Near Optimal Learning-Driven Mechanisms for Stable NFV Markets in Multitier Cloud Networks

Zichuan Xu, *Member, IEEE*, Haozhe Ren, *Student Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, Qiufen Xia, *Member, IEEE*, Wanlei Zhou, *Member, IEEE*, Pan Zhou, *Senior Member, IEEE*, Wenzheng Xu, *Member, IEEE*, Guowei Wu, and Mingchu Li

***Abstract*—More and more 5G and AI applications demand flexible and low-cost processing of their traffic through diverse virtualized network functions (VNFs) to meet their security and privacy requirements. As such, the Network Function Virtualization (NFV) market has been emerged as a major service market that allows network service providers to trade their network services among customers. Since each service market usually involves complex interplays among players with different roles, efficient mechanisms that guarantee stable and efficient operations of the NFV market are urgently needed. One fundamental problem in the NFV market is how to maximize the social welfare of all players so that all players have incentives to participate in the activities of the market. In this paper, we first formulate a novel social welfare maximization problem in an NFV market of a multi-tier edge cloud network, with the aim to maximize the total revenue collected from all players, and we implement VNF services on Virtual Machines (VMs) leased by service providers to fulfill customers with service requests, where the edge cloud network consists of both cloudlets in edge networks and remote data centers in the core network. We then design an efficient incentive-compatible mechanism for the problem, and analyze the existence of a Nash equilibrium of the mechanism. Also, we consider an online social welfare maximization problem with uncertain values of customers and without the knowledge of future request arrivals, for which we devise an online learning algorithm by adopting the Multi-Armed Bandits (MAB) method with a bounded regret. We finally evaluate the performance of the proposed mechanisms through simulations and a testbed. Results show that the proposed mechanisms deliver up to 27% higher social welfare than those of existing studies.**

***Index Terms*—Multi-tier cloud networks, network function virtualization, near optimal incentive-compatible mechanisms, price of anarchy, online learning.**

I. Introduction

RECENTLY, a market that regulates the production and consumption of 5G applications is emerging. Reports show that the global 5G market is expected to grow from USD 53.93 billion in 2020 to USD 123.27 billion by 2025, at a Compound Annual Growth Rate (CAGR) of 18.0% during the forecast period [1]. This brings ever-growing 5G applications, and most of such applications require processing their data traffic by virtualized network functions (VNFs), such as firewalls and intrusion detection systems, to guarantee the data security and privacy of the traffic. As such, the ETSI Industry Specification Group on Network Function Virtualization (NFV) has developed a set of specifications and reports to enable an open NFV market [41].

In an NFV market, network service providers provide network services that consist of a set of VNFs to customers on demand [10], [59]. Customers can utilize the provided service services to implement their 5G applications at any time through stochastic arriving requests. The key of guaranteeing the success of the NFV market is to ensure that the optimal social welfare is achievable and the market will converge to a predictable stable status in the end. Thus, both network service providers and their customers have incentives to participate in the NFV market to earn revenues. Otherwise, selfish players may seek other markets for higher revenues and are reluctant to participate in the market due to unpredictable performance. In this paper, we investigate the problem of maximizing the total social welfare of selfish players in an NFV market. To this end, given that each player is selfish, network service providers need to carefully determine their service placement locations and prices of VNF service instances to maximize their revenue.

Designing efficient mechanisms for near-optimal stable operations of an NFV market is challenging. First, there are multiple network service providers and customers. The matching between network service providers and customers

Manuscript received 15 September 2021; revised 26 January 2022 and 13 April 2022; accepted 8 May 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. He. Date of publication 8 June 2022; date of current version 20 December 2022. The work of Zichuan Xu and Qiufen Xia was supported by the National Natural Science Foundation of China (NSFC) under Grant 62172068, Grant 62172071, Grant 61802048, and Grant 61802047; and in part by the "Xinghai Scholar" Program. The work of Weifa Liang was supported by a grant from the City University of Hong Kong under Project 9380137/CS. The work of Pan Zhou was supported in part by NSFC under Grant 61972448. The preliminary version of this paper was published on IEEE INFOCOM 2021 [DOI: 10.1109/INFOCOM42981.2021.9488819]. *(Corresponding author: Wenzheng Xu.)*

Zichuan Xu, Haozhe Ren, Guowei Wu, and Mingchu Li are with the School of Software, Dalian University of Technology, Dalian 116620, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China (e-mail: z.xu@dlut.edu.cn; renhaozhe@mail.dlut.edu.cn; wgwdut@dlut.edu.cn; mingchul@dlut.edu.cn).

Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: weifa.liang@cityu.edu.hk).

Qiufen Xia is with the International School of Information Science and Engineering, Dalian University of Technology, Dalian 116620, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China (e-mail: qiufenxia@dlut.edu.cn).

Wanlei Zhou is with the Institute of Data Science, City University of Macau, Macau, China (e-mail: wlzhou@cityu.mo).

Pan Zhou is with the Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: panzhou@hust.edu.cn).

Wenzheng Xu is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu@scu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNET.2022.3179295, provided by the authors.

Digital Object Identifier 10.1109/TNET.2022.3179295

is fundamentally important to guarantee the stability of the NFV market. An ill matching may simply lead to players quitting from the market. In particular, it is critical to design an incentive-compatible and stable mechanism with a Nash equilibrium so that all players have incentives to participate in the NFV market. Furthermore, it is expected that the social welfare achieved through the Nash equilibrium should not be far from the optimal one. Second, the interactions of players in an NFV market are complex. Centralized coordination between customers and network service providers may be impossible. Instead, a distributed mechanism that allows each player to make decisions based on its own observations is needed. Third, much information about the NFV market is uncertain. For example, customers usually have preferences on different services. Such service preferences are referred to as *values* of customers. Such values of customers determine the revenues of network service providers. However, given that customers are selfish, they will not expose their values to the market. Thus, the values of customers are uncertain. Dynamically and adaptively learning the values of customers is vital to maximize the revenues of network service providers, thereby maximizing the final social welfare of the NFV market. Unfortunately, the values of customers are determined by different types of correlated contexts including obtained revenues of players, network delays, etc. It thus is challenging to design an online learning mechanism to accurately predict the values of customers.

Although there are extensive studies on NFV in software defined networks, mobile edge networks, and conventional cloud networks [32], [44], [45], [52], there are only a handful of studies on mechanism design for NFV markets in mobile edge computing (MEC) [10], [59], including a double-auction approach [10] and an online stochastic buy-sell mechanism [59]. These proposals basically require a central or multiple collaborating brokers to implement the aforementioned mechanisms. Brokers know the values of customers in each round of bidding, and even the value of each customer may be known by the others. However, customers rarely disclose the details of their values to others in real-world NFV markets. In addition, auction mechanisms require a lot of direct communications between brokers and players, and their performance is poor when the values of customers are uncertain. Therefore, the auction mechanism may not be an ideal model for the online social welfare maximization problem in a real NFV-market. Therefore, an online algorithm based on the contextual Multi-Armed Bandit (MAB) method is designed in this paper, which learns the uncertain value of customers under various correlated contexts.

To the best of our knowledge, we are the first to investigate the online social welfare maximization problem with uncertain values of customers in an NFV market under a multi-tier cloud network. The main contributions of this paper are as follows.

- We formulate the social welfare maximization problem in an NFV market with multiple network service providers leasing virtual machine (VM) resources to implement VNF services, as well as customers with NFV-enabled requests. We also formulate the online social welfare maximization problem with stochastic arriving requests.
- We develop an effective mechanism for the social welfare maximization problem, and show that the mechanism is incentive-compatible and exists at least one Nash equilibrium. We analyze the quality of the Nash equilibrium by showing its Price of Anarchy (PoA), which quantifies the worst case gap between the social welfare of the Nash equilibrium and the optimal solution with non-selfish players.
- We design a distributed mechanism based on the contextual MAB method for the online social welfare maximization problem with uncertain values of customers and stochastic arriving requests, which allows network service providers to determine the optimal locations of VNFs with a bounded regret.
- We evaluate the performance of the proposed mechanisms by both extensive experimental simulations and implementations in a flexible, scalable and real testbed. The results show that the proposed online algorithm obtains up to 27% higher social welfare than those of existing studies.

The remainder of the paper is arranged as follows. Section II summarizes the state-of-the-art of related studies. Section III introduces the system model and defines the problems. Section IV provides an incentive-compatible facility location game for the social welfare maximization problem. Section V devises an online learning algorithm for the online social welfare maximization problem with uncertain value of customers. Section VI and Section VII provide experimental results, and Section VIII concludes the paper.

II. Related Work

Network function virtualization has attracted much attention in the past few years [8], [12], [22], [25]–[27], [30], [32], [34], [37], [38], [44], [47]–[50], [54], [56], [60], [62], [63]. Most of them did not consider the VNF provisioning under an NFV market. For example, Huang *et al.* [26] proposed an optimization framework based on online federated learning and deep reinforcement learning (DRL) for extensible deployment of service chains in a network. Feng *et al.* [17] studied the NFV service distribution problem, by effectively consolidating flows into a finite number of active resources. He *et al.* [25] proposed a DRL framework with an attention mechanism to implement the placement and routing of VNFs in a network. Luo and Wu [35] proposed an online algorithm by adjusting the deployment of VNF instances to achieve efficient scaling. Jin *et al.* [27] considered the latency-aware service chaining problem in edge networks, by formulating a mixed integer linear programming with the aim to minimize the total resource consumption.

Although the game theory has been applied to allocation and pricing of VM resources in cloud networks with multiple service providers [18], [23], [24], [51], [57], there is a fundamental difference from the VNF service provisioning in an NFV market. The proposed methods thus cannot be directly applied to the VNF placement of the NFV market.

That is, serving service chain requests in an NFV market require not only the placement of VNFs (run in VMs) but also traffic routing of customer requests from their sources to their destinations.

