_{n,i}}{R_{n,i}(p_{n,i})} \tag{4}$$

So transmission latency of the $n-th$ IoT region is calculated as

$$\tau_n(p_n) = \sum_{i=1}^{K_n} \tau_{n,i}(p_{n,i}) \tag{5}$$

where $p_n = \{(p_{n,1}), (p_{n,2}), \ldots, (p_{n,K_n})\}^T$ is a column vector. In this system the transmission latency of IoT network is given by

$$\mu(\mathbf{p}) = \sum_{n=1}^{N} \tau_n(p_n) \tag{6}$$

Here $\mathbf{p} = \{p_1^T; p_2^T; \ldots; p_N^T\}^T$.

III. ENERGY EFFICIENCY MAXIMIZATION PROBLEMS FOR IoT REGION SERVERS

In this section, an energy efficiency maximization problem is formulated. As a nonlinear fractional programming problem, the energy efficiency maximization problem is transferred into an equivalent convex optimization problem by Dinkelbach algorithm. Then the classic ADMM with two blocks is introduced to solve the equivalent convex optimization problem efficiently.

A. ENERGY EFFICIENCY MAXIMIZATION PROBLEMS FORMULATIONS AND TRANSFORMATIONS

Based on (3), the energy efficiency maximization problem for each IoT region server is formulated as

$$\begin{aligned} \mathbf{P1}': \ & \underset{p_n}{maximize} \ \ \eta_n \\ & s.t. \sum_{i=1}^{K_n} p_{n,i} \le p_n^{\max} \ \ n \in \mathcal{N}, \quad i \in \mathcal{K}_n \ (C1) \\ & \quad p_{n,i} \le p_{n,i}^{\max} \ \ n \in \mathcal{N}, \quad i \in \mathcal{K}_n \ (C2) \end{aligned} \tag{7}$$

where $C1$ is the upper bound of the energy consumption for the $n-th$ IoT region. $p_n^{\max}$ is the maximum available power of the $n-th$ IoT region. $C2$ is employed to limit the power of single IoT device. $p_{n,i}^{\max}$ is the maximum consumed power for single IoT device.

We can find that $\mathbf{P1}'$ is a non-convex fractional programming problem, where the numerator is a convex function of q_n and the denominator is an affine function of q_n. Such non-convex fractional programming problems could be transformed into a series of convex optimization problems with Dinkelbach algorithm. The Dinkelbach algorithm is widely adopted to solve such fractional programming problems [11], especially in green communication networks. So the objective of $\mathbf{P1}'$ is equivalent to minimizing the following function

$$f(p_n, q_n) = -\sum_{i=1}^{K_n} R_{n,i}(p_n) + q_n \sum_{i=1}^{K_n} p_{n,i} \tag{8}$$

where $f(p_n, q_n)$ is a strictly monotonic increasing function of q_n. Further, $f(p_n, q_n)$ could be proved to be a convex function of p_n by deriving the Hessian matrix.

Mathematically, $p_n^* = \{(p_{n,1}^*), (p_{n,2}^*), \cdots, (p_{n,K_n}^*)\}^T$ is denoted as the optimal power allocation results when the $n-th$ IoT region server obtain the optimal energy efficiency q_n^*. Therefore, the following theorem could be proved.

Theorem 1: The optimal energy efficiency result could be obtained by the $n-th$ IoT region server, if and only if

$$f(p_n^*, q_n^*) = -\sum_{i=1}^{K_n} R_{n,i}(p_n^*) + q_n^* \sum_{i=1}^{K_n} p_{n,i}^* = 0.$$

Proof: The detailed proof of Theorem 1 could be found in [23]. ∎

According to Theorem 1, the original optimization problem $\mathbf{P1}'$ could be transformed as

$$\begin{aligned} \mathbf{P1}: \ & \underset{p_n}{minimize} \ \ f(p_n, q_n) \\ & s.t. \ C1, C2 \end{aligned} \tag{9}$$

B. ADMM-AND-DINKELBACH-BASED RESOURCE ALLOCATION ALGORITHM

$\mathbf{P1}$ is a convex optimization problem with $K_n, n \in \mathcal{N}$ variables. When the number of IoT devices in $n-th$ IoT region increases, tremendous time will be taken to solve $\mathbf{P1}$ using the conventional convex optimization approach. Thus, we develop an ADMM-and-Dinkelbach-based resource allocation Algorithm. Then, the detailed algorithm is shown as follow.

First, the resource allocation variable p_n is divided into two parts, i.e., $\mathbf{x} = \{(p_{n,1}), (p_{n,2}), \cdots, (p_{n,m})\}^T$ and $\mathbf{z} = \{(p_{n,m+1}), (p_{n,m+2}), \cdots, (p_{n,K_n})\}^T$. Thus, $\mathbf{P1}$ could be rewritten as

$$\begin{aligned} \widetilde{\mathbf{P1}}: \ & \underset{\mathbf{x},\mathbf{z}}{minimize} \ \ \Gamma(\mathbf{x}) + \Psi(\mathbf{z}) \\ & s.t. \ \mathbf{E_x x} + \mathbf{E_z z} = p_n^{\max} \end{aligned} \tag{10}$$

where $\mathbf{x} \in \mathbf{R}^{m\times 1}$, $\mathbf{z} \in \mathbf{R}^{(K_n-m)\times 1}$, $\mathbf{E_x} \in \mathbf{R}^{1\times m}$, and $\mathbf{E_z} \in \mathbf{R}^{1\times(K_n-m)}$. $\mathbf{E_x}$ and $\mathbf{E_z}$ are unit vectors.

$\Gamma(\mathbf{x})$ and $\Psi(\mathbf{z})$ are list as

$$\Gamma(\mathbf{x}) = -\sum_{i=1}^{m} R_{n,i}(p_{n,i}) + q_n \sum_{i=1}^{m} p_{n,i} \tag{11}$$

$$\Psi(\mathbf{z}) = -\sum_{i=m+1}^{K_n} R_{n,i}(p_{n,i}) + q_n \sum_{i=m+1}^{K_n} p_{n,i} \tag{12}$$

Thus, the augmented Lagrangian associated with $\widetilde{\mathbf{P1}}$ is shown as

$$\mathbf{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \Gamma(\mathbf{x}) + \Psi(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{r}+\boldsymbol{\mu}\|_2^2 - \frac{\rho}{2}\|\boldsymbol{\mu}\|_2^2 \tag{13}$$

where $\mathbf{r} = \mathbf{E_x x} + \mathbf{E_z z} - p_n^{\max}$ is the primal residual. $\rho > 0$ represents the penalty parameter. Let $\mathbf{y}$ be a vector of Lagrange multipliers. $\boldsymbol{\mu} = \frac{\mathbf{y}}{\rho}$ is the scaled dual variables.

