This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

Pricing and Resource Allocation Optimization for IoT Fog Computing and NFV: An EPEC and Matching Based Perspective

Neetu Raveendran, *Member, IEEE,* Huaqing Zhang, *Member, IEEE,*
Lingyang Song, *Fellow, IEEE,* Li-Chun Wang, *Fellow, IEEE,*
Choong Seon Hong, *Senior Member, IEEE,* and Zhu Han, *Fellow, IEEE*

Abstract—The number of devices connected to the Internet of Things (IoT) is growing at an enormous rate globally. In the next generation networks, distributed fog computing deployments at the network edge can provide computing resources to the users, especially for latency-sensitive applications. Further, the heterogeneous needs of the fifth generation (5G) networks demand the virtualization of network functions, termed as Network Function Virtualization (NFV). Therefore, an integrated NFV and fog computing resource allocation framework for IoT is of prime importance. Accordingly, in this paper, we model the interactions between the Data Service Operators (DSOs) and the Authorized Data Service Subscribers (ADSSs) as an Equilibrium Problem with Equilibrium Constraints (EPEC), and utilize the Alternating Direction Method of Multipliers (ADMM) as a large-scale optimization tool to obtain solutions. This results in the optimization of resource pricing for the DSOs and the amount of resources to be purchased by the ADSSs. Moreover, we propose a many-to-many matching based model to allocate the Fog Node (FN) resources according to the VNF resource requirements of the ADSSs. Simulation results show the effectiveness of our proposed approach in achieving efficient resource allocation in NFV enabled IoT fog computing.

Index Terms—Fog computing, NFV, IoT, resource allocation, EPEC, ADMM, many-to-many matching.

1 Introduction

The advent of the *Internet of Things (IoT)*[1] brings about massive internetworking of devices that we use in our everyday lives. This gives rise to the need for storing and processing tremendous amounts of data efficiently [1]. The handling of such a large amount of data can be realized by *cloud computing* [2], by providing the required resources for the users to access various applications on demand. Additionally, the heterogeneity of applications in IoT calls for the virtualization of wireless networks, which leads to better flexibility and management through the abstraction and sharing of resources [3]. *Wireless network virtualization* involves the sharing of the physical substrate network by multiple virtual networks, and to facilitate this, both spectrum and infrastructure resources are isolated and split into *slices* [4], [5].

TABLE 1: Acronyms used and their expansions

Acronym	Expansion
IoT	Internet of Things
5G	Fifth generation
NFV	Network Function Virtualization
VNFs	Virtual Network Functions
CNs	Cloud Networks
DSOs	Data Service Operators
ADSSs	Authorized Data Service Subscribers
FNs	Fog Nodes
EPEC	Equilibrium Problem with Equilibrium Constraints
ADMM	Alternating Direction Method of Multipliers
MVNO	Mobile Virtual Network Operator
NFVO	NFV Orchestrator
VM	Virtual Machine
CRBs	Computing Resource Blocks

Virtualization is one of the key driving forces of the *fifth generation (5G)* of mobile networks, which aims at delivering extremely high capacity, low latency, and high device density per area. Specifically, *Network Function Virtualization (NFV)* is a paradigm which decouples the physical network infrastructure from the network functions that run on it [6]. The different services are disintegrated into *Virtual Network Functions (VNFs)*, and are placed on top of a virtualization

- *N. Raveendran is with PoLTE, Addison, TX, USA. Email: raveendranneetu@gmail.com.*
- *H. Zhang is with Google LLC, Mountain View, CA, USA. Email: markasjunior@gmail.com.*
- *L. Song is with the School of Electrical Engineering and Computer Science, Peking University, Beijing, China. Email: lingyang.song@pku.edu.cn.*
- *L.-C. Wang is with the Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan. Email: lichun@cc.nctu.edu.tw.*
- *C. S. Hong is with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. Email: cshong@khu.ac.kr.*
- *Z. Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. Z. Han is also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea. Email: zhan2@uh.edu.*

[1] The acronyms used in this paper and their expansions are listed in Table 1.

platform as software components on Cloud Networks (CNs) [7], [8]. The increase in the demand for storage and processing capabilities due to the provisioning of 5G services can be addressed by similar convergence of network and cloud infrastructures [9].

However, the processing requirements vary according to the applications, with some applications demanding faster processing speeds than others [10]. Traditionally, the large-scale data centers built by the *Data Service Operators (DSOs)* to meet the processing needs of the *Authorized Data Service Subscribers (ADSSs)* are far from the ADSSs. In the light of the fast processing demands of next generation networks, computation resources are being moved to the edge of the network. This concept is known as *fog computing* [11], in which a number of small-scale but flexible computing devices called *Fog Nodes (FNs)* are deployed close to the ADSSs. These micro clouds are also known as *edge clouds* or *cloudlets*, as they have lesser computing resources compared to data center based clouds and are deployed at the network edge [12]. Due to their proximity to the ADSSs, the FNs can provide data services with low latency and low transmission costs [13].

Considering the heterogeneity and complexity of IoT applications, an integration of the fog computing technology with NFV is inevitable for rendering computation flexibility and scalability in next generation networks. As a result, an efficient resource allocation solution for a fog enabled NFV platform needs to effectively model the interactions between the different sets of entities: DSOs, ADSSs, and FNs, as well as enable the DSOs to allocate resources from the FNs as per the VNF requirements of the ADSSs.

The DSOs and ADSSs in a typical fog computing scenario are autonomous entities that try to maximize their own profits. The DSOs try to allocate resources from the FNs at prices which favor them, and the ADSSs purchase these resources according to their own benefits. However, the maximization of profit for one DSO might affect the profits of other DSOs. Also, if one ADSS tries to maximize the amount of purchased resources at a given price, it might affect the amount of computing resources available to the other ADSSs. Therefore, in order to reach a stable and social optimum, we need to model the competition among them, and find an equilibrium solution.

Modeling this competition results in an *Equilibrium Problem with Equilibrium Constraints (EPEC)*, which is a hierarchical optimization problem with equilibria at two levels [14]. Due to the conflicts between them and amongst themselves, there exist equilibrium criteria at both the level of the DSOs and at the level of the ADSSs. In order to balance these conflicting objectives, we use an incentive mechanism as in [15], and then perform the optimization of their utilities. The *Alternating Direction Method of Multipliers (ADMM)*, which is considered an efficient tool for large-scale optimization is adopted here, due to its decomposition and fast convergence properties [16].

A large-scale fog computing optimization framework to achieve this is proposed in [17], which formulates the interactions between the DSOs and ADSSs as an EPEC. It is then solved using an ADMM based algorithm, which provides the optimal values of the resource prices to be set by the DSOs, and the optimal amount of resources to be purchased by the ADSSs, resulting in the simultaneous optimization of profits for both sets of entities. However, a fog computing resource allocation framework for the next generation networks is incomplete without the dimension of virtualization. We need to efficiently allocate the resources from FNs as per the resource needs of various VNFs initiated at the ADSSs, by considering the localized conditions.

Accordingly, in this paper, we propose a resource allocation framework for next generation networks, by keeping in mind the flexibility of fog computing and the scalability of NFV. On that account, we extend the framework proposed in [17] for large-scale fog computing, to integrate NFV, by proposing a matching theory based algorithm for the DSOs to allocate resources from the FNs as per the VNF resource needs of the ADSSs. Taking all of this into account, we state the objectives and contributions of this paper as below.

- We model the hierarchical competitions between the DSOs and the ADSSs in an NFV integrated IoT fog computing scenario as an EPEC. We use an incentive mechanism to balance the conflicting objectives of both sets of entities, i.e., the DSOs which sell resources from the FNs to the ADSSs, and the ADSSs which purchase these resources as per the demands of the VNFs that serve them.
- An ADMM based algorithm is invoked to solve the EPEC and obtain the optimal values of the resource prices to be set by the DSOs, as well as the optimal amount of resources to be purchased by the ADSSs, resulting in profit optimization for both.
- Further, we procure the resource requirements for different VNFs to be deployed based on the resource requirements of the ADSSs. This is utilized in a *many-to-many matching* algorithm, which efficiently allocates the computing resources of the many FNs according to the resource requirements of the many VNFs, in a distributed manner.
- The effectiveness of the proposed framework is then demonstrated through simulations. The simulation results show that the proposed ADMM based EPEC algorithm converges within a few iterations to give optimum pricing for the DSOs and optimum resource allocation for the ADSSs. The proposed many-to-many matching algorithm is observed to outperform the centralized approach in terms of the costs of the FN resources allocated.