There are several studies on mechanism design for NFV markets [10], [14], [21], [59], which are closely related to the study of this paper. For example, Borjigin *et al.* [10] devised a double-auction approach for resource allocation in an NFV market, to maximize the revenue of an NFV broker, customers and resource suppliers. Gu *et al.* [21] proposed an auction-based mechanism for VNF chaining and achieves near-optimal social welfare in the NFV market. Zhang *et al.* [59] devised an online stochastic by-sell mechanism for network function chaining in an NFV market. Dieye *et al.* [14] studied the problem of multi-domain resource allocation for service chains to maximize the revenue of infrastructure and service providers through a DRL based auction. However, these studies only considered the interactions among infrastructure providers, they did not considered the interactions among network service providers who do not own the resources. In addition, most of these studies assumed that the customers values are given or can be obtained. On the contrary, we are the first to investigate the social welfare maximization problem in an NFV market with multiple customers and multiple network service providers without owning any resources. Note that this paper is an extension of our conference paper in [53], and new contributions of this paper include new algorithms for service chaining with delay requirements and an online learning algorithm for the social welfare maximization problem with uncertain values of customers.

Tackling the social welfare maximization problem is to allow multiple network service providers in an MEC network jointly place the VNFs of their services to edge locations economically and fairly. From the view of a single network service provider, the problem is analog to the facility location problem with an objective of maximizing the revenue collected by placing VNFs in the edge network. If each network service provider is treated as a player, a game with different types of players that tries to place a set of facilities to the network is referred to as the *facility location game*.

Although there are studies on mechanism design for facility location games [9], [11], [19], [20], [40] and the technique of multi-armed bandits [2], [4], [6], [15], [31], they may not be able to applied to the social welfare maximization problem. First, most mechanisms for facility location game assumed that each player has a facility to open [19], [20]. For example, Fong *et al.* [19] proposed the fractional preference model of facility location game, so that each customer has its own preference for each opening facility. Goemans *et al.* [20] provided a mechanism for connecting open facilities with cooperative customers to minimize the total cost. However, each network service provider in this paper may lease VMs at different locations. Second, the contextual MAB technique usually does not consider the availability of contexts to experts. For example, Dimakopoulou *et al.* [15] developed an integrated balancing contextual bandit algorithm to reduce the estimation biases. Instead, the experts in our consideration are independent third-parties. Each expert may not learn the statuses of all contexts in the network, due to performance and reachability constraints. That is, we need to customize the contextual MAB technique to allow each expert to learn from a subset of contexts. As such, the optimization techniques used in this paper can also provide reference for the application of facility location game and contextual MAB for similar optimization problems.

III. Preliminaries

In this section, we first formulate the system model and define notations. We then introduce the game-theoretic model. Finally, we define the problem precisely.

A. System Model

We consider a multi-tier cloud network $G = (\mathcal{L}, E)$ consisting of a set $\mathcal{L}$ of locations in cloudlets and remote data centers, and a set E of links (or VPN paths) that interconnect locations. The computing resource in locations (i.e., cloudlets and data centers) is virtualized as containers or VMs. We consider a scenario where network service providers do not own their infrastructures but can lease VMs from an infrastructure provider to implement their VNFs. We further consider an NFV market that consists of a number of network service providers offering network services, and each service consists of VNFs in the resource pools of service providers located in different cloudlets and data centers. We assume that location $L_j \in \mathcal{L}$ can only accommodate a finite number K_j VMs to implement VNFs [52]. Let $\mathcal{V}_j$ be the set of available VMs in location L_j and let $v_{j,m}$ a VM in $\mathcal{V}_j$, where $1 \leq m \leq K_j$. Each VM $v_{j,m}$ has an uploading bandwidth capacity $B^u_{j,m}$ and a download bandwidth capacity $B^d_{j,m}$ [7]. Note that each VM $v_{j,m} \in \mathcal{V}_j$ may or may not be instantiated, considering that the infrastructure provider has its own VM management policy.

There are Q network service providers in the system, and let q_i denote the ith network service provider with $1 \leq i \leq Q$. The services provided by these network service providers are demanded by N customers. Let u_k be the kth customer with $1 \leq k \leq N$. Each customer u_k is selfish and chooses an instance of its network service from any network service provider. Without loss of generality, we consider that the network service required by each customer composes a set of VNFs organized as a *VNF service graph* [3], [17], [61]. Fig. 1 illustrates a multi-tier cloud network with multiple network service providers and customers.

B. NFV Markets and Network Services

In an NFV market, customers purchase network services to process and transfer their traffic. To this end, each customer issues a request that demands a set of VNFs as specified in a VNF service graph, such that the performance and security of its traffic transfer is guaranteed. Denote by r_k an NFV-enabled request of customer u_k. It specifies a source node s_k and a destination node t_k, and needs to transfer the amount of its traffic ρ_k from the source to the destination. In addition, the

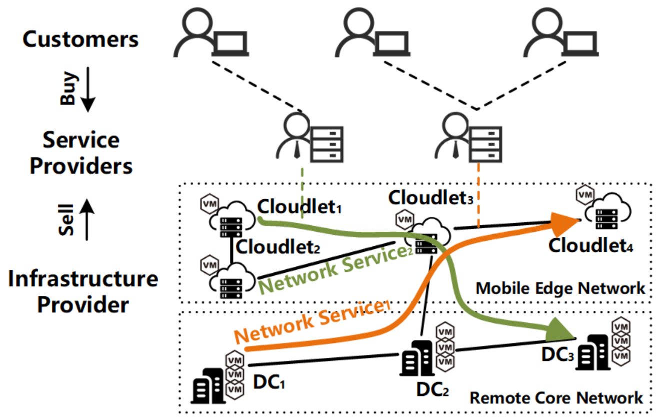


Fig. 1. A multi-tier cloud network with network service providers offering network services to customers through issuing NFV-enabled requests.

traffic of request r_k needs to be processed by a set of VNFs organized as a VNF service graph, before reaching destination t_k. Let H_k be the VNF service graph of request r_k. Let f_l be a VNF in H_k. The data traffic of each request will be processed by the VNFs in service graph H_k in the specified order in H_k. Let $\mathcal{F}$ be the set of network functions provided by all network service providers for all requests in a multi-tier cloud network.

C. Cost and Social Welfare Model

Network service providers sell network services in terms of VNF service graphs to customers for implementing requests of customers. The cost of implementing a request consists of the usage costs of both computing and bandwidth resources in a cloud network. Assume that the cost of processing the traffic of request r_k in a VNF at location $L_j \in \mathcal{L}$ is proportional to the amount of the traffic to be processed. Let $c^p_{k,j}$ be the cost of processing a unit traffic of r_k in L_j. The cost $c_{k,j}$ of placing the VNFs of service graph H_k of r_k in location L_j thus is

$$c_{k,j} = c^p_{k,j} \cdot \rho_k, \tag{1}$$

if VNF service graph H_k is placed to an existing VM instance in L_j; otherwise, when the H_k is assigned to a newly instantiated VM, we have

$$c_{k,j} = c^p_{k,j} \cdot \rho_k + \xi_j, \tag{2}$$

where ξ_j the *start-up* cost of instantiating a new VM in location L_j. Note that the start-up cost is usually given and does not change as time goes [16].

Following the settings of most infrastructure providers [36], [46], network service providers lease a certain amount of bandwidth resource to transfer traffic in/out of a location $L_j \in \mathcal{L}$. The bandwidth cost of request r_k is proportional to the amount of traffic it needs to transfer. Let $c^b_{k,j}$ be the cost of transmitting a unit of traffic from s_k of request r_k to location L_j, which can be derived by finding a shortest path from s_k to L_j. Then, the bandwidth consumption cost for request r_k on the shortest path is $c^b_{k,j} \cdot \rho_k$.

Network service provider q_i sells its network services to customers. Let $\gamma_{i,k}$ be the price that network service provider q_i asks for an instance of service graph H_k. A customer u_k has to pay the asked price by q_i for using the service graph. The revenue δ_i received by q_i thus is

$$\delta_i = \gamma_{i,k} - \rho_k(c^p_{k,j} + c^b_{k,j}). \tag{3}$$

Each customer u_k has a value for a *service instance* of service graph H_k provided by network service provider q_i, which is denoted by $\pi_{i,k}$. Note that such a value of each customer is the primary criteria that indicates whether it prefers a network service provider. A higher value for a network service provider usually means that its provided network service has a higher quality. Unfortunately, such values of each customer are private and not known by others If the customer pays a price $\gamma_{i,k}$ to use service graph H_k, it collects a revenue of

$$\Delta_k = \pi_{i,k} - \gamma_{i,k}. \tag{4}$$

A customer only buys an instance of its service graph from network service provider q_i if $\pi_{i,k} \geq \gamma_{i,k}$, and pays $\gamma_{i,k}$. Although network service provider q_i does not know the exact value of $\pi_{i,k}$, it can only observe whether the customer buys the implementation.

Since there are multiple players (network service providers and customers) in the NFV market, we aim to maximize the *social welfare*, i.e., the total revenue received by all players participating in the NFV market. Let $\Phi_{Q,N}$ be the social welfare in an NFV market with Q network service providers and N customers. $\Phi_{Q,N}$ can be calculated by

$$\Phi_{Q,N} = \sum_{i=1}^{Q} \delta_i + \sum_{k=1}^{N} \Delta_k. \tag{5}$$

D. Delay Model

Processing and transmitting the data traffic of request r_k incurs both processing and transmission latencies [5], [55], [61]. Each request usually has a delay requirement to make sure its traffic being transmitted to its destination node t_k while meeting its specified Quality of Service (QoS) requirement. The processing delay of r_k is due to the processing of its traffic by its service graph H_k. The transmission delay of request r_k is due to the transmission of its traffic from source node s_k to the destination node in H_k, which is the delay of the longest transmission path in the network from s_k to t_k in H_k.