Then, we can iteratively update both primal and dual variables as

$$\mathbf{x}^{j+1} = \arg\min \Gamma\left(\mathbf{x}^j\right) + \frac{\rho}{2}\left\|\mathbf{E_x}\mathbf{x}^j + \mathbf{E_z}\mathbf{z}^j - p_n^{\max} + \boldsymbol{\mu}^j\right\|_2^2 \quad (14)$$

$$\mathbf{z}^{j+1} = \arg\min \Psi\left(\mathbf{z}^j\right) + \frac{\rho}{2}\left\|\mathbf{E_x}\mathbf{x}^{j+1} + \mathbf{E_z}\mathbf{z}^j - p_n^{\max} + \boldsymbol{\mu}^j\right\|_2^2 \quad (15)$$

$$\boldsymbol{\mu}^{j+1} = \boldsymbol{\mu}^j + \mathbf{E_x}\mathbf{x}^{j+1} + \mathbf{E_z}\mathbf{z}^{j+1} - p_n^{\max} \quad (16)$$

where j denotes the index of iteration.

The primal residual $\mathbf{r}_p$ and the dual residual $\mathbf{r}_d$ are expressed as

$$\mathbf{r}_p^{j+1} = \mathbf{E_x}\mathbf{x}^{j+1} + \mathbf{E_z}\mathbf{z}^{j+1} - p_n^{\max} \quad (17)$$

$$\mathbf{r}_d^{j+1} = \rho\mathbf{E_x}^T\mathbf{E_z}\left(\mathbf{z}^{j+1} - \mathbf{z}^j\right) \quad (18)$$

The termination criteria for ADMM is defined as

$$\left\|\mathbf{r}_p^{j+1}\right\|_2 \leq \epsilon^{pri} \quad and \quad \left\|\mathbf{r}_d^{j+1}\right\|_2 \leq \epsilon^{dual} \quad (19)$$

where $\epsilon^{pri} > 0$ and $\epsilon^{dual} > 0$ denote feasibility tolerances with respect to primal conditions and dual conditions.

On the other hand, q_n will update its value during every iteration and finally reach the optimal q_n^* according to Dinkelbach algorithm. q_n^{j+1} could be updated at j iteration as

$$q_n^{j+1} = \frac{\sum_{i=1}^{K_n} R_{n,i}\left(\left((\mathbf{x}^{j+1})^T, (\mathbf{y}^{j+1})^T\right)^T\right)}{\mathbf{E_x}\mathbf{x}^{j+1} + \mathbf{E_z}\mathbf{z}^{j+1}} \quad (20)$$

The termination criteria for Dinkelbach algorithm is denoted as

$$\pi^{j+1} = \left|\Gamma\left(\mathbf{x}^{j+1}\right) + \Psi\left(\mathbf{z}^{j+1}\right) - \Gamma\left(\mathbf{x}^j\right) - \Psi\left(\mathbf{z}^j\right)\right| \leq \varepsilon \quad (21)$$

Here ε is a positive constant that approaches to zero.

Based on the above work, the proposed ADMM-and-Dinkelbach-based resource allocation algorithm for IoT region servers will stop iteration until (19) and (21) are satisfied at the same time. The detailed ADMM-and-Dinkelbach-based resource allocation algorithm is shown as Algorithm 1.

IV. NETWORK LATENCY MINIMIZATION PROBLEM FOR CONTROLLER

In this section, a network latency minimization problem is formulated, which is a convex problem with $\sum_{n=1}^{N} K_n$ variables and $\sum_{n=1}^{N} K_n + N$ constraints. Then the Jacobian-ADMM-based resource allocation parallel algorithm is proposed to solve the network latency minimization problem.

Algorithm 1 ADMM-and-Dinkelbach-Based Resource Allocation Algorithm

Initialize: j, $\mathbf{x}$, $\mathbf{z}$, $\boldsymbol{\mu}$, q_n, ρ, ϵ^{pri}, ϵ^{dual}, ε and π.
output: $\mathbf{x}$, $\mathbf{z}$, q_n.
while $\left\|\mathbf{r}_p{}^j\right\|_2 > \epsilon^{pri}$ or $\left\|\mathbf{r}_d{}^j\right\|_2 > \epsilon^{dual}$ or $\pi^j > \varepsilon$ **do**
  Update $\mathbf{x}^{j+1}$ according to (14);
  Update $\mathbf{z}^{j+1}$ according to (15);
  Update $\boldsymbol{\mu}^{j+1}$ according to (16);
  Update $\left\|\mathbf{r}_p{}^{j+1}\right\|_2$ according to (17);
  Update $\left\|\mathbf{r}_d{}^{j+1}\right\|_2$ according to (18);
  Update q_n^{j+1} according to (20);
  Update π^{j+1} according to (21);
  Update $j \rightarrow j+1$;
end while

A. NETWORK LATENCY MINIMIZATION PROBLEM FORMULATIONS

Notably, many envisioned applications of IoT, such as industrial automation [24], vehicle-to-everything (V2X) networks, smart grids, and remote surgery, have stringent transmission latency and reliability requirements. Thus, the network latency minimization problem for controller is formulated as

$$\begin{aligned} \mathbf{P2}: \quad & \underset{\mathbf{p}}{minimize} \quad \mu(\mathbf{p}) \\ & s.t. \sum_{i=1}^{K_n} p_{n,i} \leq p_n^{\max} \quad \forall n \in \mathcal{N} \ (C3) \\ & p_{n,i} \leq p_{n,i}^{\max} \quad \forall n \in \mathcal{N}, \ \forall i \in \mathcal{K}_n \ (C4) \end{aligned} \quad (22)$$

where $C3$ and $C4$ are the energy consumption constraints for all IoT regions and all IoT devices, respectively.

B. JACOBIAN-ADMM-BASED RESOURCE ALLOCATION PARALLEL ALGORITHM

P2 can be proved as a convex optimization problem by verifying its corresponding second-order derivative. The classic ADMM with two blocks used in Algorithm 1, is no longer suitable for convex optimization problems with high dimensional variables such as **P2**. We prefer to extending the ADMM framework to solve **P2**, especially when K_n or N is large. A natural extending is to simply replace the two-block alternating minimization scheme by a sweep of Gauss-Seidel update. However such Gauss-Seidel ADMM has a disadvantage, i.e., the blocks are updated one after another which is not amenable for parallelization [13]. To overcome this disadvantage, we will introduce a Jacobian-type scheme which updates all the blocks in parallel.