The remainder of this paper is arranged as follows. We discuss some of the relevant previous works in Section 2. In Section 3, we introduce the system model, and formulate the problem in Section 4. We analyze the proposed framework in Section 5, where firstly, we introduce the concept of ADMM in Section 5.1. Secondly, we discuss the design of the incentive function in Section 5.2, and then the ADMM based EPEC algorithm is discussed in detail in Section 5.3. The many-to-many matching algorithm for VNF resource allocation is discussed in detail in Section 5.4. We discuss the performance of our model through simulation results in Section 6. Finally, the paper is concluded and some future research directions are provided in Section 7.

2 Literature Review

Wireless network virtualization resource allocation for next generation networks has been widely discussed in the literature [18], [19], [20], [21]. [22] discusses the dynamic allocation of resources to different network slices in order to maximize user satisfaction. [23] proposes an information-centric wireless network virtualization architecture for 5G mobile networks, where the end-to-end network performance is improved by integrating virtualization with *Information-Centric Networking (ICN)*. A user mobility and service usage oriented approach for virtual wireless networks is discussed in [24]. [25] proposes a three-sided matching based framework for wireless network virtualization resource allocation considering the spectrum and infrastructure resources and mobile users. The research works in NFV generally deal with either determining the optimal number of required VNFs or the placement of VNFs in CNs, and [7] proposes a solution to address both jointly. [26] proposes a matching based framework for NFV resource allocation, by jointly considering both the user requirements for VNFs as well as their placements in different CNs.

The management of resources in fog computing is challenging due to a large number of FN deployments, and is extensively discussed in research areas [13]. A multi-dimensional framework has been proposed in [27], where a *Quality of Service (QoS)* consistent contract providing a comprehensive payment plan to the FNs, revenue maximization of the network operators, and incentives to the FNs has been evaluated. A mathematical framework for service-oriented heterogeneous resource sharing has been proposed in [28], and [29] proposes a *Distributed Dataflow (DDF)* programming model that coordinates the resources distributed across hosts in fog computing. [30] proposes a matching game based joint radio and computational resource allocation problem for optimizing system performance and improving user satisfaction in IoT fog computing. [31] investigates the formation of stable coalitions among *Fog Infrastructure Providers (FIPs)*, and proposes a mathematical model for profit maximization in order to allocate IoT applications to sets of FIPs.

There are a few recent works that integrate fog computing and virtualization in IoT. [32] proposes *Virtual Fog*, which is a complete layered framework for IoT and connects the layers from fog computing through virtualization. A dynamic resource allocation framework for NFV enabled *Mobile edge-cloud (MEC)* is discussed in [12], in which both low latency requirements and MEC cost efficiency are addressed. [9] demonstrates three use cases of an integrated cloud/fog and heterogeneous networks orchestration through a 5G NFV experimental platform, and performs testing of end-to-end IoT and mobile services. Even though these works perform the indispensable integration of NFV and fog computing for IoT, a resource allocation framework for IoT fog computing and NFV, by taking into consideration the DSOs, ADSSs, and FNs, and also the resource requirements of the various VNFs, has not been proposed.

The fog computing scenario considered in [17] addresses the competition among multiple DSOs and multiple ADSSs, which results in an EPEC, as opposed to related previous works. As discussed in [33], [34], algorithms aimed specifically at solving EPECs with a published convergence analysis have not been developed. [33] mentions that engineering approaches to solve EPECs are *diagonalization methods* that are based on *nonlinear programs (NLPs)* or approaches that solve each *Mathematical Problem with Equilibrium Constraints (MPEC)* individually until an equilibrium is reached. [33] also proposes a novel sequential nonlinear complementarity (SNCP) algorithm for solving EPECs. In this paper, the optimization of the EPEC scenario consisting of a large number of entities is handled by the convergence properties of ADMM [35], [36], which is a powerful tool for large-scale optimization.

In order to bring in the NFV perspective here, we have to determine the resource requirements for various VNFs that serve the ADSSs, which are to be deployed in the FNs operated by different DSOs. To that end, we need to model the localized preferences of the FNs and the VNF resource requirements in a distributed manner. A promising candidate here is matching theory, which has gained popularity in recent years as an efficient distributed framework, which considers the localized preferences of different sets of entities [37].

The formation of mutually beneficial relationships between different sets of entities forms the basis for matching theory [38], [39], [40], and it overcomes certain limitations of optimization and game theory [41]. [42] discusses in detail the advantages of matching theory in wireless resource allocation. [37] highlights how the distributed nature of matching takes into account the preferences of users on resources, and vice versa, based on localized information. It is also emphasized in [43] that there exists at least one stable matching for every resource allocation problem, by means of the *deferred acceptance* method.

Considering a large number of FN deployments and VNF initiations, and also the fact that different instances of the same VNF initiated to serve different ADSSs, can be deployed in the same FN for ease of management (and also that many FNs can together host a single VNF instance), in this paper, we propose a many-to-many matching [44] based algorithm for NFV enabled IoT fog computing, to perform VNF resource allocation in FNs.

3 System Model

We consider an IoT fog computing scenario consisting of multiple DSOs, ADSSs, and FNs, as shown in Fig. 1. As mentioned before, unlike the massive data centers which are usually located far from the ADSSs, the FNs are deployed closer to the ADSSs, which helps to reduce the service latency and congestion by computation offloading. The ADSSs request for computing resources from the DSOs, and the DSOs serve the ADSSs by allocating computing resources from the FNs operated by them. At the same time, the traditional practice in an NFV scenario is that the *Mobile Virtual Network Operator (MVNO)* communicates the resource needs based on the requirements of the users to the *NFV Orchestrator (NFVO)*. Then, the NFVO translates these resource needs into software component requirements that need to be initiated in the network [26].

In an NFV enabled IoT fog computing scenario, these software components would be the *Virtual Machine (VM)*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

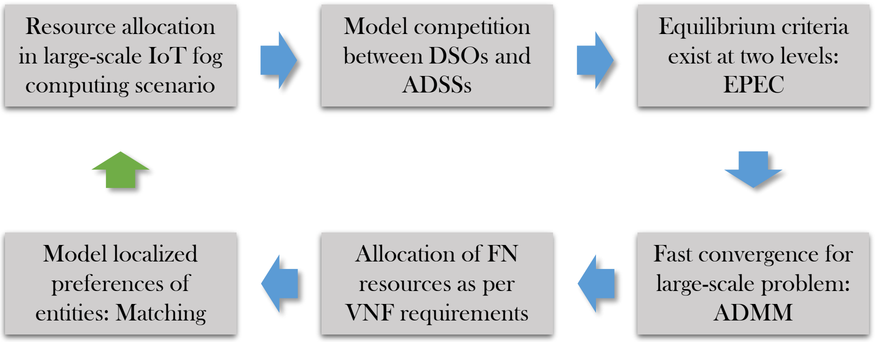
Resource allocation in large-scale IoT fog computing scenario
Model competition between DSOs and ADSSs
Equilibrium criteria exist at two levels: EPEC
Fast convergence for large-scale problem: ADMM
Allocation of FN resources as per VNF requirements
Model localized preferences of entities: Matching

Fig. 2: Proposed NFV resource allocation framework for fog computing in IoT.

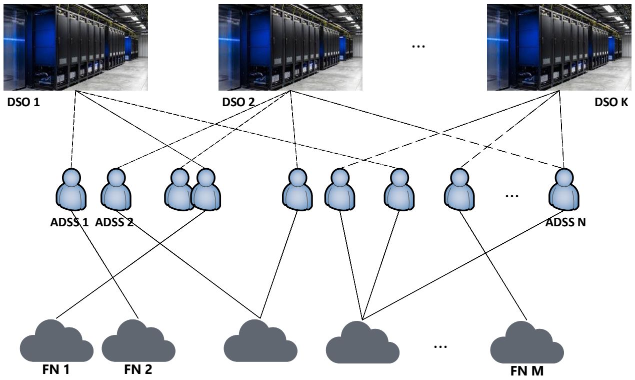

Fig. 1: System architecture.

resources available in the distributed network of FNs. Once the VNF requirements of each ADSS are known to the DSOs, they can allocate VM resources from the FNs operated by them. The VM resources in the FNs can then host the allocated instances of VNFs. Fig. 2 summarizes the proposed framework. In this paper, we consider the NFVO to be a centralized entity that coordinates the proposed distributed resource allocation schemes.

Let us consider a network with K DSOs, N ADSSs, and M FNs. We denote the computing resources from the FNs in terms of *Computing Resource Blocks (CRBs)*. The price for one CRB set by DSO i for ADSS j is denoted by $\{\theta_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$, and $\boldsymbol{\theta_i}$ is the pricing profile for DSO i. The number of CRBs purchased from DSO i by ADSS j is denoted by $\{x_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$.