E. Game Theory and Nash Equilibrium

We consider a game that consists of Q network service providers and N customers as players. Each network service provider $q_i \in Q$ provides its network services to customers by instantiating service graphs in leased VMs. Each customer $u_k \in N$ selects an instance of its service graph H_k from one of the Q network service providers. Therefore, the strategy space of each customer u_k is the set of network service providers, and each candidate network service provider has a number of service instances of H_k of u_k, i.e., $\{q_1, q_2, \cdots, q_Q\}$. On the other hand, network service provider q_i makes its decision of where to implement request r_k. It also decides the price for the implemented request so that its revenue can be maximized. Let $\mathcal{L}'_i$ be the candidate locations for q_i. For each service graph H_k, the strategy space of network service provider q_i includes all locations in $\mathcal{L}'_i$. Notice that each location L_j is a candidate location of service provider q_i if q_i can instantiate (or has) VMs in L_j. Consequently, different service providers may have common locations to implement their service graphs. However, a network service provider may not instantiate any VM in a candidate location. It instead strategically places instances of service graphs to one of the locations such that its revenue is maximized.

The aforementioned game can be considered as a facility location game. It must be mentioned that the facility location game is essentially different from the facility location problem [20], the latter deals with placing facilities in a network so that each facility can serve a certain number of clients. In contrast, the facility location game puts much emphasis on different parties to place their facilities in the network. A special case of the facility location game can be reduced to the facility location problem, when all network service providers are not selfish, the VNFs of each service graph can be consolidated in a single location, and there are no delay requirements and capacity constraints. The facility location game problem is reduced to the facility location problem by taking each request as a client and any edge location with a placed facility has enough VMs for its assigned requests.

F. Problem Definitions

Given an NFV market under a multi-tier edge cloud network $G = (\mathcal{L}, E)$ with Q network service providers offering network services to customers, and a set $\mathcal{R}$ of NFV-enabled requests, the following two optimization problems in G will be considered.

Problem 1: *The social welfare maximization problem with given leased resources:* In an NFV market of a multi-tier edge cloud network G, the problem is to maximize the social welfare of the market, subject to that the number of available VMs can be leased in each location $L_j \in \mathcal{L}$, the upload and download bandwidth allocated to each VM $v_{j,m}$ in location $L_j \in \mathcal{L}$, and the delay requirement of each request.

Problem 2: In most practical scenarios, NFV-enabled requests arrive into G one by one without the knowledge of future arrivals, and customers may deviate from their best strategies that usually are not revealed to other customers and network service providers. *The online social welfare maximization problem with uncertain values of customers* in G is to admit or reject each incoming NFV-enabled service request immediately, with the aim to maximize the social welfare of the market, subject to the number of available VMs that can be leased in each location $L_j \in \mathcal{L}$, the upload and download bandwidth allocated to each $v_{j,m}$ of the VMs in location $L_j \in \mathcal{L}$, and the delay requirement of each request.

For clarity, the symbols used in this paper are summarized in Table I.

IV. A Facility Location Game for the Social Welfare Maximization Problem With Given Leased Resources

We now devise an efficient mechanism for the social welfare maximization problem with given leased resources. We also analyze the quality of the mechanism by showing the PoA.

A. Overview

The essence of the proposed mechanism is a multi-stage facility location game with both network service providers and customers as strategic agents. Specifically, the game consists of three stages. In the first stage, network service providers decide which cloudlets or data centers to implement NFV-enabled requests. In the second stage, service providers set the prices for customers. In the last stage, each customer selects a network service provider and pays the specified price. The rest is to specify how prices are set and how requests are assigned to their candidate locations by each network service provider.

The basic idea of our mechanism is to adopt a pricing mechanism that allows each player in the game to make its decision based on its true values for the service graphs offered by network service providers. The problem of assigning requests to VMs of each network service provider then is reduced to a minimum weight perfect matching problem [29] in an auxiliary bipartite graph.

We consider two types of customers: (1) the customers always select the network service provider that could achieve their own maximum profits, and they are referred to as *best response customers*; and (2) the customers strategically decide their selections by observing and interacting with other customers.

B. Mechanism With Best Response Customers

We now design a mechanism for the problem with best response customers, which consists of three stages: **Stage 1**, location selection by each network service provider q_i; **Stage 2**, pricing by each network service provider q_i for each customer; and **Stage 3**, network service provider selection by each customer.

Stage 1. Given a set $\mathcal{L}'_i$ of candidate locations, each network service provider q_i first decides which locations for NFV-enabled requests. From the network service provider's point of view, it aims to maximize its own revenue. To this end, it needs to strategically admit a subset of requests that could lead to the maximum revenue, given the limited number of available VMs in the multi-tier cloud network. To enable low-cost admissions of requests, we reduce the problem of location choices by each network service provider q_i to the problem of finding a minimum weight perfect matching in a bipartite graph $G' = (V', E')$ as follows.

The node set V'_i of G'_i is divided into two disjoint subsets, i.e., V'_a and V'_b. Each node in set V'_a corresponds an NFV-enabled request $r_k \in \mathcal{R}$, and each node in set V'_b denotes an available VM in $\cup_{L_j \in \mathcal{L}} \mathcal{V}_j$ of network service provider q_i. We add an edge between each node in V'_a and each node in V'_b, to represent an assignment of an NFV-enabled request r_k to a VM owned by network service provider q_i, if the VM has sufficient upload and download bandwidths for the request. Let $(r_k, v_{j,m})$ be an edge in G'_i.

Recall that the revenue received by q_i due to serving request r_k is $\gamma_{i,k} - \rho_k(c^p_{k,j} + c^b_{k,j})$ if its assigned VM is already instantiated; otherwise, the revenue is $\gamma_{i,k} - \rho_k(c^p_{k,j} + c^b_{k,j}) - \xi_j$ as the start-up cost has to be incorporated when instantiating a VM. Since $\gamma_{i,k}$ is not determined by the network service

TABLE I

SYMBOLS

Symbols	Meaning
$G = (\mathcal{L}, E)$	a set of locations $\mathcal{L}$ in cloudlets and remote data centers, and a set E of links
L_j	a location in $\mathcal{L}$
$\mathcal{V}_j$	the set of available VMs in location L_j
K_j	the number of VMs that can be implemented in location L_j
$v_{j,m}$	the VM in $\mathcal{V}_j$, where $1 \le m \le K_j$
$B^u_{j,m}$,$B^d_{j,m}$	the upload bandwidth capacity constraint and the download bandwidth capacity constraint of $\mathcal{V}_j$
q_i	the ith network service provider, where $1 \le i \le Q$
u_k	the kth customer, where $1 \le k \le N$
r_k	the NFV-enabled request of customer u_k
s_k,t_k, ρ_k	the source node, the destination node and the amount of the traffic of r_k
H_k	the service graph of request r_k of customer u_k
f_l	the lth VNF in H_k, where $1 \le l \le \lvert H_k \rvert$
$\mathcal{F}$	the set of network functions provided by all network service providers in the multi-tier cloud network
$c^p_{k,j}$,$c^b_{k,j}$	the cost of processing and transmitting a unit traffic of r_k
$c_{k,j}$	the cost of placing service graph H_k of r_k in location L_j
ξ_j	the start-up cost of instantiating a new VM in location L_j
$\gamma_{i,k}$	the price that network service provider q_i asks for an instance of service graph H_k
$\pi_{i,k}$	the value for a *service graph instance* of H_k provided by network service provider q_i
δ_i	the revenue received by network service provider q_i
Δ_k	the revenue received by customer u_k
$\mathcal{L}'_i$	the candidate locations for q_i
$\Phi_{Q,N}$	the social welfare in an NFV market with Q network service providers and N customers
$G' = (V', E')$	a bipartite graph with a set V' of nodes, a set E' of edges, which is constructed base on $G = (\mathcal{L}, E)$
$w(r_k, v_{j,m})$	the weight of edge $(r_k, v_{j,m})$ in the bipartite graph $G' = (V', E')$
M_i	the minimum weighted perfect matching in G'_i that minimizes the cost of network service provider q_i
$\mathcal{R}_i$	the set of requests assigned to the VMs of network service provider q_i
$\mathcal{R}^{exd}_i$, $\mathcal{V}^{exd}_i$	the set of excluded requests and the corresponding set of VMs to which those requests in $\mathcal{R}^{exd}_i$ are assigned in matching M_i.
S_i	be the strategy of player i
OPT and O	the set of locations for the socially optimal solution and its corresponding strategy space
$\Gamma_{<\pi_{i,k}}$	the set of network service providers whose prices are smaller than its corresponding value of request r_k
ψ_{max}, ψ_{min}	the maximum and minimum ratio of customer's value and the cost
ϵ	the real non-negative parameter of ϵ-equilibrium
θ	a context available in the system
exp_n	the expert, where $1 \le n \le Y$
$w_{r,\theta}(u_k)$	the weight of context θ for customer u_k
σ	the penalty factor of the weight $w_{r,\theta}(u_k)$ on customer u_k
$p_{\theta,k}$	the recommendation probability of a customer u_k for each context θ
$\Theta(exp_n)$	the contexts observed by expert exp_n
$p_{n,k}$	the probability of the expert exp_n selects a customer u_k
$c_{r,\theta,k}$	the penalty of context θ
$Reg(T)$	the regret of **Algorithm** `OL_MT`
$\mathcal{U}^*$	the best subset of customers to participate in the game which can get the maximum social welfare
$\Phi_{\mathcal{U}^*}$	the social welfare if a best subset $\mathcal{U}^*$ of customers are selected to participate online the game
W'_r , $W'_{r,n}$	the total weight of all customers and the customers of expert exp_n

provider q_i, maximizing its revenue is equivalent to minimizing the cost of implementing request r_k, i.e., $\rho_k(c^p_{k,j} + c^b_{k,j})$ or $\rho_k(c^p_{k,j} + c^b_{k,j}) + \xi_j$. We thus consider the cost of assigning r_k to $v_{j,m}$ in location L_j as the weight of edge $(r_k, v_{j,m})$ by

$$w(r_k, v_{j,m}) = \rho_k(c^p_{k,j} + c^b_{k,j}), \tag{6}$$

if $v_{j,m}$ is already instantiated; otherwise,

$$w(r_k, v_{j,m}) = \rho_k(c^p_{k,j} + c^b_{k,j}) + \xi_j. \tag{7}$$

The delay of the edge $(r_k, v_{j,m})$ is set to the corresponding delay of implementing request r_k in $v_{j,m}$. It must be mentioned that $|V'_a|$ may not be equal to $|V'_b|$. If $|V'_a| \ge |V'_b|$, we add $|V'_a| - |V'_b|$ *dummy VM nodes* to V'_b. Each request node in V'_a connects to each dummy VM node, and the weight of the edge is set to infinity. Otherwise, we add $|V'_b| - |V'_a|$ *dummy request nodes* to V'_a. Fig. (2) shows an example of the proposed bipartite graph G'_i.