On one hand, the controller achieves its goal by coordinating the resource allocation schemes from N IoT region servers. On the other hand, compared with large-scale centralized computing, the controller is more inclined to decentralize computing tasks to IoT region servers. Thus we will partition the matrix of resource allocation variables into N parts, i.e.,

$$x_1 = \{(p_{1,1}), (p_{1,2}), \ldots, (p_{1,K_1})\}^T$$

$$x_2 = \{(p_{2,1}), (p_{2,2}), \ldots, (p_{2,K_2})\}^T$$
$$\vdots$$
$$x_n = \{(p_{n,1}), (p_{n,2}), \ldots, (p_{n,K_n})\}^T$$
$$\vdots$$
$$x_N = \{(p_{N,1}), (p_{N,2}), \ldots, (p_{N,K_N})\}^T$$

Then **P2** is rewritten as

$$\widetilde{\mathbf{P}}2: \ \underset{\mathbf{X}}{minimize} \sum_{n=1}^{N} G(x_n)$$
$$s.t. \sum_{n=1}^{N} \mathbf{A}_n x_n = \mathbf{C} \quad \forall n \in \mathcal{N} \ (C5) \tag{23}$$

where $\mathbf{X} = \{x_1^T; x_2^T; \ldots; x_N^T\}^T$, $\mathbf{A}_n \in \mathbf{R}^{N \times K_n}$. Let $\mathbf{C} = \{p_1^{max}, p_2^{max}, \ldots, p_N^{max}\}^T$. $G(x_n)$ satisfies

$$G(x_n) = \sum_{i=1}^{K_n} \tau_{n,i}(p_{n,i}) \tag{24}$$

The augmented Lagrangian associated with $\widetilde{\mathbf{P}}2$ is shown as

$$\mathcal{L}_\rho(\mathbf{X}, \lambda) = \sum_{n=1}^{N} G(x_n) - \lambda^T (\sum_{n=1}^{N} \mathbf{A}_n x_n - \mathbf{C}) + \frac{\kappa}{2} \left\| \sum_{n=1}^{N} \mathbf{A}_n x_n - \mathbf{C} \right\|_2^2 \tag{25}$$

$\kappa > 0$ represents the penalty parameter of the quadratic. λ is the Lagrange multiplier which is a column vector. Then, we can iteratively update primal variables as

$$x_1^{l+1} = \arg\min G(x_1) + \frac{\kappa}{2} \left\| \mathbf{A}_1 x_1 + \sum_{n=2}^{N} \mathbf{A}_n x_n^l - \mathbf{C} - \frac{\lambda^l}{\kappa} \right\|_2^2$$
$$\vdots$$
$$x_n^{l+1} = \arg\min G(x_n) + \frac{\kappa}{2} \left\| \sum_{a=1}^{n-1} \mathbf{A}_a x_a^l + \mathbf{A}_n x_n + \sum_{a=n+1}^{N} \mathbf{A}_a x_a^l - \mathbf{C} - \frac{\lambda^l}{\kappa} \right\|_2^2$$
$$\vdots$$
$$x_N^{l+1} = \arg\min G(x_N) + \frac{\kappa}{2} \left\| \mathbf{A}_N x_N + \sum_{n=1}^{N-1} \mathbf{A}_n x_n^l - \mathbf{C} - \frac{\lambda^l}{\kappa} \right\|_2^2 \tag{26}$$

Further, the dual variables are updated as

$$\lambda^{l+1} = \lambda^l - \kappa \left(\sum_{n=1}^{N} \mathbf{A}_n x_n - \mathbf{C} \right) \tag{27}$$

where l denotes the index of iteration.

According to [13] and [25], the termination criteria is defined as

$$\chi^{l+1} = \left| \mathcal{L}_\rho \left(\mathbf{X}^{l+1}, \lambda^{l+1} \right) - \mathcal{L}_\rho \left(\mathbf{X}^l, \lambda^l \right) \right| \leq \varepsilon \tag{28}$$

Consequently, the Jacobian-ADMM-based resource allocation parallel algorithm is summarized in Algorithm 2.

Remark 1: If the $n-th$ IoT region server can get the information related to the controller' objective, i.e., $G(x_n)$ and the constraint C5, the update progress of $x_n, \forall n \in \mathcal{N}$ could be completed by the $n-th$ IoT region server referred to (26). Then the updated $x_n, \forall n \in \mathcal{N}$ will be sent to the controller. The work of the controller is reduced to update dual variables and send the updated λ to IoT region servers. That is to say the proposed Algorithm 2 succeeds to decentralize the computing tasks of controller to N IoT region servers.

Algorithm 2 Jacobian-ADMM-Based Resource Allocation Parallel Algorithm

1: **Initialize:** l, $\mathbf{X}$, λ, κ, ε and χ.
2: **output:** $\mathbf{X}$.
3: **while** $\chi^l > \varepsilon$ **do**
4: Update $\mathbf{X}^{l+1}$ according to (26);
5: Update λ^{l+1} according to (27);
6: Update χ^{l+1} according to (28);
7: Update $l \to l+1$;
8: **end while**

V. GAME-BASED RESOURCE ALLOCATION PROBLEM

The key goal of this section is to efficiently utilize the limited energy resources, while guarantee the real-time transmission of data. That is to say we expect an optimal resource allocation scheme which satisfies the controller's and IoT region servers' objectives simultaneously. However, it is challenging to find such an optimal resource allocation scheme.

First, the controller could not accomplish the resource allocation without the help of IoT region server. Because the IoT devices' resources are handled by the IoT region servers directly. Second, IoT region servers are selfish and prefer to achieve its own objectives, unless the controller provides incentives for IoT region servers [26], i.e., paying token/money for additional consumed energy to IoT region servers. Third, as mentioned in Section IV, large scale centralized computing puts tremendous pressures on the controller and the computing tasks are expected to offloaded to IoT region servers.

Accordingly, we formulate a game-based resource allocation problem, where the controller is modeled as a leader and the IoT region servers act as followers. Then a game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm is developed to solve this problem in a distributed and parallel way.

A. GAME-BASED RESOURCE ALLOCATION PROBLEM FORMULATIONS

Based on the above discussions, the following rules are considered to design the incentive function for IoT region servers.

- The controller's objective information is better to be included in the incentive function. So that the IoT region servers can clarify controller's optimization direction to schedule resources.
- Though the incentive function will be optimized instead of the original objective listed in **P1**, the original objective of IoT region server must not be ignored.
- The gain part which follower gets from leader to make up for the energy efficiency losses is the non-negligible.