Therefore, according to the profits and costs of DSOs, we express the utility function of DSO i, $\forall i \in \{1, 2, \ldots, K\}$, as

$$P_i(\boldsymbol{\theta_i}) = \sum_{j=1}^{N} U_{i,j}(\theta_{i,j}), \tag{1}$$

where

$$U_{i,j}(\theta_{i,j}) = R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}, \tag{2}$$

$$R_{i,j}(\theta_{i,j}) = x_{i,j}\theta_{i,j} \tag{3}$$

is the revenue from the computing resources provided to ADSS j by DSO i, where $x_{i,j}$ is the number of CRBs purchased from DSO i by ADSS j, and $\theta_{i,j}$ is the price for one CRB purchased by ADSS j from DSO i.

$$D_{i,j} = n_{i,j} + q_{i,j} \tag{4}$$

is the cost due to service delay. Here,

$$n_{i,j} = \gamma d_{i,j} \tag{5}$$

is the cost incurred by the network delay, which is the delay from the physical service provider (i.e., FN) to ADSS j. Consider its value to be a linear function of the distance from the physical service provider to ADSS j, $d_{i,j}$, and γ is the cost per unit distance.

We consider the workload of ADSS j to follow a *Poisson* process, as it is one of the classical models used for traffic arrival in communication networks, and it can effectively model the random workload process here, where the points are stochastically independent to each other. If we were to consider a non-Poisson model for the ADSS workload arrival, then the property of *independent increment* would no longer hold. As per this property, for mutually disjoint intervals $(t_1, t_1^{'}], (t_2, t_2^{'}], \ldots, (t_n, t_n^{'}]$, the random variables $A(t_l^{'}) - A(t_l), \forall l \in \{1, 2, \ldots, n\}$ are mutually independent [45]. We consider the workload rate of ADSS j to be w_j, and the queue length as the workload rate/computing service rate, and as a result, we get the cost incurred by the queuing delay at the servers, $q_{i,j}$ as

$$q_{i,j} = \kappa \frac{w_j}{\mu x_{i,j}}, \tag{6}$$

where κ is the cost per unit queue length, and each CRB can provide computing service at the rate of μ.

$$O_{i,j} = x_{i,j}\eta_{i,j} \tag{7}$$

is the operational and measurement cost for the resources provided by the FNs to ADSS j, thus helping DSO i in offloading. Here, $\eta_{i,j}$ is the price set by the FNs helping DSO i serve ADSS j.

As mentioned before, the number of CRBs purchased from DSO i by ADSS j is denoted by $\{x_{i,j}|i = 1,2,...,K; j = 1,2,...,N\}$, and $\mathbf{x}_j$ is the CRB purchase profile of ADSS j. According to the profits and costs of the ADSSs, we express the utility function of ADSS j, $\forall j \in \{1,2,\ldots,N\}$, as

$$Q_j(\mathbf{x}_j) = \sum_{i=1}^{K} W_{i,j}(x_{i,j}), \tag{8}$$

where

$$W_{i,j}(x_{i,j}) = T_{i,j} - D_{i,j} - R_{i,j}(x_{i,j}). \tag{9}$$

Here,

$$T_{i,j} = \beta_j w_j \tag{10}$$

is the revenue obtained by ADSS j from its workload data, where β_j is the revenue obtained by ADSS j per unit workload rate. $D_{i,j}$ is the cost due to service delay, similar to the case of the DSOs.

$$R_{i,j}(x_{i,j}) = x_{i,j}\theta_{i,j} \tag{11}$$

is the cost of the computing resources provided to ADSS j by DSO i, which is the same as the revenue obtained by DSO i from the computing resources provided to ADSS j, and hence, we have used the same notation, $R_{i,j}$.

4 Problem Formulation

As mentioned before, the DSOs and the ADSSs are assumed to be autonomous entities, that aim to maximize their own profits. However, the maximization of $P_i(\boldsymbol{\theta_i})$ for one DSO may affect the utilities of other DSOs and the ADSSs, and similarly, the maximization of $Q_j(\mathbf{x_j})$ for one ADSS may affect the utilities of other ADSSs and the DSOs. Also, the optimization of the utility functions of the ADSSs should be performed in such a way that the optimization of the utility functions of the DSOs are not affected. When the number of DSOs and ADSSs are large as in a typical IoT fog computing scenario, a centralized optimization of the utilities of all the DSOs as in (1), and those of all the ADSSs as in (8) simultaneously, is a difficult task.

From the utility function discussed in the above section, we can express the optimization problem of DSO i as

$$\max_{\theta_\mathbf{i}} P_i = \sum_{j=1}^{N} [R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}] \tag{12}$$

$$s.t. \begin{cases} \sum\limits_{j=1}^{N} A_{i,j}\theta_{i,j} = B_i, \\ D_{i,j} \leq D_{th}, \end{cases} \tag{13}$$

where $\boldsymbol{\theta_i} = (\theta_{i,1}, \theta_{i,2}, ..., \theta_{i,N})$ is the row vector that represents the prices set by DSO i for each of the N ADSSs. The first linear constraint for DSO i in (13) indicates the limit on the total price per CRB offered by DSO i, where all $\{A_{i,j}|j = 1,2,...,N\}$ and B_i are real, scalar constants. D_{th} in the second constraint denotes the upper bound for the cost of service delay between DSO i and ADSS j.

We can also express the optimization problem of ADSS j as

$$\max_{\mathbf{x_j}} Q_j = \sum_{i=1}^{K} [T_{i,j} - D_{i,j} - R_{i,j}(x_{i,j})] \tag{14}$$

$$s.t. \begin{cases} \sum\limits_{i=1}^{K} X_{i,j}x_{i,j} = Y_j, \\ D_{i,j} \leq D_{th}, \end{cases} \tag{15}$$

where $\mathbf{x_j} = [x_{1,j}, x_{2,j}, ..., x_{K,j}]^T$ is the vector that represents the resources purchased by ADSS j from each of the K DSOs. The first linear constraint for ADSS j in (15) indicates the total resource requirement of ADSS j in terms of the number of CRBs purchased from all the DSOs, where all $\{X_{i,j}|i = 1,2,...,K\}$ and Y_j are real, scalar constants. The second constraint is similar to the one in the optimization problem of DSO i.

Since $x_{i,j}$ denotes the number of CRBs purchased from DSO i by ADSS j, the values in $\{x_{i,j}|i = 1,2,...,K; j = 1,2,...,N\}$ are decided by the ADSSs. Hence, this matrix would consist of values which are the optimal values of $x_{i,j}$, so as to maximize the utilities of the ADSSs as in (14), rather than the optimal values of $x_{i,j}$, so as to maximize the utilities of the DSOs as in (12). Therefore the DSOs need to provide incentives to the ADSSs, in order to make the ADSSs choose values in $\{x_{i,j}|i = 1,2,...,K; j = 1,2,...,N\}$ favoring the DSOs. To this end, we can formulate the problem as an incentive mechanism design, which can lead to an optimum result as per the utilities of the DSOs in (12), while simultaneously considering the utilities of the ADSSs in (14).

Here, we can consider $\theta_{i,j}$ to be the incentive factor provided by DSO i to ADSS j, as the DSO can influence the value of $x_{i,j}$ by setting the price at a certain $\theta_{i,j}$. By controlling the incentive factor $\{\theta_{i,j}|i = 1,2,...,K; j = 1,2,...,N\}$, DSO i can get each of the ADSSs to choose the values of $x_{i,j}$ such that its profit, $P_i(\theta_{i,*})$ is maximized. $\boldsymbol{\theta_j} = (\theta_{1,j}, \theta_{2,j}, ..., \theta_{K,j})^T$ is the vector of incentive factors for ADSS j. We can use this to design an incentive function $\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$, which indicates the interactions between the DSOs and ADSS j.

In summary, the DSOs' optimization problem can be formulated as:

$$\max_{\theta_\mathbf{i}} P_i = \sum_{j=1}^{N} [R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}]$$

$$s.t. \begin{cases} \sum\limits_{j=1}^{N} A_{i,j}\theta_{i,j} = B_i, \\ D_{i,j} \leq D_{th}, \\ \mathbf{x_j} = \arg\max \Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j}), \\ \qquad s.t. \begin{cases} \sum\limits_{i=1}^{K} X_{i,j}x_{i,j} = Y_j, \\ D_{i,j} \leq D_{th}, \end{cases} \end{cases} \tag{16}$$

$\forall i \in \{1,2,\ldots,K\}$, and $\forall j \in \{1,2,\ldots,N\}$.