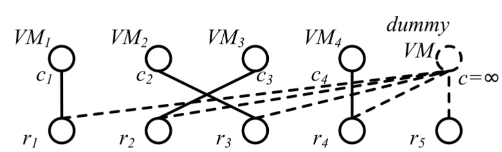


Fig. 2. An example of the constructed bipartite graph G'_i.

Having graph G'_i, we then find a minimum weighted perfect matching M_i in G'_i that minimizes the cost of network service provider q_i [29].

Given a perfect matching M_i in G'_i for network service provider q_i, each edge in M_i denotes a preference of q_i of admitting the request. Let $\mathcal{R}_i$ be the set of requests assigned to the VMs of network service provider q_i. Initially, $\mathcal{R}_i$ includes the requests that are included in matching M_i. Specifically, if all requests in $\mathcal{R}_i$ select q_i, each implementation cost corresponds to the weight of the edge in matching M'_i. However, not all VMs can meet the delay requirements of their requests. We thus remove such requests from $\mathcal{R}_i$ for consideration. Further, not all requests in $\mathcal{R}_i$ will select q_i as they may have other choices with lower implementation costs. Specifically, for each request $r_k \in \mathcal{R}_i$, we exclude it from $\mathcal{R}_i$ if there exists another network service provider $q_{i'}$ with $i' \neq i$ that results in a lower implementation cost. Let $\mathcal{R}_i^{exd}$ be the set of excluded requests and $\mathcal{V}_i^{exd}$ the corresponding set of VMs to which those requests in $\mathcal{R}_i^{exd}$ are assigned in matching M_i.

Given requests in $\cup_{i=1}^{Q} \mathcal{R}_i^{exd}$ that are excluded from the initial matching of network service providers, this procedure continues until there is no excluded requests. So far, each request in $\mathcal{R}_i$ achieves its minimum implementation cost.

Stage 2. Each network service provider q_i decides the price of the instances of its service graphs. Intuitively, to make each request r_k in $\mathcal{R}_i$ select q_i, the price it sets for the request should not be higher than that of the network service provider achieving the second lowest implementation cost for it. This is the highest price that q_i could expect to get away with charging request r_k; charging any more would give some network service provider $q_{i'}$ an incentive to undercut q_i. Note that all network service providers can obtain the implementation costs of requests in different locations, because the computing and bandwidth resource consumptions in edge clouds are usually public information.

Stage 3. Following best-response strategies, each customer selects a network service provider with the lowest price to implement its request r_k if $\pi_{i,k} \geq \gamma_{i,k}$.

Algorithm 1 details the proposed mechanism for the social welfare maximization problem, which is referred to as `FLG_SWM`.

Algorithm 1 A Facility Location Game for the Social Welfare Maximization Problem (`FLG_SWM`)

Input: A multi-tier cloud network $G = (\mathcal{L}, E)$ and a set of NFV-enabled requests $\mathcal{R}$.

Output: The assignment of each request in $\mathcal{R}$ to a network service provider.

/***Stage (1)**: Location selection by each network service provider*/
$\mathcal{V}_i \leftarrow \cup_{L_j \in \mathcal{L}} \mathcal{V}_{j,i}$; /*the set of available VMs of each network service provider q_i*/
$\mathcal{R}^{adt} \leftarrow \mathcal{R}$; /*the set of to-be-admitted requests*/
while $\mathcal{V}_i \neq \emptyset$ or $\mathcal{R}^{adt} \neq \emptyset$ **do**
  for each network service provider q_i **do**
    if $\mathcal{V}_i \neq \emptyset$ **then**
      $\mathcal{V}_i^{exd} \leftarrow \emptyset$; /*The set of VMs that are excluded from the initial minimum weight matching in a bipartite graph G_i'*/
      Construct the bipartite graph $G_i' = (V_i', E_i')$ as illustrated in Fig. 2;
      Find a minimum weight perfect matching M_i in G_i';
      Exclude a matched pair of VM and request r_k in M_i, if the VM cannot meet the delay requirement of their requests or the VM is not the first choice of r_k (i.e., there is another VM of another network service provider $q_{i'}$ that can achieve a lower implementation cost for r_k);
      Add the excluded VM into $\mathcal{V}^{exd}$, and the request into $\mathcal{R}_i^{exd}$;
      $\mathcal{V}_i \leftarrow \mathcal{V}^{exd}$;
  If there exists a request in $\cup_{i=1}^{Q} \mathcal{R}_i^{exd}$ that considers a VM in $\mathcal{V}_i$ as its best choice (the VM that can achieve the minimum implementation cost), $\mathcal{R}^{adt} \leftarrow \cup_{i=1}^{Q} \mathcal{R}_i^{exd}$;
  Otherwise, $\mathcal{R}^{adt} \leftarrow \emptyset$;
/***Stage (2)**: Pricing by each network service provider*/
For each network service provider q_i, set its price for request r_k as the cost of implementing r_k in a VM by another network service provider that has the second lowest implementation cost;
/***Stage (3)**: Service provider selection by each customer*/
Each customer selects its best response strategy, by choosing the network service provider with the lowest price to implement its NFV-enabled request r_k;

Algorithm 2 A Facility Location Game for the Social Welfare Maximization Problem With Strategic Customers (`FLG_LC`)

Input: A multi-tier cloud network $G = (\mathcal{L}, E)$ and a set of NFV-enabled requests $\mathcal{R}$.

Output: The assignment of each request in $\mathcal{R}$ to a network service provider.

Invoke **Stage (1)** and **Stage (2)** of **Algorithm** `FLG_SWM`;
for each customer u_k **do**
  if it's best-response network service provider q_i has been selected by many customers **then**
    Customer u_k chooses the least congested network service provider among the ones that offer prices that are smaller than its value;
  else
    Customer u_k chooses the network service provider with the lowest price;

C. Mechanism With Strategic Customers

The rest is to design a mechanism for the game with strategic customers that observe decisions of each other. In **Stage 3** of the proposed mechanism `FLG_SWM`, we assumed that customers adopt the best response strategy by selecting a network service provider with the lowest price. As such, many customers may select the same network service provider of a location L_j if the location has the cheapest price, this leads to a higher delay and cost due to the congestion at the location. In reality, customers may observe the preferences of other customers, and decide whether to deviate from its best response (a.k.a. selecting the network service provider with the lowest price). Specifically, if customer u_k observes that there are already many customers selecting its best-response network service provider q_i, it may deviate from q_i to avoid congestion and a long delay. To this end, customer u_k choose the least congested network service provider among the ones offering prices less than its value. The major difference of this mechanism from `FLG_SWM` is **Stage 3**, which is referred to as **Algorithm** `FLG_LC`.

D. Extensions for Service Graphs

So far we assumed that the VNFs of the service graph H_k of each request r_k are consolidated into a single VM of a location for processing. We now extend the proposed algorithm to consider the chaining of VNFs of each request r_k in its specified order.

A simple extension is to consider each pair of VNF f_l in H_k and its request r_k as a *virtual request*, and then use the proposed algorithm `FLG_SWM` to assign the virtual requests to the VM. Although this extension is simple, it basically assigns each f_l in H_k independently without considering the order of VNFs in each service graph. As such, the VNFs of a service graph H_k may be assigned to VMs in different cloudlets, leading to prohibitively-high costs and delays.

To enable algorithm `FLG_SWM` to incorporate the dependency of functions in service graph H_k, we propose an efficient heuristic algorithm for the problem. That is, we replace the minimum weighted matching in **Stage 1** of algorithm `FLG_SWM` with an iterative matching that progressively assigns the VNFs to VMs. To this end, we first find a topological sort of the functions in service graph H_k to determine a linear order of the functions, such that each directed edge $\langle f_{l-1}, f_l \rangle$ function f_{l-1} appears before f_l in the order. Given the linear order of functions in H_k, we then place the functions iteratively. Each iteration deals with a VNF in linear order of H_k and all requests that require H_k. For example, in the current iteration l, the lth function f_l in each service graph is considered.

Each f_l and each of its requests are considered as a *virtual request*. To assign each virtual request, a similar bipartite graph is built as algorithm FLG_SWM. The only difference is that the cost of edge from the virtual request to the VM is set to a weighted sum of the cost and delay of implementing the VNF f_l. The location of the VM matching with the virtual request is the chosen location of the VNF f_l in H_k. Note that the transmission cost of f_l is set to zero, if it is co-located to the location with f_{l-1} of r_k. The reason is that the transmission cost is no longer to be included as functions f_{l-1} and f_l when they are placed to the same VM. Note that if the delay of implementing request r_k is violated after considering its last VNF, all previously admitted VNFs of r_k in its service graph will be removed, and the request will be rejected. All the rest steps and stages are the same as FLG_SWM. For simplicity, this algorithm is referred to as FLG_SC.