Therefore, the incentive function $\Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right)$ of the $n-th$ follower is illustrated as

$$\Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right) = L_n\left(p_n, \Lambda_n\right) + H_n\left(p_n, q_n, \theta_n\right) \quad (29)$$

where

$$L_n\left(p_n, \Lambda_n\right) = \tau_n(p_n) - \Lambda_n \mathbf{A}_n p_n \quad (30)$$
$$H_n\left(p_n, q_n, \theta_n\right) = f_n\left(p_n, q_n\right) - \theta_n p_n \quad (31)$$

where $\theta_n = \{(\theta_{n,1}), (\theta_{n,2}), \ldots, (\theta_{n,K_n})\}$ is the incentive parameter vector, which represents the unit price of energy. Λ_n is the Lagrange multiplier. $L_n\left(p_n, \Lambda_n\right)$ is a segmental Lagrangian function related to controller. $H_n\left(p_n, q_n, \theta_n\right)$ is calculated by subtracting the gain from the $n-th$ follower's objective. $\theta_n p_n$ is the gain obtained by $n-th$ IoT region server.

Remark 2: The incentive function could be understood as

- *Controller divides its Lagrangian function associated with $\mu(\mathbf{p})$ into N parts according to ADMM theory, which could be referred to (25). Then the part $L_n\left(p_n, \Lambda_n\right)$ related to the $n-th$ IoT region is fed back to the $n-th$ IoT region server.*
- *After receiving the objective information of controller, the $n-th$ IoT region server will take on the task of optimizing $L_n\left(p_n, \Lambda_n\right)$ with obtaining the payment of $\theta_n p_n$ from controller.*
- *At the same time, the $n-th$ follower' original optimization objective $f_n\left(p_n, q_n\right)$ is also optimized. Actually, $H_n\left(p_n, q_n, \theta_n\right)$ indicates the utility of the $n-th$ follower.*

In proceed, the $n-th$ IoT region server will optimize $\Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right)$ instead of $f_n\left(p_n, q_n\right)$. So a Stackelberg game is formulated as

$$\begin{aligned} \mathbf{P3}: Leader: &\ \underset{\mathbf{p}}{minimize}\ \ \mu(\mathbf{p}) \\ Follwer: &\ \underset{p_n, q_n}{minimize}\ \ \Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right) \\ &\ s.t.\ C4, C5 \end{aligned} \quad (32)$$

For leader's objective, each part $\tau_n(p_n)$ is a strongly convex function of p_n. Furthermore, the followers objective $f_n\left(p_n, \theta_n\right)$ is a strongly convex function of p_n. Thus, $\Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right)$ is a strongly convex function of p_n. Considering the tremendous amount of variables and constrains, **P3** is encouraged to be computed in parallel. So a game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm is designed as follows.

B. GAME-AND-JACOBIAN-ADMM-BASED TWO-LAYER ITERATIVE RESOURCE ALLOCATION ALGORITHM

At the beginning, the leader transmits the form of $L_n\left(p_n, \Lambda_n\right)$ to the $n-h$ follower, $\forall n \in \mathcal{N}$. Define υ as the iteration times of outer loop. At each step υ, the followers update the resources allocation scheme $p_n\{\upsilon\}, \forall n \in \mathcal{N}$ to minimize their incentive functions, while the given incentive factors from the leaders, $\theta_n\{\upsilon\}, \forall n \in \mathcal{N}$, are taken into account. At the next step $\upsilon+1$, leader will adjust the incentive factors based on the updated resource allocation $p_n\{\upsilon\}, \forall n \in \mathcal{N}$ [27]. Then, the followers will reach a new optimal resource allocation scheme $p_n\{\upsilon+1\}, \forall n \in \mathcal{N}$, with the adjusted incentive parameter $\theta_n\left(\upsilon+1\right), \forall n \in \mathcal{N}$.

Based on the above discussion, the process to achieve $p_n\{\upsilon\}, \forall n \in \mathcal{N}$ at each step υ is called the inner loop, and the one to update $\theta_n\{\upsilon+1\}, \forall n \in \mathcal{N}$ from step υ to step $\upsilon+1$ is called the outer loop. t is defined as the iteration times for the inner loop. The detailed updating processes of the inter loop and outer loop are presented as follows.

1) INTER LOOP

Parallelly follower's update:

$$\begin{aligned} p_n^{t+1}\{\upsilon\} = &\arg\min L_n\left(p_n^t\{\upsilon\}, \Lambda_n^t\{\upsilon\}\right) \\ &+ H_n\left(p_n^t\{\upsilon\}, q_n^{t+1}\{\upsilon\}, \theta_n\{\upsilon\}\right) \\ &+ \frac{\rho}{2}\left\|\sum_{n=1}^{N}\mathbf{A}_n p_n^t\{\upsilon\} - \mathbf{C}\right\|_2^2 \end{aligned} \quad (33)$$

Leader's dual update:

$$\Lambda^{t+1}\{\upsilon\} = \Lambda^t\{\upsilon\} - \rho\left(\sum_{n=1}^{N}\mathbf{A}_n p_n^t\{\upsilon\} - \mathbf{C}\right) \quad (34)$$

Here $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n, \ldots, \Lambda_N\}^T$.

q_n update based on Dinkelbach algorithm:

$$q_n^{t+1}\{\upsilon\} = \frac{\sum_{i=1}^{K_n} R_{n,i}^t(p_{n,i}^t)\{\upsilon\}}{\sum_{i=1}^{K_n} p_{n,i}^t\{\upsilon\}} \quad (35)$$

$$\Delta_\Phi\{\upsilon\} = \left|\Phi_n\left(f_n\left(p_n^{t+1}\{\upsilon\}, q_n^{t+1}\{\upsilon\}\right), \theta_n^{t+1}\{\upsilon\}\right) - \Phi_n\left(f_n\left(p_n^t\{\upsilon\}, q_n^t\{\upsilon\}\right), \theta_n^t\{\upsilon\}\right)\right| \quad (36)$$

Inter loop termination criteria: The termination criteria of inter loop is defined as (36), as shown at the bottom of the previous page, when $\Delta_\Phi\{\upsilon\} \le \varepsilon$ the inter loop will stop iteration.