This is an example of an EPEC, which is a hierarchical optimization problem that contains equilibrium problems at both the upper and lower levels [14]. That is to say, there exist equilibrium criteria at the upper level as well, rather than just minimizing the real-valued functions subject to equilibrium constraints. In our scenario, both the DSOs as well as the ADSSs have a set of equilibrium constraints, as shown in (16). As there are two levels of entities with equilibrium constraints, a centralized solution that is feasible for everyone is difficult. Here, the DSOs are at advantage, as they make the first move by declaring the prices for the computing resources they provide. They can predict the amount

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

of resources going to be purchased by the ADSSs and reach an optimal price to maximize their utilities. However, we need a solution that can optimize the utilities of the DSOs, while simultaneously considering the utilities of the ADSSs.

The ADSSs can only control the values of $x_{i,j}$, the number of CRBs purchased from the DSOs, and the DSOs can only decide the values of $\theta_{i,j}$, the incentive factors provided to the ADSSs. As the DSOs can predict the amount of resources going to be purchased, they can use the incentive factors to control the resources purchased by the ADSSs. Even though mechanisms like Stackelberg games [46] can be applied here, they work well only in scenarios with one leader and multiple followers. In our case, the coordination of multiple conflicting utilities might demand high complexity to give an optimal result. Also, the network size can practically be very large. Therefore, we need an algorithm that would converge regardless of the network size. These requirements point us to the ADMM for the above optimization problem in an IoT fog computing network. In the next section, the detailed analysis of ADMM for EPEC is considered.

5 Algorithm Analysis

Here, we firstly discuss the basic concept of the ADMM in Section 5.1. After that, we move on to the design of the incentive function in Section 5.2. That is followed in Section 5.3 by the detailed explanation of the ADMM based EPEC algorithm used to optimize the profits of both DSOs and ADSSs in IoT fog computing. Finally, the many-to-many matching algorithm for the allocation of FN resources as per the VNF requirements of the ADSSs is discussed in detail in Section 5.4.

5.1 Alternating Direction Method of Multipliers

To understand the working of the ADMM, let us consider a network with one service provider and N users, where the provider wants to maximize its utility as

$$\begin{aligned} &\max H(\mathbf{y_j}) = \sum_{j=1}^{N} h_j(y_j) \\ &s.t. \sum_{j=1}^{N} C_j y_j - D = 0, \end{aligned} \tag{17}$$

where each $h_j(y_j)$ is a strongly convex function, y_j is a real, scalar variable, and C_j and D are given real, scalar constants [15].

Here, the values of $\mathbf{y_j}$ can be updated by the provider as

$$\mathbf{y_j}(t+1) = \arg\max\left(H\left(y_j\right)\right) + \sum_{j=1}^{N} \lambda_j\left(t\right) C_j y_j + \Psi, \tag{18}$$

where

$$\Psi = \frac{\rho}{2}\sum_{j=1}^{N} \|C_j y_j - D\|_2^2. \tag{19}$$

Here, $\|.\|_2$ denotes the Frobenius norm, $\rho > 0$ is a damping factor, and t is the iteration step index [15]. λ is the dual variable, and it is updated as

$$\lambda_j\left(t+1\right) = \lambda_j\left(t\right) + \rho\left(\sum_{j=1}^{N} C_j y_j\left(t+1\right) - D\right). \tag{20}$$

When each $h_j(y_j)$ is strongly convex, it has been proved that the ADMM converges quickly [15]. [47] discusses the global linear convergence of the ADMM even when strong convexity is absent. Hence, it can be well used for large-scale optimization problems in big networks.

5.2 Incentive Function Design

DSO i wants to maximize its profit, P_i, by providing certain incentives to the ADSSs. Here, as the DSOs set the prices, $\theta_{i,j}$ per unit of computing resource that the ADSSs purchase, the incentive factor can be assumed to be a discount from the initial prices set by the DSOs. Let ${\theta_{i,j}}^{(p)}$ denote the price set by the DSOs at the beginning of the p^{th} iteration. Let ${\theta_{i,j}}'^{(p)}$ denote the value of price that the DSOs have evaluated at the end of the p^{th} iteration. Then the incentive factor can be expressed as

$$\delta_{i,j} = {\theta_{i,j}}^{(p)} - {\theta_{i,j}}'^{(p)}. \tag{21}$$

This would result in an incentive function expressed as

$$\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j}) = \Delta\delta_{i,j}, \tag{22}$$

where Δ is a positive scalar value, which can be the same or different for each DSO.

5.3 ADMM based EPEC in IoT Fog Computing

As mentioned before, the DSOs initially announce the prices for the CRBs that they provide. This announced set consists of prices, $\{\theta_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$ set by each DSO i for each ADSS j, that maximize the profit, $P_i(\boldsymbol{\theta_i})$ for each DSO i.

Next, we explain the ADMM based EPEC method, which is an iterative process. Each iteration of the ADMM can be explained as a two-step process as given below:

1) **Optimization Problem of the ADSSs**: Each ADSS j uses the announced prices at the start of each iteration p, ${\theta_{i,j}}^{(p)}$ to calculate the values of $\{{x_{i,j}}^{(p)} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$, the number of CRBs to be purchased from each DSO i, to maximize its profit $Q_j(\mathbf{x_j})$. Here, the superscript (p) denotes the value at the p^{th} iteration of the method. This is the *inner loop* of the ADMM. t is the iteration step index of the inner loop.
We described $\mathbf{x_j}$ in (16) as $\mathbf{x_j} = \arg\max \Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$, where $\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$ is the incentive function as described above. For each ADSS j, maximizing the incentive function is equivalent to maximizing its profit, $Q_j(\mathbf{x_j})$ to form a set of values, $\mathbf{x_j}$ which can in turn maximize the incentives provided by the DSOs. Hence, the value of $\mathbf{x_j}$ is updated at each iteration of the inner loop by ADSS j as

$$\mathbf{x_j}^{(p)}(t+1) = \arg_{\mathbf{x}_j}\max\left(Q_j\left(\mathbf{x_j}\right)\right) + \sum_{i=1}^{K} {\lambda_i}^{(p)}\left(t\right) X_{i,j} x_{i,j} + \Psi, \tag{23}$$

where

$$\Psi = \frac{\rho}{2}\sum_{i=1}^{K}\left\| \sum_{m=1, m\neq j}^{N} X_{i,m} {x_{i,m}}^{(p)}\left(\tau\right) + X_{i,j} x_{i,j} - Y_j \right\|_2^2 \tag{24}$$

and $\tau = t+1$ if $m < j$, $\tau = t$ if $m > j$. Here, $\rho > 0$ is a damping factor as mentioned above, and λ is the dual variable which is updated as

$$\lambda_i{}^{(p)}(t+1) = \lambda_i{}^{(p)}(t) + \rho\left(\sum_{i=1}^{K} X_{i,j} x_{i,j}{}^{(p)}(t+1) - Y_j\right). \tag{25}$$

At the end of the inner loop during each iteration p of the outer loop, the ADSSs arrive at a set of values, $\mathbf{x_j}$, which maximizes their profits. At the same time, these values are predicted by the DSOs, and are used to update the values of $\theta_{i,j}$.

2) **Optimization Problem of the DSOs**: The DSOs are able to predict the behaviors of ADSSs and the values of $x_{i,j}$. The DSOs then invoke ADMM as

$$\boldsymbol{\theta_i}'^{(p)}(t+1) = \arg_{\theta_i} \max\left(P_i\left(\boldsymbol{\theta_i}\right)\right) + \sum_{j=1}^{N} \lambda_j{}^{(p)}(t) A_{i,j}\theta_{i,j} + \Psi, \tag{26}$$

where

$$\Psi = \frac{\rho}{2}\sum_{j=1}^{N}\left\|\sum_{m=1, m\neq i}^{K} A_{m,j}\theta_{m,j}{}^{(p)}(\tau) + A_{i,j}\theta_{i,j} - B_i\right\|_2^2 \tag{27}$$

and $\tau = t+1$ if $m < i$, $\tau = t$ if $m > i$. Here, $\rho > 0$ is the damping factor, and λ is the dual variable which is updated as

$$\lambda_j{}^{(p)}(t+1) = \lambda_j{}^{(p)}(t) + \rho\left(\sum_{j=1}^{N} A_{i,j}\theta_{i,j}{}^{(p)}(t+1) - B_i\right). \tag{28}$$

Thus, the DSOs recalculate the values of $\theta_{i,j}$ that maximize their profits. This would result in an updated set of values for the price, $\boldsymbol{\theta_i}'^{(p)}$. As discussed in Section 5.2, $\delta_{i,j}$ denotes the difference between the updated values of price, $\theta_{i,j}'^{(p)}$ and $\theta_{i,j}{}^{(p)}$, the price at the start of iteration p. Based on this difference in prices, the DSOs calculate the incentive factor as $\Delta\delta_{i,j}$, which is a discount in the announced prices. Here, Δ is a positive scalar value, which can be the same or different for each DSO. This would result in the values of $\theta_{i,j}$ for the next iteration as

$$\theta_{i,j}{}^{(p+1)} = \theta_{i,j}{}^{(p)} \pm \Delta\delta_{i,j}. \tag{29}$$

The updated values, $\theta_{i,j}{}^{(p+1)}$ are then provided to the ADSSs for the $(p+1)^{th}$ iteration. This is the *outer loop* of the ADMM. The outer loop terminates when

$$\left\|\sum_{i=1}^{K} P_i(\boldsymbol{\theta_i}'^{(p)}) - \sum_{i=1}^{K} P_i(\boldsymbol{\theta_i}'^{(p-1)})\right\| < \varepsilon, \tag{30}$$

where ε is a pre-determined small-valued threshold. The ADMM algorithm is shown in detail in Algorithm 1.