Algorithm 3 A Facility Location Game Algorithm for the Social Welfare Maximization With Service Graphs (FLG_SC)

Input: A multi-tier cloud network $G = (\mathcal{L}, E)$ and a set of NFV-enabled requests $\mathcal{R}$.

Output: The assignment of each request in $\mathcal{R}$ to a network service provider.

1: Invoke Step 1 to Step 7 of **Algorithm** FLG_SWM;

2: **Topological sort:** find a topological sort of the functions in service graph H_k to determine a linear order of the functions f_l;

3: **Iterative matching:** each iteration deals with a VNF in the linear order of service graph, construct the bipartite graph $G'_i = (V'_i, E'_i)$ as illustrated in Fig. 2 by treating each f_l and its request as a virtual request, and find a minimum weight perfect matching M_i in G'_i;

4: **Excluding the matchings that violate delay requirements:** exclude all previously admitted VNFs of r_k in its services graph, if the delay of implementing request r_k is violated or VMs are not the best choice of r_k (i.e., there is another network service provider $q_{i'}$ that can achieve a lower implementation cost for r_k);

5: Invoke Step 11 to Step 18 of **Algorithm** FLG_SWM;

E. Algorithm Analysis

In the following we analyze the economic properties and performance of the proposed mechanisms.

Lemma 1: The proposed mechanism FLG_SWM for the social welfare maximization problem in an NFV market under a multi-tier cloud network G is incentive compatible.

See the proof of Lemma 1 in the supplementary file.

Lemma 2: The proposed mechanism FLG_SWM for the social welfare maximization problem is a potential game with potential function $\Phi_{Q,N}$.

See the proof of Lemma 2 in the supplementary file.

Lemma 3: The proposed facility game has the following three properties:

- **Property (1)**: $\Phi_{Q,N}(S)$ is submodular: for any strategy set $S \subset S' \subset A$ and any element $s \in A - S'$, we have $\Phi_{Q,N}(S \cup \{s\}) - \Phi_{Q,N}(S) \geq \Phi_{Q,N}(S' \cup \{s\}) - \Phi_{Q,N}(S')$
- **Property (2)**: Let $\alpha_i(S)$ be the added welfare of each player i to the social welfare. For all the Q network service providers, we have $\sum_i^Q \alpha_i(S) \leq \Phi_{Q,N}(S)$
- **Property (3)**: The value for one player is at least its added welfare for the society: $\alpha_i(S) \geq \Phi_{Q,N}(S) - \Phi_{Q,N}(S \backslash \{S_i\})$.

See the proof of Lemma 3 in the supplementary file.

Theorem 1: The best response dynamics of the proposed mechanism FLG_SWM converges to a pure strategy equilibrium, and the PoA of the proposed mechanism is 2.

See the proof of Theorem 1 in the supplementary file.

Theorem 2: The proposed mechanism FLG_LC converges to a ϵ-equilibrium, and its PoA is 2, where $\epsilon = \frac{\psi_{max} - \psi_{min}}{(\psi_{min} - 1)}$ with ψ_{max} denoting the maximum ratio of customer's value and the cost and ψ_{min} being the minimum one.

See the proof of Theorem 2 in the supplementary file.

V. Online Learning Algorithm for the Online Social Welfare Maximization Problem

In this section, we consider the online social welfare maximization problem with uncertain values of customers.

A. Overview

In reality, the values of customers play a vital role in maximizing the social welfare of an NFV market. Since customers are selfish, they will not reveal their values. Network service providers need to make pricing decisions based on the uncertain values of customers. To this end, network service providers can learn the values of customers, according to the behaviors of customers in terms of service selections. Because it is challenging to predict the values of customers, network service providers may not do that by themselves, instead they resort to the third parties serving as 'experts' for the values of customers. Specifically, assume that there are multiple experts in the NFV market. They serve as trustworthy third parties to collect and learn customer distributions of service selections. Each of the experts recommends a set of customers to the network service providers. Each network service provider however may not fully trust experts, by dynamically evaluating the experts. Furthermore, to be trustworthy, each expert needs to make precise predictions, which is challenging, too, due to the value of each customer may dynamically change with its context changes. For example, a customer may lower its value on a service if the service is no longer meeting its delay requirement. Also, it may increase its value on a service if a higher revenue can be returned or a network service provider has a better reputation. To enable decision makings under such correlated contexts, we propose an online learning algorithm for the online welfare maximization problem with uncertain values of customers, via leveraging the technique of contextual MAB.

B. Algorithm

In the proposed online learning algorithm, each expert observes the contexts that influence the values of customers,

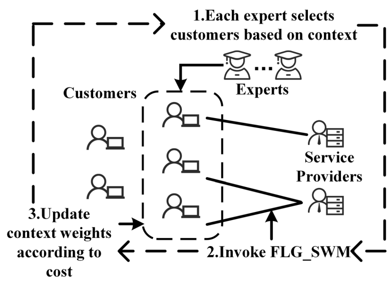


Fig. 3. The basic idea of the proposed online learning algorithm for the online social welfare maximization problem.

and learns the correlations among each context and the value of each customer. The expert then selects customers participating in the game by the learnt correlations, as illustrated by Fig. 3.

To this end, we assume that under each context, there is a set of certain probabilities of recommending the customers. Such probabilities are learnt iteratively through multiple rounds, because the values of customers on services in each context are not known. Specifically, let θ be a context available in the system. Let $p_{\theta,k}$ be the recommendation probability of customer u_k for each context. As the context changes, the probability is adaptively adjusted. Each context θ has a weight on customer u_k at each round r, denoted by $w_{r,\theta}(u_k)$. The probability $p_{\theta,k}$ is calculated by

$$p_{\theta,k} = w_{r,\theta}(u_k)/(\sum_{k'=1}^{N} w_{r,\theta}(u_{k'})). \tag{8}$$

Let exp_n be an expert with $1 \le n \le Y$. Each expert then observes different contexts related to its customers and decides to make decisions according to its observed contexts. Specifically, let $\Theta(exp_n)$ be the contexts observed by expert exp_n. The expert exp_n selects customer u_k with probability of

$$p_{n,k} = (\sum_{\theta\in\Theta(exp_n)} p_{\theta,k})/|\Theta(exp_n)|, \tag{9}$$

where $|\Theta(exp_n)|$ is the number of contexts available to expert exp_n.

The algorithm then allows each expert exp_n to choose customer u_k with probability $p_{n,k}$. After all experts have chosen their customers, **Algorithm** `FLG_SWM` is invoked to assign the requests of the chosen customers to network service providers.

However, with the progress of customer selection and request assignment in different rounds, different customers may cause different rewards. Thus, the weight $w_{r,\theta}(u_k)$ of each customer u_k under context θ participating in the game is updated in the end of each round, when the real revenues of each network service provider and each customer are revealed. In other words, penalties are assigned to the contexts with lower rewards.

It must be mentioned that, instead of predicting the values of customers directly, customers are chosen with a certain probability. Recall that our objective is to maximize the total social welfare of all players. We thus use the inverse of obtained revenue of each customer and its assigned network service provider as its penalty on its weight. That is, the penalty of context θ is the inverse of the revenue if customer u_k is selected by context θ and it uses the network provider q_i, i.e.,

$$c_{r,\theta,k} = \delta_{max} - \delta_i + \Delta_{max} - \Delta_k, \tag{10}$$

where δ_{max} and Δ_{max} are the maximum revenues that can be obtained by a network service provider and a customer, which are given as a priori.

For each context, the weight on customer u_k is updated by

$$w_{r+1,\theta}(u_k) = w_{r,\theta}(u_k) \cdot (1-\sigma)^{c_{r,\theta,k}}. \tag{11}$$

where σ is a penalty factor in the range of (0, 1/2). The above-mentioned procedure continues iteratively. The detailed steps of the proposed algorithm is shown in **Algorithm** 4, referred to as `OL_MT`.

We extend algorithm `OL_MT` to a generic case of the VNF chain of each request r_k in its specified order, the only difference from algorithm `OL_MT` is at step 9 by invoking algorithm `FLG_SC`. The rest are identical to algorithm `OL_MT`. For simplicity, we refer to this modified algorithm as algorithm `OL_SC`.

Algorithm 4 An Online Learning Algorithm for the Online Social Welfare Maximization Problem (`OL_MT`)

Input: $G = (\mathcal{L}, E)$, a set of NFV-enabled requests.
Output: An assignment of each request to either a cloudlet or datacenter for processing.
1: Initialize the weight each customer under each context θ as $w_{1,\theta}(u_k) = 1$ for the expert exp_n in time slot 1;
2: **for** each round $r \leftarrow 1 \ldots T$ **do**
3: $\quad \mathcal{U} \leftarrow \emptyset$; // the set of selected customers
4: $\quad$ **for** each expert exp_n **do**
5: $\quad\quad$ Each context update its probability of selecting a customer by Eq. (8);
6: $\quad\quad$ Collecting the probabilities under all contexts in $\Theta(exp_n)$, and calculate the average probability of selecting a customer $p_{n,k}$;
7: $\quad\quad$ For each customer u_k of expert exp_n, select it with probability $p_{n,k}$;
8: $\quad\quad \mathcal{U} \leftarrow \mathcal{U} \cup \{u_k\}$, if u_k is selected;
9: $\quad$ Invoke **Algorithm** `FLG_SWM`;
10: $\quad$ In the end of round r, observe the costs of customers, and update its weight by Eq. (11);

C. Algorithm Analysis

In the following we analyze the accumulative regret of the proposed algorithm `OL_MT`.

Theorem 3: The regret by algorithm `OL_MT` is upper bounded by $\frac{\ln T}{\sigma} + 2\sigma U_d$, assuming that there is a bound on the gap between the minimum and maximum values of each customer. Let U_d be the bound, i.e., $U_d = \pi_{max} - \pi_{min}$, where $\pi_{max} = \max_{i,k}\{\pi_{i,k}\}$, $\pi_{min} = \min_{i,k}\{\pi_{i,k}\}$, and σ is a given constant in the range of $(0, 1/2)$.