2) OUTER LOOP

a: INCENTIVE PARAMETER UPDATE

After the inner loop, each follower feeds back its marginal cost $\nabla_{p_n^t\{\upsilon\}} H_n\left(p_n^t\{\upsilon\}\right)$ to the leader. The leader then changes its strategy as the adjustment of incentive parameters, to set the price $\theta_n\{\upsilon+1\}$, as the current marginal cost of each follower, i.e.,

$$\theta_n\{\upsilon+1\} = \nabla_{p_n^t\{\upsilon\}} H_n\left(p_n^t\{\upsilon\}\right) \tag{37}$$

Then, the leader and followers can reach a new optimal point $\{p_n, \Lambda_n\}$ at the next step $\upsilon+1$.

b: OUTER LOOP TERMINATION CRITERIA

The outer loop ends when the Lagrangian function of the leader satisfies a primal stopping criterion $\Delta_L\{\upsilon\} \le \varepsilon$, $\Delta_L\{\upsilon\}$ is given by

$$\begin{aligned}\Delta_L&\{\upsilon\}\\ &= L\left(p_n^t\{\upsilon+1\}, \Lambda_n^t\{\upsilon+1\}\right) - L\left(p_n^t\{\upsilon\}, \Lambda_n^t\{\upsilon\}\right)\end{aligned} \tag{38}$$

The augmented Lagrangian function $L\left(p_n, \Lambda_n\right)$ is the sum of segmental Lagrangian functions, i.e., $L_n\left(p_n, \Lambda_n\right), \forall n \in \mathcal{N}$ plus a second-order norm of the constraint, which is given by

$$\begin{aligned}L\left(p_n^t\{\upsilon\}, \Lambda_n^t\{\upsilon\}\right) &= \sum_{n=1}^{N} L_n\left(p_n^t\{\upsilon\}, \Lambda_n^t\{\upsilon\}\right) + \sum_{n=1}^{N} \Lambda_n p_n^{\max}\\ &\quad + \frac{\rho}{2}\left\|\sum_{n=1}^{N} \mathbf{A}_n p_n^t\{\upsilon\} - \mathbf{C}\right\|_2^2\end{aligned} \tag{39}$$

Based on the above work, the proposed game-and-Jacobian-ADMM-based two-layer iterative resource allocation algorithm is shown as Algorithm 3.

Algorithm 3 Game-and-Jacobian-ADMM-Based Two-Layer Iterative Resource Allocation Algorithm

Initialize: υ, θ_n, and ε.
Output: p_n^*, Λ^*, θ_n^*, $\forall n \in \mathcal{N}$.
Outer loop:
while $\Delta_L\{\upsilon\} > \varepsilon$ **do**
  Inter loop:
  Initialize: t, p_n, Λ, ρ and $\theta_n\{\upsilon\}$.
  Output: $p_n\{\upsilon\}$.
  while $\Delta_\Phi\{\upsilon\} > \varepsilon$ **do**
    Update ${p_n}^{t+1}\{\upsilon\}$ according to (33);
    Update $\Lambda^{t+1}\{\upsilon\}$ according to (34);
    Update $q_n^{t+1}\{\upsilon\}$ according to (35);
    Update $\Delta_\Phi\{\upsilon\}$ according to (36);
    Update $t \rightarrow t+1$;
  end while
  Update $\theta_n\{\upsilon\}$ according to (37);
  Update $\Delta_L\{\upsilon\}$ according to (38);
  Update $\upsilon \rightarrow \upsilon+1$;
end while

TABLE 1. Simulation parameters.

Parameter	Value
Number of IoT region server	$N = 4$
Number of devices in the $n-th$ IoT region server	$K_n = 100$
Channel bandwidth	$B = 1$ MHz
Maximum consumed energy for each IoT region	$p_n^{max} = 18$ watts
Maximum consumed energy for single device	$p_{n,i}^{max} = 0.2$ watts
Path loss exponent	$\alpha = 4$
Gaussian white noise	$\sigma^2 = -84$dBm

C. PERFORMANCE ANALYSIS

1) Convergence

Theorem 2: Algorithm 3 is guaranteed to linearly converge into a Stackelberg equilibrium. Mathematically, given any small scaler ε_0, when the primal residue of the augmented Lagrangian function is less than ε_0, $\Delta_L\{\upsilon\} \le \varepsilon_0$, the iteration time Υ has the following upper bound, $\Upsilon \le \frac{\Upsilon_0}{\varepsilon_0}$, where Υ_0 is a positive constant.

Proof: The proof of Theorem 2 could be found in Appendix. ∎

2) OPTIMALITY

In Algorithm 3, the followers' optimization problems, i.e. $\Phi_n\left(f_n\left(p_n, q_n\right), \theta_n\right), \forall n \in \mathcal{N}$, are guaranteed to be solved by the Jacobian-ADMM and Dinkelbach algorithm at any step υ. According to Theorem 2, the outer loop of Algorithm 3 linearly converges. Thus the leader's optimization problem is also been solved.