In this paper, we assume that the communications between the set of DSOs and the set of ADSSs for the inner and outer loops of the ADMM happen through the NFVO as a coordinating entity, and the iterative process of the ADMM results in the optimization for the DSOs and the ADSSs. However, these are competing entities who do not communicate amongst themselves (i.e, there exists no communication amongst the DSOs, and also amongst the ADSSs, themselves). The communication overhead between the set of DSOs and the set of ADSSs includes the data transmission overheads between the NFVO and these entities plus the storage and processing overheads at the NFVO [48], and can be expressed as

$$\mathbf{O} = \sum_{i=1}^{K}\mathbf{O}_i + \sum_{p=1}^{\mathcal{P}}[2*\sum_{j=1}^{N}\mathbf{O}_j + 2*\sum_{i=1}^{K}\mathbf{O}_i] + \sum_{j=1}^{N}\mathbf{O}_j + \mathbf{O}_{NFVO}, \tag{31}$$

where $\mathcal{P}$ is the number of ADMM iterations, $\mathbf{O}_i$ is the communication overhead between DSO i and the NFVO, $\mathbf{O}_j$ is the communication overhead between ADSS j and the NFVO, $\mathbf{O}_{NFVO}$ is the total processing and storage overhead at the NFVO.

Lemma 1. *The utility function of DSO i as in (1) is linear.*

Proof.
$$\frac{d}{d\theta_{i,j}}P_i(\boldsymbol{\theta_i}) = \frac{d}{d\theta_{i,j}}\sum_{j=1}^{N} U_{i,j}(\theta_{i,j})$$
$$= \sum_{j=1}^{N}\frac{d}{d\theta_{i,j}}[U_{i,j}(\theta_{i,j})] = \sum_{j=1}^{N}\frac{d}{d\theta_{i,j}}[R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}]$$
$$= \sum_{j=1}^{N}\frac{d}{d\theta_{i,j}}[x_{i,j}\theta_{i,j} - \gamma d_{i,j} - \kappa\frac{w_j}{\mu x_{i,j}} - x_{i,j}\eta_{i,j}] = \sum_{j=1}^{N} x_{i,j}.$$
$$\frac{d^2}{d\theta_{i,j}^2}P_i(\boldsymbol{\theta_i}) = \frac{d}{d\theta_{i,j}}\sum_{j=1}^{N} x_{i,j} = 0, \forall\theta_{i,j}.$$ □

Hence, proved that the utility function of DSO i as in (1) is linear, which is said to be both convex and concave [49]. For the theoretical proof of convergence of the ADMM in the case of convex functions which are not strongly convex, the readers are referred to [47].

Lemma 2. *The utility function of ADSS j as in (8) is non-convex.*

Proof.
$$\frac{d}{dx_{i,j}}Q_j(\mathbf{x_j}) = \frac{d}{dx_{i,j}}\sum_{k=1}^{K} W_{i,j}(x_{i,j})$$
$$= \sum_{k=1}^{K}\frac{d}{dx_{i,j}}[W_{i,j}(x_{i,j})] = \sum_{k=1}^{K}\frac{d}{dx_{i,j}}[T_{i,j} - D_{i,j} - R_{i,j}(x_{i,j})]$$
$$= \sum_{k=1}^{K}\frac{d}{dx_{i,j}}[\beta_j w_j - \gamma d_{i,j} - \kappa\frac{w_j}{\mu x_{i,j}} - x_{i,j}\theta_{i,j}]$$
$$= \sum_{k=1}^{K}[\kappa\frac{w_j}{\mu x_{i,j}{}^2} - \theta_{i,j}].$$
$$\frac{d^2}{dx_{i,j}^2}Q_j(\mathbf{x_j}) = \sum_{k=1}^{K}\frac{d}{dx_{i,j}}[\kappa\frac{w_j}{\mu x_{i,j}{}^2} - \theta_{i,j}]$$
$$= -2\sum_{k=1}^{K}\kappa\frac{w_j}{\mu x_{i,j}{}^3} \ngeq 0, \forall x_{i,j} > 0.$$ □

Hence, proved that the utility function of ADSS j as in (8) is non-convex. For the theoretical proof of convergence of the ADMM in the case of non-convex functions, the readers are referred to [35].

5.4 Many-to-Many Matching Algorithm for VNF Resource Allocation

After the execution of the ADMM based EPEC algorithm, once the optimal values for the CRBs to be purchased by the ADSSs, and the price offered by the DSOs are obtained, the next step is the allocation of the required CRBs from the FNs, as per the VNF requirements of the ADSSs. In this

Algorithm 1 ADMM based EPEC in IoT Fog Computing

```
Input: $\{\theta_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}, p = 1$
Output: $\theta_{i,j}{}^{(opt)}, x_{i,j}{}^{(opt)}, i = 1, 2, ..., K, j = 1, 2, ..., N$
while $\left\| \sum_{i=1}^{K} P_i(\boldsymbol{\theta_i}'^{(p)}) - \sum_{i=1}^{K} P_i(\boldsymbol{\theta_i}'^{(p-1)}) \right\| \geq \varepsilon$ do
    (1) Optimization for ADSSs using ADMM (inner loop):
    ADSSs use the announced prices, $\theta_{i,j}{}^{(p)}$ to evaluate $x_{i,j}$ values, and their maximum profits, $Q_j(\mathbf{x_j})$;
    (The incentive function $\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j}) = \Delta\delta_{i,j}$, where $\delta_{i,j}$ is the incentive factor, which is a discount in the prices announced by the DSOs. For ADSS $j$, maximizing its profit $Q_j(\mathbf{x_j})$, gives a set of values $\mathbf{x_j}$, which can in turn maximize the incentives provided by the DSOs.)
    (2) Optimization for DSOs using ADMM (outer loop):
    DSOs predict the behavior of ADSSs and $x_{i,j}$ values, invoke ADMM to perform maximization of profits, resulting in new prices, $\theta_{i,j}{}'^{(p)}$, and update the prices to $\theta_{i,j}{}^{(p+1)}$ by evaluating incentives;
    (3) $p = p + 1$;
end while
Result: Optimal values of CRBs purchased, $\mathbf{x}^{(opt)} = \mathbf{x}^{(p)}$
Optimal values of price, $\boldsymbol{\theta}^{(opt)} = \boldsymbol{\theta}^{(p)}$
```

paper, we assume that only one VNF instance is initiated to serve an ADSS at a given time, and that the CRB requirement of an ADSS at a given time is the amount of VM resources required by that VNF instance. Also, as already mentioned, different instances of the same VNF initiated to serve different ADSSs, can be deployed in the same FN for ease of management, as well as many FNs can together host a single VNF instance.

In an IoT fog computing scenario, the DSOs might have different preferences on the FNs based on the resource prices set by the FNs. The DSOs would naturally prefer the FNs offering them the lowest price. Accordingly, the DSOs create their preference lists by arranging the FNs in the ascending order of their prices as

$$PL_{DSO}(i) = \eta_{k,i}, \tag{32}$$

$\forall i \in \{1, 2, \ldots, K\}$, and $\forall k \in \{1, 2, \ldots, M\}$.

Similarly, the FNs have preferences on the DSOs based on the CRB (VM resource) requirements of the ADSSs served by the DSOs. That is, the FNs consider VNF instances which require more VM resources (imply needing faster computation) as of having higher priority. Accordingly, the FNs arrange each (DSO,ADSS) pair in the descending order of the CRB (VM resource) requirement to form their preference lists as

$$PL_{FN}(k) = x_{i,j}, \tag{33}$$

$\forall i \in \{1, 2, \ldots, K\}$, $\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, M\}$. Hence, the preference list of each FN has $K \times N$ entries.

Once the preference lists are generated, a many-to-many matching can be generated between the two sets of entities. A many-to-many matching is a matching problem in which entities from a set can be assigned to multiple entities in the other set, and vice versa, based on their capacity constraints [39]. The detailed definition of the many-to-many matching model is given in Definition 1.