See the proof of Theorem 3 in the supplementary file.

VI. Simulations

In this section, we evaluate the performance of the proposed algorithms via extensive simulations and experiments on a testbed.

A. Parameter Settings

We consider multi-tier cloud networks by varying their sizes from 50 to 250 nodes with 5 data centers, where each network topology is generated using GT-ITM [58]. The number of cloudlets in each network is set at 10% of the network size. The cloudlets are randomly distributed in the core edge network. We also use a real network topology AS1755 from [28]. The numbers of VMs provided by each cloudlet and data center are randomly generated from [10, 15] and [15, 50], respectively. The bandwidth capacity of each VM is drawn from the range of [10 Mpbs, 100 Mbps]. Each NFV-enabled request demands a service graph containing at most 5 VNFs, picked from a set of functions: Firewall, Proxy, NAT, IDS, and Load Balancing (LB). The costs of transmitting and processing of 1 GB traffic are set within [\$0.05, \$0.12] and [\$0.15, \$0.22]. The start-up cost of a VM is set within \$0.005, following typical charges in Amazon EC2 with small variations. The pre-defined threshold of revenue decreases ϑ is set 20%. The traffic volume of each request is randomly drawn from $[10, 200]$ Megabytes. The transmission delay of a network link varies between 0.1ms and 1ms [28]. The processing delay of a unit of traffic for each VNF is randomly drawn from 0.045ms to 0.3ms, and the processing delay of a service graph is the total processing delay of its VNFs. Each request has a delay requirement ranging from 10ms to 100ms. The running time of each algorithm is obtained based on a machine with an Intel Xeon Gold 5118 dual CPUs and 256GB RAM. Unless otherwise specified, these parameters will be adopted in the default setting.

We evaluated the performance of the proposed algorithms against the following benchmarks. We first considered the double auction mechanism in [10], in which potential customers submit their bids while potential network service providers submit their ask prices to an auctioneer simultaneously. The auctioneer then chooses some price that clears the market: all the sellers who asked less than the chosen price sell and all buyers who bid more than the chosen price buy at the chosen price. The offline and online versions of this double auction mechanism are referred to as `DA` and `Online_DA`, respectively. We then considered an online auction mechanism in [33], in which potential customers calculate their revenues of potential service providers based on their true values, and choose a service provider with the highest revenue, and we refer to this benchmark as `Online_A`.

B. Performance of Algorithms `FLG_SWM` and `FLG_SC`

We evaluated the performance of algorithms `FLG_SWM`, `FLG_LC`, and `FLG_SC` against that of algorithm `DA`, in terms of the social welfare, revenues of customers and network service providers, average delay, and the running time, in GT-ITM generated networks with network sizes varying from 50 to 250.

From Fig. 4 (a), we can see that the social welfares achieved by algorithms `FLG_SWM`, `FLG_LC`, and `FLG_SC` are higher than that of `DA`. The reason is that `FLG_SWM` finds an optimal matching between buyers and sellers as shown in Fig. 2; while in `DA`, the trade between a network service provider and a customer happens as long as the bid is greater than the ask. In addition, we can see that algorithms `FLG_LC` and `FLG_SC` have lower social welfares than that of `FLG_SWM`. This is due to the fact that in `FLG_LC`, customers may deviate from its best response strategy to avoid congestion, thereby decreasing the social welfare. Also, algorithm `FLG_SC` greedily finds a location for each VNF in the service graph of a request, which easily saturates the locations with low costs and implements the VNFs in a service graph with higher costs. Furthermore, we can see that the social welfare decreases with the growth on network size, and then becomes stabilized afterwards when the network size increases from 200. This is because a larger network size means that each request can be implemented in a data center or cloudlet that is far from its source and destination with a higher probability, thereby increasing the transmission cost of data traffic. Similar patterns of revenues can be observed from Fig. 4 (b) and Fig. 4 (c), respectively.

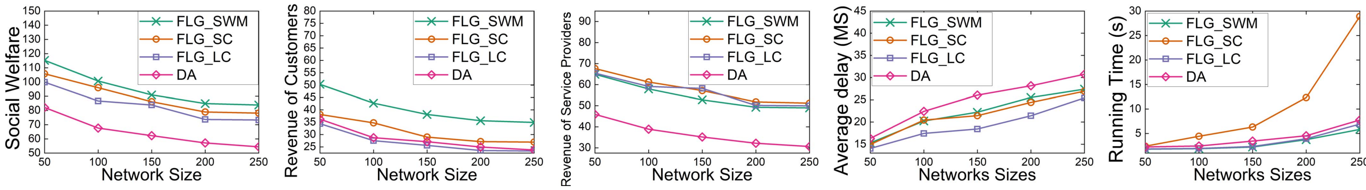
We can see from Fig. 4 (d) that the average delay achieved by `FLG_LC` is the lowest one among the three comparison algorithms. The reason is that customers in `FLG_LC` avoids locations with high congestion, which usually incur higher delays. In addition, the average delay increases with the growth of network size. This is because in larger networks the source and destination of a request may be far away from them in smaller networks. From Fig. 4 (e), we can see that `FLG_SWM` takes less running time than those of `DA`, `FLG_LC` and `FLG_SC` respectively, since `DA` spends more time in messaging among different players and `FLG_SC` takes more time in the topological sorting of each service graph.

We now evaluated the performance of algorithms `FLG_SWM`, `FLG_LC` and `FLG_SC` by varying the number of available VMs in each cloudlet from 5 to 20.

We can see from Fig. 5 (a) and Fig. 5 (c) that with the increase on the number of VMs, the social welfare increases but the revenue of service providers decreases. The reason is that the increase in the number of available VMs for deployment leads to the decrease in the second-lowest price used by service providers in pricing. Fig. 5 (b) shows that the revenues of customers by algorithms `FLG_LC` and `FLG_SC` are lower than that of algorithms `FLG_SWM` and `DA` when there are 5 VMs in each cloudlet. As shown in Fig. 5 (d), the average delays incurred by algorithms `FLG_LC` and `FLG_SC` are lower than those of algorithms `FLG_SWM` and `DA`, respectively. The rationale behind is `FLG_LC` leverages a fine-grained trade-off between the delay and the cost of implementing VNFs in each service graph, by allowing customers to deviate from its best response strategy and choose the least congested network service provider.

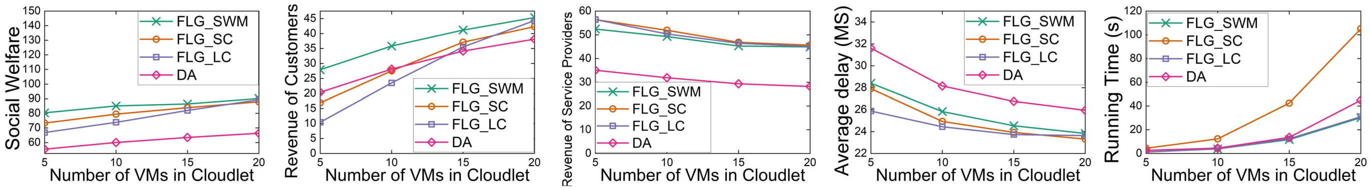
C. Performance of Algorithms `OL_MT` and `OL_SC`

We end up by evaluating the performance of algorithms `OL_MT` and `OL_SC` against that of algorithms `Online_A` and `Online_DA` in terms of the social welfare, revenues of customers and service providers, average delay, and running times, in a network with size of 200. The social welfare achieved by algorithms `OL_MT`, `OL_SC`, `Online_A` and `Online_DA` are shown in Fig. 6 (a). It can be seen that algorithms `OL_MT` and `OL_SC` achieve higher social welfares



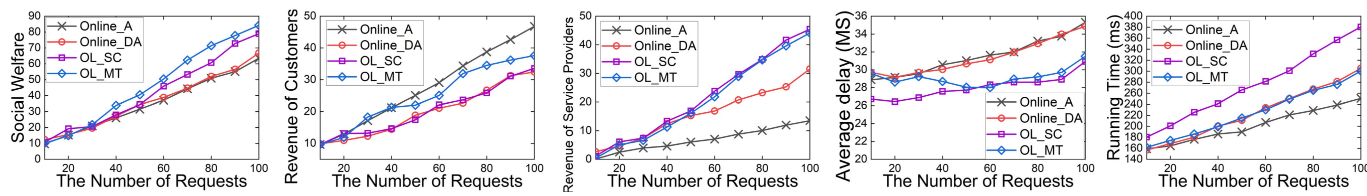
(a) Social welfare. (b) Revenue of customers. (c) Revenue of service providers. (d) Average delay. (e) Running time.

Fig. 4. The performance of algorithms `FLG_SWM`, `FLG_SC` and `DA`.



(a) Social welfare. (b) Revenue of customers. (c) Revenue of service providers. (d) Average delay. (e) Running time.

Fig. 5. The performance of algorithms `FLG_SWM`, `FLG_SC` and `DA`.



(a) Social welfare. (b) Revenue of customers. (c) Revenue of service providers. (d) Average delay. (e) Running time.

Fig. 6. The performance of algorithms `OL_MT`, `Online_A` and `Online_DA`.

because they predict the values of customers with various side information and select customers based on the context probability to maximize the total social welfare of all players. In addition, algorithm `OL_SC` has a lower social welfare compared with that of algorithm `OL_MT`. The reason behind is that algorithm `OL_SC` places the VNFs of each service graph to multiple locations of the cloud network, which increases the cost of bandwidth resource usages. Specifically, locations for the VNFs of a service graph in `OL_SC` incur cost of bandwidth resource usages if the VNFs are placed to different locations. However, if all VNFs are consolidated into a single location, only that location incurs a cost of the bandwidth resource consumption.