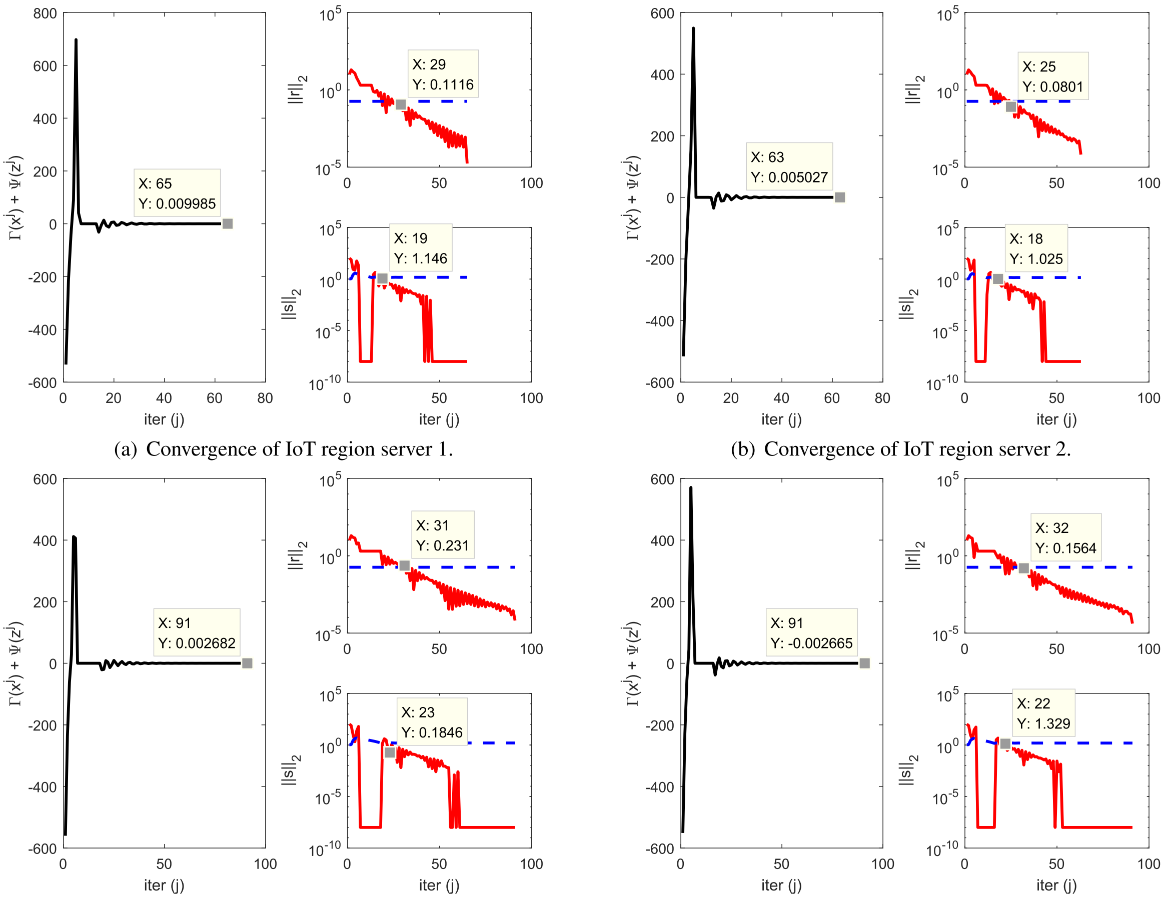
3) SCALABILITY

Through the proof of Theorem 2 in Appendix, we can find that the iteration time only depends on the selection of the initial point. It is indicated that the convergence speed of Algorithm 3 is independent of the number of followers or devices i.e.,N or K_n. Thus, Algorithm 3 is scalable and adaptable to the large scale resource allocation problem.

VI. SIMULATION RESULTS

In this section, we verify the performances of the proposed algorithms through simulation results. The parameters are listed in Table 1.

Figure 2 demonstrates the convergence speeds under Algorithm 1 for each IoT region server. As above mentioned, Algorithm 1 stops iteration when (19) and (21) are satisfied at the same time. From Figure 2, we can find the primal residual and dual residual demands i.e., (19), are met first for every IoT region server. Then Algorithm 1 keeps on updating for searching the optimal solution until (21) is satisfied. Moreover the objectives for IoT region servers are $0.009985, 0.005027, 0.002682, -0.002665$ which are approaching to zero. That is to say the $f_n(p_n^*, q_n^*) = 0$ is valid and the optimal solution p_n^* and q_n^* found by Algorithm 1 are



(a) Convergence of IoT region server 1.

(b) Convergence of IoT region server 2.

(c) Convergence of IoT region server 3.

(d) Convergence of IoT region server 4.

FIGURE 2. **Convergence of the Algorithm 1.**

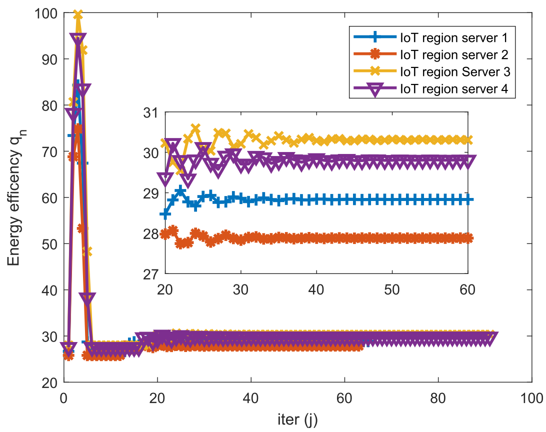


FIGURE 3. **$q_n, \forall n \in \mathcal{N}$ versus iteration times j.**

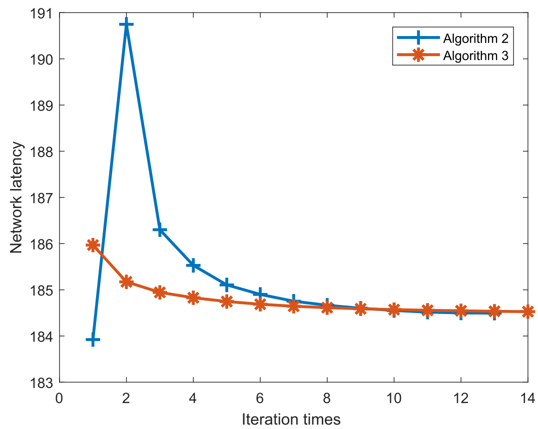


FIGURE 4. **Network latency versus iteration times.**

the demanded solution. The original optimization problem $\widetilde{\mathbf{P1}}$ could obtain the optimal objective q_n^* by Algorithm 1.

Figure 3 shows the detailed updating process of q_n, $\forall n \in \mathcal{N}$. The small picture in Figure 3 is a detailed enlarged picture from $j = 20$ to $j = 60$. It is observed that the numerical value of $q_n, \forall n \in \mathcal{N}$ fluctuates greatly when $j < 20$. However, the numerical value of $q_n, \forall n \in \mathcal{N}$ is gradually stable and converges to a constant.

Figure 4 shows the iteration processes of Algorithm 2 and Algorithm 3. Compare the convergence speeds of Algorithm 1 and Algorithm 2. Then we can find that Algorithm 1 has iterated for more than 20 times before the primal residual and dual residual are satisfied, but Algorithm 2 stops iteration when $l = 13$. In addition, Algorithm 1 solves an optimization problem with $K_n \times 1$ variables but Algorithm 2 solves an optimization problem with $N \times K_n$ variables. It is

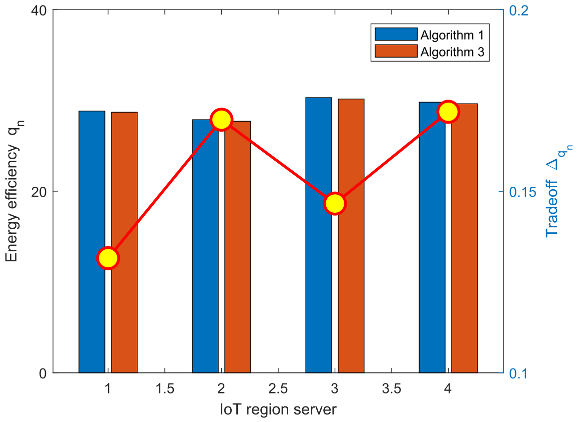


FIGURE 5. Comparison of $q_n, \forall n \in \mathcal{N}$ under Algorithm 1 and Algorithm 3.