TABLE 2: Parameter settings for simulations

Parameter	Value
K	5
N	20
M	25
d_{ij}	$U(0, 1)\ km$
D_{th}	0.1
γ	10^{-5}
κ	10
w_j	Poisson distributed with mean = 1000 s^{-1}
μ	500 Ks^{-1}
η	$U(0, 10^{-3})$
β	$U(0, 10^{-1})$
Δ	0.5
ε for ADMM	10^{-3}
ρ for ADMM	1.5

Definition 1. ***Many-to-Many Matching Problem:*** *The objective of the many-to-many matching problem is to find a matching* $\mathcal{M} = \{(k, (i, j))\}$ *with the maximum cardinality:*

$$\textbf{\textit{max}}\ |\mathcal{M}|, \tag{34}$$

$$\textbf{\textit{s.t.}}\ \mathcal{N}(\mathcal{M}, k) \leq C^k, \tag{35}$$

$\forall k \in \{1, 2, \ldots, M\}$, $\forall i \in \{1, 2, \ldots, K\}$, and $\forall j \in \{1, 2, \ldots, N\}$. (34) represents the cardinality of the matching $\mathcal{M}$ (the number of $(k, (i, j))$ pairs in the matching). $\mathcal{N}(\mathcal{M}, k)$ represents the number of (i, j) pairs that FN k is matched to, and C^k is the maximum capacity of FN k. The many-to-many matching algorithm proposed for our NFV integrated IoT fog computing scenario is described in detail in Algorithm 2.

6 Simulation Results

This section evaluates the performance of the proposed ADMM for EPEC and many-to-many matching based framework with MATLAB. The values of the different parameters used for the simulations are shown in Table 2. The number of ADSSs, N, and FNs, M are varied in certain cases; if either of N or M values is not shown to change in the simulation figure, then it has the value as shown in Table 2. The notations in the $U(a, b)$ format in Table 2 denote random values from continuous uniform distributions in the interval (a, b).

Fig. 3 demonstrates the convergence of the ADMM based EPEC algorithm. It shows how the total profit of the DSOs, $\sum_{i=1}^{K} P_i(\boldsymbol{\theta}_i)$, behaves during the optimization using ADMM. For an error threshold of $\varepsilon = 10^{-3}$, it takes only $p = 4$ iterations for the ADMM to converge. Hence, the conflicting utilities of the DSOs have been optimized in just a few iterations. It also shows how the error value of the ADMM converges to the threshold value in those few iterations.

Fig. 4 shows the relation between the total profit of the ADSSs, $\sum_{j=1}^{N} Q_j(\mathbf{x_j})$, and the mean of the workload arrival rate of ADSSs, w_j for five cases: 200 s^{-1}, 500 s^{-1}, 1000 s^{-1},

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

Algorithm 2 Many-to-Many Matching Algorithm for VNF Resource Allocation

```
for FN $k$ do
    Construct the preference list $PL_{FN}(k)$ on all (DSO,ADSS) according to (33);
    One pointer is set as the indicator pointing at the first (DSO,ADSS) in the preference list.
end for
for DSO $i$ do
    Construct the preference list $PL_{DSO}(i)$ on all FNs according to (32);
end for
We set a $flag_k$, $\forall k \in \{1, 2, \dots, M\}$, as an indicator to show if the CRBs of FN $k$ were selected by the (DSO,ADSS) in the previous round, but not in the current round. The initial value of $flag_k = 1$;
while the pointers of all FNs have not pointed at all the (DSO,ADSS) in their preference list do
    FNs propose to (DSO,ADSS) with their prices;
    for FN $k$ who still has available CRBs do
        if $flag_k = 1$ then
            The pointer stays at the current position in the list;
        else
            The pointer jumps to the next position in the list;
        end if
        The FN proposes to the pointed (DSO,ADSS) in its preference list with its available CRBs;
        We set $flag_k = 0$;
    end for
    (DSO,ADSS) determine which FNs to select;
    for (DSO,ADSS) $x_{i,j}$ do
        if The total available number of CRBs proposed by the FNs exceed its requirements then
            (DSO,ADSS) $x_{i,j}$ selects the required number of CRBs from the FNs, and rejects the rest;
            For CRBs of the FN $k$ which are selected by the (DSO,ADSS) in the last round, but not in the current round, we set $flag_k = 1$;
        end if
    end for
end while
```

2000 s^{-1}, and 5000 s^{-1}. It can be seen that as the mean value of w_j increases from $200s^{-1}$ to $5000s^{-1}$, the total profit of the ADSSs increases.

Fig. 5 to Fig. 8 compare the performance of the proposed many-to-many matching algorithm with a centralized algorithm. The centralized algorithm is an approach in which the NFVO, which acts as a centralized entity in NFV enabled IoT fog computing, performs the resource allocation itself. The NFVO allocates the resources from the FNs according to the VNF requirements of the CRBs. Similar to the case of the many-to-many matching algorithm, we assume that the largest CRB requirement, i.e., the largest value of $x_{i,j}$ maps to the VNF instance with the highest priority. However, as opposed to the distributed approach with preference lists for the two sets of entities, in the centralized approach, the NFVO arranges the VNF instances as per their priorities (CRB requirements or $x_{i,j}$ values), and allocates them to the FNs as per their available resources. Each comparison has been executed for 50 times, and the average values are plotted here.

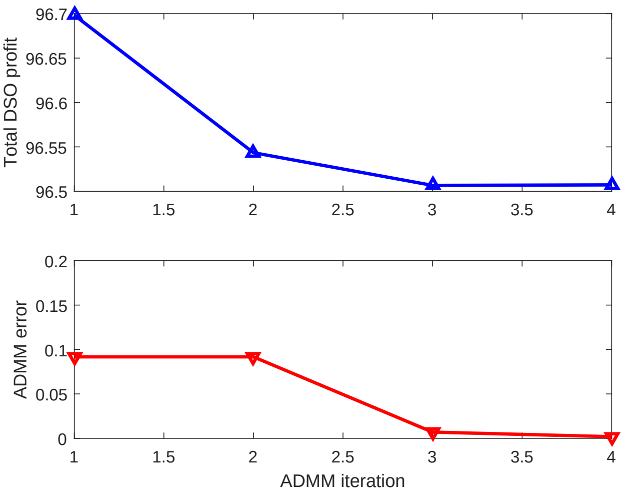

Fig. 3: Total DSO profit and ADMM error vs. number of ADMM iterations.

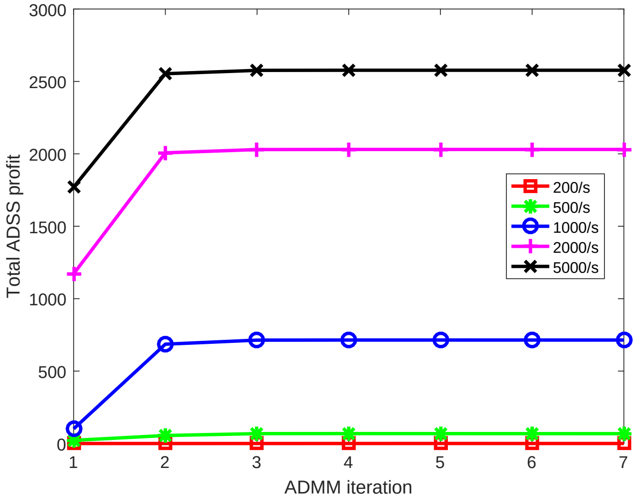

Fig. 4: Total ADSS profit vs. mean of workload rate, w_j.

Fig. 5 compares the total cost for the CRBs paid to FNs by the DSOs, between the proposed many-to-many matching algorithm and the centralized algorithm. The comparison is performed for four cases: $M = 15$, $M = 20$, $M = 25$, and $M = 30$. It can be observed that in all the four cases, the total CRB cost is lesser in the proposed approach than in the centralized approach. It can also be noted that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

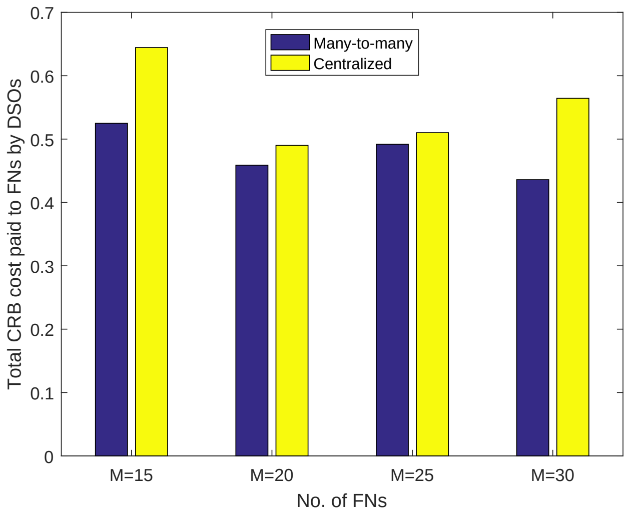

Fig. 5: The total cost for the CRBs paid to FNs by the DSOs.