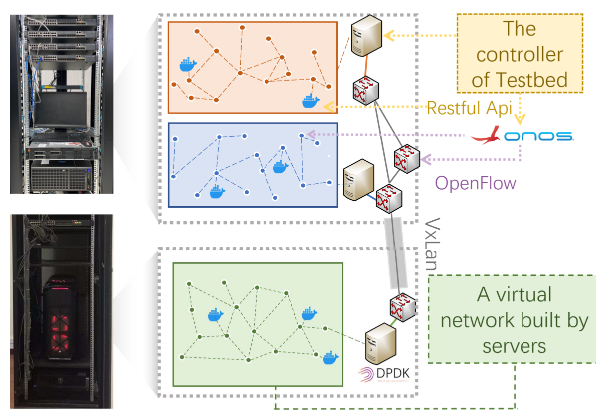


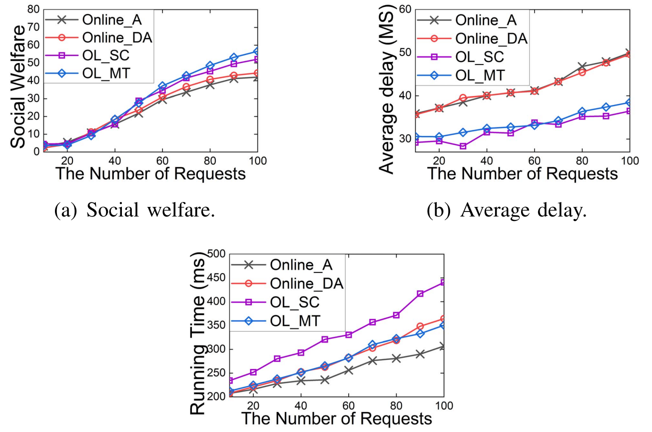
Fig. 7. A hybrid test-bed with both physical and virtual network elements.

VII. Test-Bed Implementations

The rest is to evaluate the performance of the proposed mechanisms in a real test-bed. Note that after a testbed is built, it can process requests dynamically. We thus only evaluated the proposed online games, i.e., `OL_MT` and `OL_SC` in the testbed.

A. Testbed Design Rationale

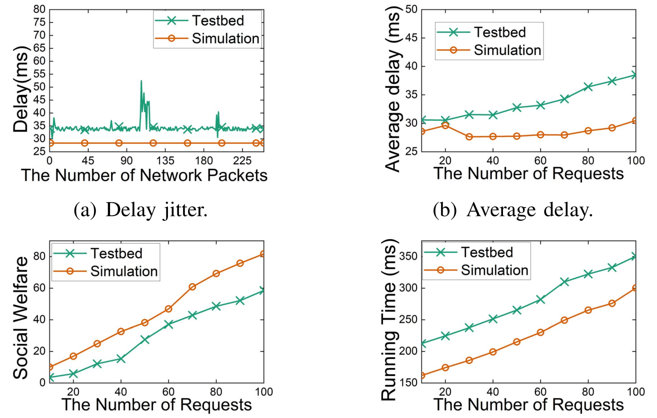
We observed that an ideal testbed is capable to evaluate the scalability of the proposed online mechanism in real environments close to production systems. Specifically, since we consider an NFV market with multiple network service providers that offer different types of service graphs in a multi-tier cloud network. It is important to evaluate the performance of the proposed mechanisms in large-scale, flexible and real testbed. Although a common approach is to build a hierarchical testbed by using an overlay virtual network and an underlay physical network, it is hard to tell the capability of the virtual or physical network determines the algorithm performance. An alternative is to interconnect the virtual and



(a) Social welfare. (b) Average delay.

(c) Running time.

Fig. 8. The performance of algorithms `OL_MT`, `Online_A`, and `Online_DA` in the test-bed.



(a) Delay jitter. (b) Average delay.

(c) Social welfare. (d) Running time.

Fig. 9. The performance of algorithms `OL_MT` in simulation and the testbed.

physical networks in a single layer such that virtual and physical network switches are peers and perform identical functionalities. However, it is challenging to implement a unified management and control to manage both virtual and physical network switches.

To enable unified control and management in a flexible, scalable, and real testbed, we build a hybrid testbed with both physical and virtual network switches, where virtual network switches can be scaled dynamically into large scale and physical ones are used to resemble properties of real environments. Since we consider a multi-tier cloud network, we build the testbed in different domains interconnected by a physical core network, where each domain has a virtual network, as shown in Fig. 7.

The physical core network composes of five H3C s5560x physical switches as shown in Fig. 7. It adopts software-defined networking and VxLan technologies to interconnect the physical switches, customers, and cross-domain servers.

The virtual network in each domain extends the physical core network by using Open vSwitch (OVS) [43] nodes as virtual forwarding devices and deploying VNFs into Docker instances, to guarantee the large-scale network expansion and high-speed forwarding.

We deploy them on two servers with each having Intel Xeon Gold 5118 dual CPUs and 256GB RAM, and 3 workstations with each having an Intel 10700HF and 16GB RAM. Furthermore, to improve the forwarding efficiency of the virtual network, we use the Data Plane Development Kit (DPDK) [13] to optimize the transmission performance of OVS and reduce the unnecessary computing overhead caused by multiple internal memory copies and excessive memory paging. We deploy the VNFs in Dockers to meet their performance and resource isolation requirements. However, the virtual network bridge cannot build a multi-tier cloud network, as it do not have multi-domain support. It does not support forwarding table operations either. We thus replace the network bridge of Docker with an OVS node that supports forwarding table operations.

Joint control of physical and virtual networks: To integrate the proposed algorithms in the testbed, we manage the testbed using an Open Network Operating System (ONOS) [42] controller. This component provides the platform traffic routing and configuration management function. ONOS can control either OVSes or physical switches to forward data traffic, but it does not support the joint control of both OVSes and physical switches. Therefore, we implement the joint control of physical and virtual resources by running agent services that implement the RestAPI in both physical server and virtual container. Specifically, each agent service in either a physical server or OVS to implement VNF placement or routing decisions obtained by the proposed mechanism.

We use DPDK-based pktgen as a traffic generator to generate the traffic of the requests. We also implement the algorithms as northbound applications in ONOS. Several experimental topologies based on AS1755 were constructed through the experimental platform to verify the algorithm. All other settings are the same as the simulations.

B. Evaluation Results

Fig. 8 shows the performance of algorithms `OL_MT`, `OL_SC`, `Online_A` and `Online_DA` in terms of social welfare and running times in the test-bed. As shown in the figure, the experimental results in the test-bed are similar to those in the simulation experiment, which verifies the accuracy of the simulation results and the availability of the proposed mechanism in the real-world NFV-Market. It can be seen that algorithm `OL_MT` delivers much better social welfare than those of algorithms `Online_A` and `Online_DA`. For example, when there are 100 requests, `OL_MT` has around 27.5%, 21.6%, and 7.9% higher social welfare than that of algorithms `Online_A`, `Online_DA`, and `OL_SC`. From Fig. 8 (c), we can see that `OL_SC` takes more running time than those of `OL_MT`, `Online_A` and `Online_DA`, since `OL_SC` invokes `FLG_SC` and takes more time in the topological sorting on the service graph of each request.

Fig. 9 illustrates the difference of different algorithm performance in the test-bed and the simulation experiments. From Fig. 9 (a), we can see that algorithm `OL_MT` in the testbed has a higher delay jitter compared with that in the simulation environment. The average delay of `OL_MT` in the testbed is higher than that in the simulation, as shown in Fig. 9 (b). The reason is that the testbed is deployed in a real network environment with a large amount of background traffic and control data across the network, the data queue or buffer in the network causes delay jitter and increases the average delay. This also leads to more delayed violations when the algorithm runs in testbed, which reduces the social welfare, as shown in Fig. 9 (c). As shown in Fig. 9 (d), `OL_MT` takes a longer running time in the testbed than that in simulation. This is due to that algorithm `OL_MT` needs to handle more packets related to network protocols, such as PacketIn and probe packets in OpenFlow [39].

VIII. Conclusion and Future Works

In this paper, we studied the (online) social welfare maximization problem for an NFV market in a multi-tier cloud network. We first devised an efficient incentive-compatible mechanism and analyzed the existence of a Nash Equilibrium for the problem. We then designed a contextual MAB based method for the online social welfare maximization problem with uncertain values of customers and without the knowledge of future request arrivals. We finally evaluated the performance of the proposed mechanisms by simulations and test-bed implementations. Results show that the performance of the proposed mechanisms obtain 27% higher social welfare than those of existing studies.

Built upon the work in this paper, we will consider a more generic setting of the problem, where multiple infrastructure providers provide resources for network service providers as our future work. We will focus on the design of stable and near-optimal algorithms for a hierarchy NFV market. In addition, the proposed facility location game also allows wide applications with similar optimization objectives. For example, the facility location game can also be extended to a general social welfare maximization problem with different types of players, where, there may have large-scale network service providers leading the market, and small-scale network service providers may just follow the strategies of these large-scale ones. Thus, a generic pricing method dealing with different types of players can be included in the proposed facility location game.

Acknowledgment

The authors appreciate the three anonymous referees and the associate editor for their constructive comments and valuable suggestions, which helped them improve the quality and presentation of the paper greatly.

References

[1] *5G Services Market Share Analysis With Recent Developments, Business Prospects and Forecast 2022–2031*. Accessed: May 2019. [Online]. Available: https://www.marketsandmarkets.com

[2] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2011, pp. 1–9.

[3] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1943–1951.

[4] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 127–135.

[5] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware VNF placement and chaining based on a flexible resource allocation approach," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, 2017, pp. 1–7.

[6] R. Allesiardo, R. Féraud, and D. Bouneffouf, "A neural networks committee for the contextual bandit problem," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*. Kuching, Malaysia: Springer, 2014, pp. 374–381.