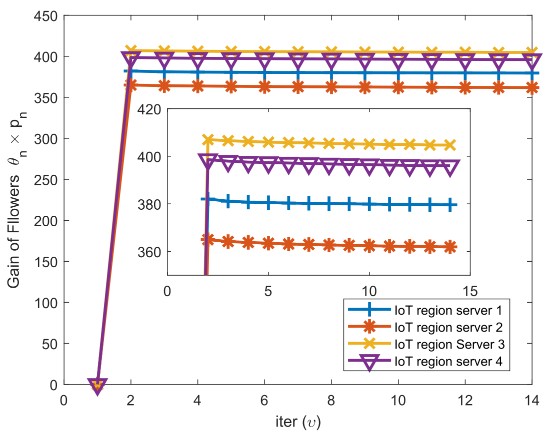


FIGURE 6. Gain $\theta_n q_n$ obtained by each IoT region server under Algorithm 3.

verified that the Jacobian ADMM is more adaptable to solve a large scale optimization problems for IoT network, compared to the classic ADMM.

Second, the outer loop of Algorithm 3 stops iteration when $\upsilon = 14$. Algorithm 3 effectively converges into the minimum point of leader's objective. Compared with the obtained objective of Algorithm 2, Algorithm 3 may get the same optimal objective. That is to say Algorithm 3 succeeds finding the optimal objective for leader under the Stackelberg game architecture.

Figure 5 uses the bar graphs to present the optimal energy efficiency obtained by Algorithm 1 and Algorithm 3. Moreover, the solid circles are used to show the energy efficiency losses $\Delta_{q_n}, \forall n \in \mathcal{N}$. Δ_{q_n} is calculated by subtracting the optimal objective of Algorithm 3 from the optimal objective of algorithm Algorithm 1. Compared with Algorithm 1, followers lost 0.45%, 0.61%, 0.49% and 0.57% energy efficiency respectively under the game architecture. Thus, it is considered the optimal $q_n^*, \forall n \in \mathcal{N}$ obtained by Algorithm 3 is almost equal to the optimal $q_n^*, \forall n \in \mathcal{N}$ obtained by Algorithm 1.

Furthermore, Figure 6 illustrates the every IoT region server's gain $\theta_n \times p_n, \forall n \in \mathcal{N}$. The small picture in Figure 6 is a detailed enlarged picture from $\upsilon = 2$ to $\upsilon = 14$. When $\upsilon = 1$, θ is set up as 0, so the gains of IoT region are equal to 0. Combining Figure 4, Figure 5, and Figure 6, we can conclude that the followers drive Algorithm 3 to achieve leader's goal by getting compensation from leader under the Stackelberg game architecture. With the incentive mechanism based on the Stackelberg game, not only the leader could obtain the optimal network latency, but also the followers could get the acceptable energy efficiency.

VII. CONCLUSION

In this paper, we investigate the large scale resource allocation problems in a centrally controlled hierarchical architecture. First, ADMM-and-Dinkelbach-based resource allocation algorithm is developed to solve a non-convex fractional optimization problem, i.e., energy efficiency maximization problem for IoT region server. Second, Jacobian-ADMM-based resource allocation parallel algorithm is used to solve a convex optimization problem with high dimensional variables, i.e., network latency minimization problem for controller. Last, a two-layer iterative resource allocation algorithm based on the Dinkelbach algorithm and Jacobian-ADMM algorithm is proposed to solve the game-based resource allocation problem. The game-based resource allocation problem aims at overcoming the conflicts between controller's objective and IoT region servers' objectives. Through the simulation results the scalability and convergence speeds of the proposed three algorithms are verified.

APPENDIX

Before proving the Theorem 2, some lemmas are given first.

Lemma 1: The original objective function of each follower satisfies a uniform Lipschitz gradient condition [26], there exits $\varsigma > 0$, for each $f_n(p_n, q_n)$, given any $p_n \prec= \widetilde{p_n}$, we have

$$\nabla_{\widetilde{p_n}} f_n\left(\widetilde{p_n}, q_n\right) - \nabla_{p_n} f_n\left(p_n, q_n\right) \prec= \varsigma\left(\widetilde{p_n} - p_n\right) \quad (40)$$

where $\mathbf{D} \prec= \widetilde{\mathbf{D}}$ indicates that each element of the vector $\mathbf{D}$ is less than or equal to that of the vector $\widetilde{\mathbf{D}}$. The Lipschitz gradient condition usually indicates that the first order gradient of a function cannot change too fast, where the speed is below ς.

Lemma 2: For any $\mathbf{p}\{\upsilon\}$ and $\Lambda\{\upsilon\}$, we have

$$\left\|\nabla_{p_n} L\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right)\right\|_2^2 \geq 2\varphi\left[L\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right) - L\left(p_n^*, \Lambda_n^*\right)\right] \quad (41)$$

According to ADMM and Dinkelbach algorithm, the inner loop in Algorithm 3 can reach a current optimal point, then we have the following equation for any index n,

$$\begin{aligned} 0 &= \nabla_{p_n} L_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \\ &\quad + \nabla_{p_n} H_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}, \theta_n\{\upsilon+1\}\right) \\ &= \nabla_{p_n} L_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \\ &\quad + \nabla_{p_n} f_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) - \theta_n\{\upsilon+1\} \\ &= \nabla_{p_n} L_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \\ &\quad + \nabla_{p_n} f_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \\ &\quad - \nabla_{p_n} f_n\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right) \end{aligned} \quad (42)$$

$$\begin{aligned} & L\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right) \\ & \geq L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)+\nabla_{p_n} L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \times\left(p_n\{\upsilon\}-p_n\{\upsilon+1\}\right) \\ & \geq L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right) \\ & \quad+\sum_{n=1}^N\left[\left(\nabla_{p_n} L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)\right)^T \frac{1}{\varsigma}\left(\nabla_{p_n} f_n\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right)-\nabla_{p_n} f_n\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)\right)\right] \\ & = L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)+\frac{1}{\varsigma}\left\|\left(\nabla_{p_n} L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)\right)\right\|_2^2 \\ & \geq L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)+\frac{2\varphi}{\varsigma}\left[L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)-L\left(p_n^*, \Lambda_n^*\right)\right] \end{aligned} \tag{43}$$

Then, based on Lemma 1 and Lemma 2, (43), as shown at the top of this page, could be derived.