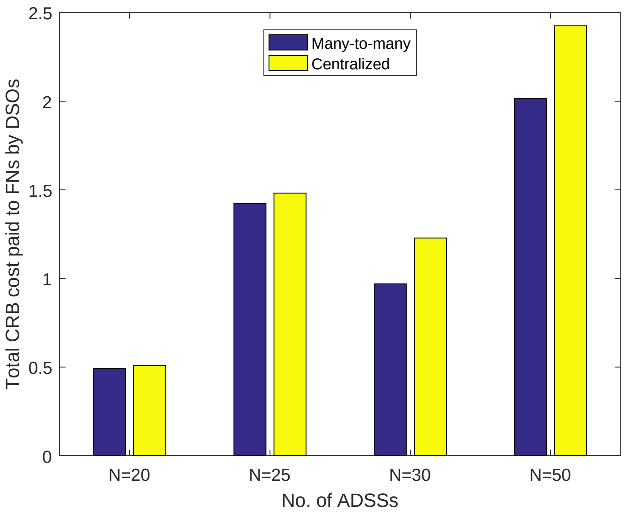

Fig. 7: The total cost for the CRBs paid to FNs by the DSOs.

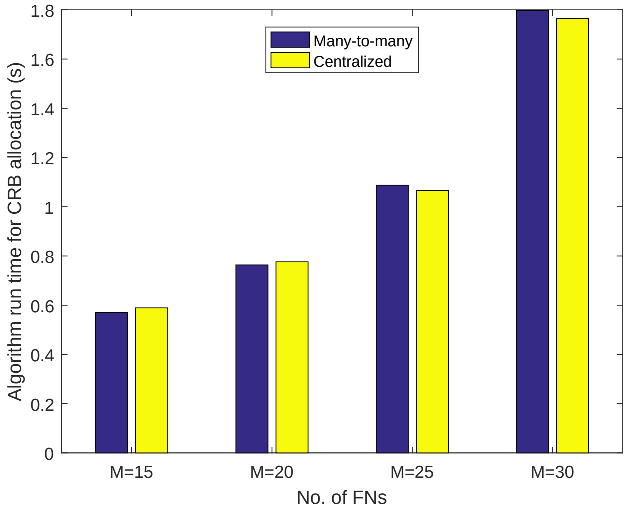

Fig. 6: Algorithm run time for the allocation of CRBs.

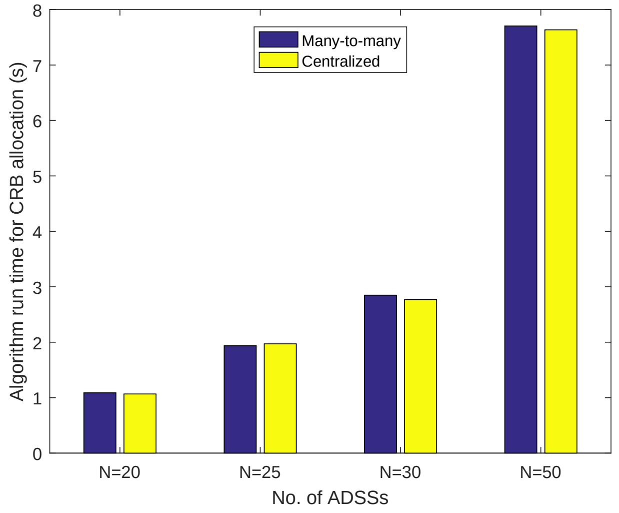

Fig. 8: Algorithm run time for the allocation of CRBs.

difference is more in the first and last cases. When $M = 20$ and $M = 25$, the number of FNs is comparable to that of the ADSSs, $N = 20$, and hence, the resource allocation might be comparable in both the approaches.

Fig. 6 compares the run times of the many-to-many matching and the centralized algorithms. Again, the comparison is performed for four cases: $M = 15$, $M = 20$, $M = 25$, and $M = 30$. It is obvious that the algorithm run times increase with the number of entities, as can be observed. The proposed algorithm has larger run times when the number of FNs increases, which is reasonable as it is a distributed approach.

Fig. 7 analyzes the total cost for the CRBs paid to FNs by the DSOs for the two approaches, similar to Fig. 5. However, the number of ADSSs is varied here to study four cases: $N = 20$, $N = 25$, $N = 30$, and $N = 50$. It can be observed again that the proposed many-to-many matching approach outperforms the centralized approach. It can also be noted that the difference is small in the first and second cases, since the number of FNs, $M = 25$, might be comparable to that of the ADSSs. However, the difference increases as the number of ADSSs grows larger.

Fig. 8 compares the algorithm run times similar to Fig. 6. Here, the number of ADSSs is varied to study four cases: $N = 20$, $N = 25$, $N = 30$, and $N = 50$. Intuitively, the algorithm run times increase with the number of entities. Again, the proposed algorithm has slightly larger run times when the number of ADSSs increases, which is reasonable as it is a distributed approach.

The system used for executing the simulations has a small scale Intel(R) Core(TM) i7 $-$ 7500U CPU with a 16 GB RAM. Therefore, the algorithm run time values shown in Fig. 6 and Fig. 8 are in hundreds of milliseconds. In a practical IoT fog computing network, the proposed many-to-many matching can be executed at the NFVO. For the matching algorithm, once the preference lists are generated, it can be executed by a centralized entity. With a large-scale processor in a practical network, the algorithm run times will decrease tremendously.

The computational complexity of Algorithm 1 is distributed among the K DSOs and the N ADSSs. For $\mathcal{P}$ iterations of the ADMM, which is linearly related to $log_{10}(\varepsilon^{-1})$ [15], the computational complexity due to the inner loops at each ADSS can be expressed as $O(\mathcal{P}K)$, and that due to the outer loops at each DSO can be expressed as $O(\mathcal{P}N)$. This is due to the computation of the optimal amount of CRBs to be purchased, by each ADSS, based on the resource prices set by the DSOs, in each iteration of the inner loop of the ADMM. Similarly, for each iteration of the outer loop of the ADMM, each DSO uses the calculated amount of CRBs to be purchased by the ADSSs, to compute the resource prices. In the case of Algorithm 2, the computational complexity can be expressed as $O(\sum_{k=1}^{M} |PL_{FN}(k)|)$, which is the sum of lengths of preference lists of the FNs. From the aforementioned complexity expressions, it is evident that the model is easily scalable as per the size and requirements for larger networks.

7 Conclusions and Future Works

In this paper, we propose a distributed resource allocation framework for an NFV integrated IoT fog computing scenario. Initially, we propose an ADMM based EPEC algorithm to model the competitions between the DSOs and the ADSSs, which provides the optimal values of the amount of resources to be purchased by the ADSSs, and the optimal values of the resource prices to be set by the DSOs. Thereafter, we invoke a many-to-many matching based algorithm to allocate the computing resources of the FNs according to the VNF resource requirements of the ADSSs. The simulation results demonstrate that the ADMM based EPEC algorithm converges quickly to give optimum results. It is also observed from the simulation results that the proposed many-to-many matching algorithm outperforms a centralized approach in terms of the cost of the FN resources. The proposed resource allocation model combining EPEC and matching can be efficiently used in NFV enabled IoT fog computing scenarios.

Even though this paper deals with allocating the resources of IoT FNs to ADSSs as per VNF resource needs, the assumption we have made is that only one VNF instance is initiated to serve an ADSS at a given time. However, in practice, there can be more than one instance of the same VNF, or instances of another VNF altogether, that are initiated to serve an ADSS at a given time. Inclusion of this aspect in our model can widen the scope of its application. Additionally, studying initiation patterns of VNF instances based on the different IoT use cases can make the model more scalable. Another research direction to explore is the application of asynchronous distributed ADMM (AD-ADMM) [50], to improve the time efficiency of distributed optimization due to the asynchrony in such large-scale heterogeneous networks. These are some reasonable future research directions for this work.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, nol. 7, pp. 1645-1660, Sep. 2013.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “Mobile Edge Computing: Survey and Research Outlook,” arXiv preprint arXiv:1701.01090, 2017.

[3] N. M. M. K. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862876, Apr. 2010.

[4] C. Liang, and F. R. Yu, “Wireless network virtualization: A survey, some research issues and challenges,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 358-380, Mar. 2015.

[5] X. Wang, P. Krishnamurthy, and D. Tipper, “Wireless network virtualization,” *in International Conference on Computing, Networking and Communications (ICNC)*, San Diego, CA, Jan. 2013.

[6] R. Mijumbi, J. Serrat, J-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, “Network function virtualization: state-of-the-art and research challenges,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 236-262, Sep. 2015.

[7] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, “Coalitional Game for the Creation of Efficient Virtual Core Network Slices in 5G Mobile Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 469-484, Mar. 2018.

[8] Y. Li, and M. Chen, “Software-Defined Network Function Virtualization: A Survey,” *IEEE Access*, vol. 3, pp. 2542-2553, Dec. 2015.

[9] R. Vilalta, A. Mayoral, R. Casellas, R. Martnez, and R. Muoz, “Experimental demonstration of distributed multi-tenant cloud/fog and heterogeneous SDN/NFV orchestration for 5G services,” *in European Conference on Networks and Communications (EuCNC)*, Athens, Greece, Jun. 2016.

[10] R. Vilalta, A. Mayoral, D. Pubill, R. Casellas, R. Martnez, J. Serra, C. Verikoukis, and R. Muoz, “End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node,” *in Optical Fiber Communications Conference and Exhibition (OFC)*, Anaheim, CA, Mar. 2016.

[11] S. Yi, C. Li, and Q. Li, “A Survey of Fog Computing: Concepts, Applications and Issues,” *in Proceedings of the Workshop on Mobile Big Data (Mobidata)*, pp. 37-42, Hangzhou, China, Jun. 2015.

[12] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, “Cost-Efficient NFV-Enabled Mobile Edge-Cloud for Low Latency Mobile Applications,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 475-488, Mar. 2018.

[13] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, “Computing Resource Allocation in Three-Tier IoT Fog Networks: a Joint Optimization Approach Combining Stackelberg Game and Matching,” *IEEE Internet Of Things Journal, Special Issue on Fog Computing in the Internet of Things*, vol. 4, no. 5, p.p. 1204-1215, Oct. 2017.

[14] B. S. Mordukhovich, “Equilibrium Problems with Equilibrium Constraints via Multiobjective Optimization,” *Research Report 17*, p. 14, 2003, *Optimization Methods and Software 19*, pp. 479-492, 2004.

[15] Z. Zheng, L. Song and Z. Han, “Bridge the Gap Between ADMM and Stackelberg Game: Incentive Mechanism Design for Big Data Networks,” *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 191-195, Feb. 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3025189, IEEE Transactions on Mobile Computing

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning,* vol. 3, no. 1, pp. 1-122, Jul. 2011.

[17] N. Raveendran, H. Zhang, Z. Zheng, L. Song, and Z. Han, "Large-Scale Fog Computing Optimization using Equilibrium Problem with Equilibrium Constraints," *in IEEE Global Communications Conference (Globecom),* Singapore, Dec. 2017.

[18] H. Zhang and Z. Han, "Distributed Resource Allocation for Network Virtualization," *in Handbook of Cognitive Radio,* Springer, editors: L. Song and W. Zhang, 2019.

[19] G. Zhang, K. Yang, H. Jiang, X. Lu, K. Xu, and L. Zhang, "Equilibrium Price and Dynamic Virtual Resource Allocation for Wireless Network Virtualization," *Mobile Networks and Applications,* vol. 22, no. 3, pp. 564-576, Jun. 2017.

[20] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management,* vol. 13, no. 3, pp. 518-532, Sep. 2016.

[21] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Network,* vol. 28, no. 6, pp. 18-26, Nov. 2014.

[22] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management and prioritization in 5g mobile systems," *in 22nd European Wireless Conference*, Oulu, Finland, May 2016.

[23] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5g mobile wireless networks," *IEEE Network*, vol. 29, no. 3, pp. 68-74, May 2015.

[24] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5g network infrastructure," *in IEEE International Conference on Communications (ICC)*, London, UK, Jun. 2015.

[25] N. Raveendran, Y. Gu, C. Jiang, N. H. Tran, M. Pan, L. Song, and Z. Han, "Cyclic Three-Sided Matching Game Inspired Wireless Network Virtualization," *IEEE Transactions on Mobile Computing*, DOI 10.1109/TMC.2019.2947522.

[26] N. Raveendran, Y. Zha, Y. Zhang, X. Liu, and Z. Han, "Virtual Core Network Resource Allocation in 5G Systems using Three-Sided Matching," *in IEEE International Conference on Communications (ICC),* Shanghai, China, May 2019.

[27] Y. Zhang, N. H. Tran, D. Niyato, and Z. Han, "Multi-Dimensional Payment Plan in Fog Computing with Moral Hazard," *in 15th IEEE International Conference on Communication Systems (ICCS),* Shenzhen, China, Dec. 2016.

[28] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," *in Proceedings of the First International Workshop on Mobile Cloud Computing and Networking (MobileCloud),* pp. 19-26, Bangalore, India, Jul. 2013.

[29] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT Applications in the Fog: A Distributed Dataflow Approach," *in 5th International Conference on the Internet of Things (IOT),* Seoul, South Korea, Oct. 2015.

[30] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 8, p.p. 7475-7484, Aug. 2018.

[31] C. Anglano, M. Canonico, P. Castagno, M. Guazzone, and M. Sereno, "A Game-Theoretic Approach to Coalition Formation in Fog Provider Federations," *in Proceedings of the 3rd IEEE International Conference on Fog and Mobile Edge Computing (FMEC),* pp. 123-130, Barcelona, Spain, Apr. 2018.

[32] J. Li, J. Jin, D. Yuan, and H. Zhang, "Virtual Fog: A Virtualization Enabled Fog Computing Framework for Internet of Things," *IEEE Internet of Things Journal,* vol. 5, no. 1, pp. 121-131, Feb. 2018.

[33] C.-L. Su, "Equilibrium Problem with Equilibrium Constraints: Stationarities, Algorithms and Applications," *in Ph.D. Dissertation, Department of Management Science and Engineering, Stanford University,* Stanford, CA, Sep. 2005.

[34] A. Ehrenmann, "Equilibrium Problems with Equilibrium Constraints and their Application to Electricity Markets," *in Ph.D. Dissertation, Judge Institute of Management, Cambridge University,* Cambridge, UK, 2004.

[35] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems," *SIAM Journal on Optimization,* vol. 26, no. 1, pp. 337-364, Jan. 2016.

[36] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A General Analysis of the Convergence of ADMM," *in Proceedings of the 32nd International Conference on Machine Learning (ICML),* pp. 343-352, Lille, France, Jul. 2015.

[37] E. A. Jorswieck, "Stable matchings for resource allocation in wireless networks," *in Proceedings of the 17th International Conference on Digital Signal Processing (DSP),* pp. 343-352, Corfu, Greece, Jul. 2011.

[38] A. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis,* Cambridge University Press, 1992.

[39] D. F. Manlove, *Algorithmics of Matching Under Preferences,* World Scientific, 2013.

[40] R. W. Irving, P. Leather, and D. Gusfield, "An efficient algorithm for the optimal stable marriage," *Journal of the ACM (JACM),* vol. 34, no. 3, pp. 532-543, Jul. 1987.

[41] Z. Han, D. Niyato, W. Saad, and T. Basar, *Game Theory for Next Generation Wireless and Communication Networks: Modeling, Analysis, and Design,* Cambridge University Press, UK, 2019.

[42] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Communications Magazine,* vol. 53, no. 5, pp. 52-59, Apr. 2015.

[43] D. Gale, and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly,* vol. 69, no. 1, pp. 9-15, Jan. 1962.

[44] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-Many Matching Games for Proactive Social-Caching in Wireless Small Cell Networks," *in 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt),* Hammamet, Tunisia, May 2014.

[45] W. A. Massey, "The Analysis of Queues with Time-Varying Rates for Telecommunication Models," *Telecommunication Systems,* vol. 21, no. 24, pp. 173204, 2002.

[46] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications,* Cambridge University Press, UK, 2011.

[47] M. Hong, and Z.-Q. Luo - On the Linear Convergence of the Alternating Direction Method of Multipliers. [Online]. Available: https://arxiv.org/pdf/1208.3922.pdf.

[48] GS NFV-MAN 001 - V1.1.1 - Network Functions Virtualisation (NFV); Management and Orchestration. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.

[49] M. J. Osborne - Mathematical methods for economic theory. [Online]. Available: https://mjo.osborne.economics.utoronto.ca/index.php/tutorial/index/1/cv1/t.

[50] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang - Asynchronous Distributed ADMM for Large-Scale Optimization- Part I: Algorithm and Convergence Analysis. [Online]. Available: https://arxiv.org/pdf/1509.02597.pdf.