Subtracting $L\left(p_n^*, \Lambda_n^*\right)$ on (43), we have

$$\begin{aligned} & L\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right)-L\left(p_n^*, \Lambda_n^*\right) \\ & \geq\left(1+\frac{2\varphi}{\varsigma}\right)\left[L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)-L\left(p_n^*, \Lambda_n^*\right)\right] \end{aligned} \tag{44}$$

This shows the step size between two steps υ and $\upsilon+1$. We denote

$$\begin{aligned} \psi & =\frac{L\left(p_n\{\upsilon+1\}, \Lambda_n\{\upsilon+1\}\right)-L\left(p_n^*, \Lambda_n^*\right)}{L\left(p_n\{\upsilon\}, \Lambda_n\{\upsilon\}\right)-L\left(p_n^*, \Lambda_n^*\right)} \\ & =\frac{1}{\left(1+\frac{2\varphi}{\varsigma}\right)} \end{aligned} \tag{45}$$

That is to say, $\psi^{\Upsilon}\left[L\left(p_n\{0\}, \Lambda_n\{0\}\right)-L\left(p_n^*, \Lambda_n^*\right)\right] \leq \varepsilon_0$ is guaranteed, where $L\left(p_n\{0\}, \Lambda_n\{0\}\right)$ is the current optimal point at step 0. In particular, we draw the relationship between the step Υ and ε_0 as below

$$\Upsilon \leq \frac{\log \left[L\left(p_n\{0\}, \Lambda_n\{0\}\right)-L\left(p_n^*, \Lambda_n^*\right)\right]}{\varepsilon_0 \log \left({}^1/_{\psi}\right)} \tag{46}$$

The log-linear convergence of Υ indicates that there exists a constant Υ_0, where $\Upsilon \leq \frac{\Upsilon_0}{\varepsilon_0}$. That is to say, the price model converges linearly.

REFERENCES

[1] L. P. Qian, Y. Wu, B. Ji, L. Huang, and D. H. K. Tsang, "HybridIoT: Integration of hierarchical multiple access and computation offloading for IoT-based smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 6–13, Mar. 2019.
[2] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-reliability and low-latency wireless communication for Internet of Things: Challenges, fundamentals, and enabling technologies," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7946–7970, Oct. 2019.
[3] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017.
[4] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for edge computing-empowered industrial IoT," *IEEE Internet Things J.*, early access, Dec. 31, 2020, doi: 10.1109/JIOT.2019.2963371.
[5] Z. Zhou, Y. Guo, Y. He, X. Zhao, and W. M. Bazzi, "Access control and resource allocation for M2M communications in industrial automation," *IEEE Trans Ind. Informat.*, vol. 15, no. 5, pp. 3093–3103, May 2019.
[6] Y. Shi, J. Zhang, K. B. Letaief, B. Bai, and W. Chen, "Large-scale convex optimization for ultra-dense cloud-RAN," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 84–91, Jun. 2015.
[7] A. Abdelnasser, E. Hossain, and D. I. Kim, "Tier-aware resource allocation in OFDMA macrocell-small cell networks," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 695–710, Mar. 2015.
[8] B. Zhuang, D. Guo, E. Wei, and M. L. Honig, "Large-scale spectrum allocation for cellular networks via sparse optimization," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5470–5483, Oct. 2018.
[9] Y. Shi, J. Zhang, and K. B. Letaief, "Group sparse beamforming for green cloud-RAN," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2809–2823, May 2014.
[10] N. Mokari, F. Alavi, S. Parsaeefard, and T. Le-Ngoc, "Limited-feedback resource allocation in heterogeneous cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2509–2521, Apr. 2016.
[11] Y. Shi, J. Zhang, B. O'Donoghue, and K. B. Letaief, "Large-scale convex optimization for dense wireless cooperative networks," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4729–4743, Sep. 2015.
[12] B. He, M. Tao, and X. Yuan, "Alternating direction method with Gaussian back substitution for separable convex programming," *SIAM J. Optim.*, vol. 22, no. 2, pp. 313–340, Jan. 2012.
[13] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $o(1/k)$ convergence," *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, May 2017.
[14] T. Chang, M. Hong, W. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization—Part I: Algorithm and convergence analysis," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3118–3130, Jun. 2016.
[15] G. Zhang, Y. Chen, Z. Shen, and L. Wang, "Distributed energy management for multiuser mobile-edge computing systems with energy harvesting devices and QoS constraints," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4035–4048, Jun. 2019.
[16] C. Liang, F. R. Yu, H. Yao, and Z. Han, "Virtual resource allocation in information-centric wireless networks with virtualization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9902–9914, Dec. 2016.
[17] F. Wang, Y. Xu, L. Zhu, X. Du, and M. Guizani, "LAMANCO: A lightweight anonymous mutual authentication scheme for N-times computing offloading in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4462–4471, Jun. 2019.
[18] T.-D. Lee, B. M. Lee, and W. Noh, "Hierarchical cloud computing architecture for context-aware IoT services," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 222–230, May 2018.
[19] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.
[20] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Industrial IoT data scheduling based on hierarchical fog computing: A key for enabling smart factory," *IEEE Trans Ind. Informat.*, vol. 14, no. 10, pp. 4590–4602, Oct. 2018.
[21] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, "A stackelberg game approach to proactive caching in large-scale mobile edge networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5198–5211, Aug. 2018.

[22] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 428–438, Jun. 2016.
[23] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 485–488, Oct. 2014.
[24] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software defined machine-to-machine communication for smart energy management," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 52–60, Oct. 2017.
[25] V. Aggarwal, M. R. Bell, A. Elgabli, X. Wang, and S. Zhong, "Joint energy-bandwidth allocation for multiuser channels with cooperating hybrid energy nodes," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9880–9889, Nov. 2017.
[26] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, "Game theory for big data processing: Multileader multifollower game-based ADMM," *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 3933–3945, Aug. 2018.
[27] Z. Zheng, L. Song, and Z. Han, "Bridge the gap between ADMM and stackelberg game: Incentive mechanism design for big data networks," *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 191–195, Feb. 2017.

YANHUA HE (Student Member, IEEE) received the B.Eng. degree in communication engineering from North China Electric Power University (NCEPU), in 2015, where she is currently pursuing the Ph.D./master's degree. Her research interests cover the resource allocation technology and green communication technology for the Internet of Vehicles (V2X), the Internet of Things (IoT), and heterogeneous networks for 5G communications, which involve the convex optimization algorithms, matching algorithms, queuing algorithms, and distributed parallel algorithms.

SUNXUAN ZHANG is currently pursuing the degree in communication engineering, with North China Electric Power University (NCEPU). His research interests include the Internet of Things in power systems (IoTiPS), the Industrial Internet of Things (IIoT), machine-to-machine communication, and unlimited resource allocation.

LIANGRUI TANG received the Ph.D. degree in Communication and Information System from the Beijing University of Posts and Telecommunications. He is currently a Professor with the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, focusing on the research of communication in power systems, wireless communications, and optical network communication.

YUN REN received the B.S. and master's degrees from the North China Electric Power University (NCEPU), Beijing, China, in 2014 and 2017, respectively. He is currently a member of State Grid Zhejiang Electric Power Company Ningbo Bureau, Zhejiang, China. His research interests include wireless communication networks and the wireless network energy saving technologies.