[7] Amazon Web Services. *Amazon EC2 Instance Configuration*. Accessed: May 2020. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html

[8] J. W. Anderson, R. Braud, R. Kapoor, G. Porter, and A. Vahdat, "XoMB: Extensible open middleboxes with commodity servers," in *Proc. 8th ACM/IEEE Symp. Archit. Netw. Commun. Syst.*, Oct. 2012, pp. 49–60.

[9] S. Anderson, A. Palma, and J. Thisse, *Discrete Choice Theory of Product Differentiation*. Cambridge, MA, USA: MIT Press, 1992.

[10] W. Borjigin, K. Ota, and M. Dong, "In broker we trust: A double-auction approach for resource allocation in NFV markets," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1322–1333, Dec. 2018.

[11] J. Cardinal and M. Hoefer, "Non-cooperative facility location and covering games," *Theor. Comput. Sci.*, vol. 411, nos. 16–18, pp. 1855–1876, 2010.

[12] R. Cziva and D. P. Pezaros, "Container network functions: Bringing NFV to the network edge," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 24–31, Jun. 2017.

[13] *Data Plane Development Kit*. Accessed: Aug. 2021. [Online]. Available: https://www.dpdk.org/

[14] M. Dieye, W. Jaafar, H. Elbiaze, and R. H. Glitho, "Market driven multidomain network service orchestration in 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1417–1431, Apr. 2020.

[15] M. Dimakopoulou, Z. Zhou, S. Athey, and G. Imbens, "Balanced linear contextual bandits," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3445–3453.

[16] X. Fei, F. Liu, H. Xu, and H. Jin, "Towards load-balanced VNF assignment in geo-distributed NFV infrastructure," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–10.

[17] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[18] M. Ficco, C. Esposito, F. Palmieri, and A. Castiglione, "A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation," *Future Gener. Comput. Syst.*, vol. 78, no. 1, pp. 343–352, 2018.

[19] C. K. K. Fong, M. Li, P. Lu, T. Todo, and M. Yokoo, "Facility location games with fractional preferences," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[20] M. Goemans and M. Skutella, "Cooperative facility location games," *J. Algorithms*, vol. 50, no. 2, pp. 194–214, 2004.

[21] S. Gu, Z. Li, C. Wu, and C. Huang, "An efficient auction mechanism for service chains in the NFV market," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[22] A. Gupta *et al.*, "SDX: A software defined internet exchange," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2014, pp. 551–562.

[23] F. Hao, M. Kodialam, T. Lakshman, and S. Mukherjee, "Online allocation of virtual machines in a distributed cloud," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 238–249, Jul. 2017.

[24] M. M. Hassan, M. S. Houssan, A. M. J. Sarkar, and E.-N. Huh, "Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform," *Inf. Syst. Frontiers*, vol. 16, no. 4, pp. 523–542, 2014.

[25] N. He, S. Yang, F. Li, S. Trajanovski, F. A. Kuipers, and X. Fu, "A-DDPG: Attention mechanism-based deep reinforcement learning for NFV," *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQoS)*, Jun. 2021, pp. 1–10.

[26] H. Huang *et al.*, "Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2558–2571, Jun. 2021.

[27] P. Jin, X. Fei, Q. Zhang, F. Liu, and B. Li, "Latency-aware VNF chain deployment with efficient resource reuse at network edge," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 267–276.

[28] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Sep. 2011.

[29] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, pp. 83–97, Mar. 1955.

[30] S. G. Kulkarni *et al.*, "NFVnice: Dynamic backpressure and scheduling for NFV service chains," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 639–652, Feb. 2020.

[31] L. Li, W. Chu, J. Langford, and R. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 661–670.

[32] Y. Li, L. T. X. Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *Proc. IEEE 35th Annu. Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[33] L. Liang, Y. Wu, and G. Feng, "Online auction-based resource allocation for service-oriented network slicing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8063–8074, Jun. 2019.

[34] J. Liu, H. Xu, G. Zhao, C. Qian, X. Fan, and L. Huang, "Incremental server deployment for scalable NFV-enabled networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 2361–2370.

[35] Z. Luo and C. Wu, "An online algorithm for VNF service chain scaling in datacenters," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1061–1073, Mar. 2020.

[36] T. Madi, H. A. Alameddine, M. Pourzandi, and A. Boukhtouta, "NFV security survey in 5G networks: A three-dimensional threat taxonomy," *Comput. Netw.*, vol. 197, Oct. 2021, Art. no. 108288.

[37] J. Martins *et al.*, "ClickOS and the art of network function virtualization," in *Proc. 11th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2014, pp. 459–473.

[38] J. Martín-Peréz, F. Malandrino, C. F. Chiasserini, and C. J. Bernardos, "OKpi: All-KPI network slicing through efficient resource allocation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 804–813.

[39] N McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[40] H. Moulin, "On strategy-proofness and single peakedness," *Public Choice*, vol. 35, pp. 437–455, Jan. 1980.

[41] *NFV IN ETSI*. Accessed: Jul. 2021. [Online]. Available: https://www.etsi.org/technologies/nfv

[42] *ONOS SDN Controller*. Accessed: Jul. 2021. [Online]. Available: https://wiki.onosproject.org/

[43] *Open vSwtich*. Accessed: Jul. 2021. [Online]. Available: https://www.openvswitch.org

[44] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016.

[45] H. Ren *et al.*, "Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2050–2066, Sep. 2020.

[46] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, Jul. 2016.

[47] X. Shang, Y. Huang, Z. Liu, and Y. Yang, "Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme," *IEEE Trans. Mobile Comput.*, early access, Jan. 1, 2021, doi: 10.1109/TMC.2020.3048885.

[48] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "NFP: Enabling network function parallelism in NFV," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2017, pp. 43–56.

[49] Q. Xia, W. Ren, Z. Xu, P. Zhou, W. Xu, and G. Wu, "Learn to optimize: Adaptive VNF provisioning in mobile edge clouds," in *Proc. 17th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2020, pp. 1–9.

[50] R. Xia, H. Dai, J. Zheng, R. Gu, X. Wang, and G. Chen, "SAFE: Service availability via failure elimination through VNF scaling," in *Proc. 48th Int. Conf. Parallel Process.*, 2019, pp. 1–10.

[51] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proc. IEEE*, vol. 102, no. 1, pp. 11–31, Dec. 2014.

[52] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2866–2880, Oct. 2016.

[53] Z. Xu *et al.*, "Near optimal and dynamic mechanisms towards a stable NFV market in multi-tier cloud networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10.

[54] Z. Xu *et al.*, "Affinity-aware VNF placement in mobile edge clouds via leveraging GPUs," *IEEE Trans. Comput.*, vol. 70, no. 12, pp. 2234–2248, Dec. 2021.

[55] B. Yi, X. Wang, M. Huang, and A. Dong, "A multi-criteria decision approach for minimizing the influence of VNF migration," *Comput. Netw.*, vol. 159, pp. 51–62, Aug. 2019.

[56] R. Yu, G. Xue, V. T. Kilari, and X. Zhang, "Network function virtualization in the multi-tenant cloud," *IEEE Netw.*, vol. 29, no. 3, pp. 42–47, Jun. 2015.

[57] S. Zaman and D. Grosu, "Combinatorial auction-based allocation of virtual machine instances in clouds," *J. Parallel Distrib. Comput.*, vol. 73, no. 4, pp. 495–508, 2013.

[58] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Mar. 1996, pp. 594–602.

[59] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Online stochastic buy-sell mechanism for VNF chains in the NFV market," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 392–406, Feb. 2017.

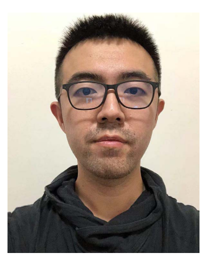
[60] G. Zhao, H. Xu, J. Liu, C. Qian, J. Ge, and L. Huang, "SAFE-ME: Scalable and flexible middlebox policy enforcement with software defined networking," in *Proc. IEEE 27th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2019, pp. 1–11.

[61] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware VNF deployment and migration for 5G network slice," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2115–2128, May 2021.

[62] J. Zheng, G. Chen, S. Schmid, H. Dai, and J. Wu, "Chronus: Consistent data plane updates in timed SDNs," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2017, pp. 319–327.

[63] D. Zheng, C. Peng, X. Liao, L. Tian, G. Luo, and X. Cao, "Towards latency optimization in hybrid service function chain composition and embedding," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1539–1548.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees from the Dalian University of Technology, China, in 2008 and 2011, respectively, and the Ph.D. degree from Australian National University in 2016, all in computer science. From 2016 to 2017, he was a Research Associate with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a Full Professor and a Ph.D. Advisor with the School of Software, Dalian University of Technology. He is also a "Xinghai Scholar" with the Dalian University of Technology. His research interests include mobile edge computing, serverless computing, networks function virtualization, software-defined networking, algorithmic game theory, and optimization problems.



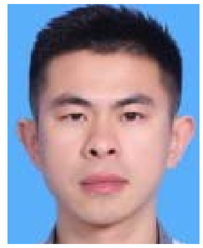
Haozhe Ren (Student Member, IEEE) received the B.Sc. degree from the University of Science and Technology Beijing, China, in 2012, and the M.E. degree from Xinjiang Normal University, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Software, Dalian University of Technology. His current research interests include networks function virtualization, software-defined networking, algorithmic game theory, and optimization problems.

Weifa Liang (Senior Member, IEEE) received B.Sc. degree from Wuhan University, China, in 1984, the M.E. degree from the University of Science and Technology of China in 1989, and the Ph.D. degree from Australian National University in 1998, all in computer science. He is a Professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a Professor with Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless *ad-hoc* and sensor networks, mobile edge computing, networks function virtualization, the Internet of Things, software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS.

Qiufen Xia (Member, IEEE) received the B.Sc. and M.E. degrees from the Dalian University of Technology, China, in 2009 and 2012, respectively, and the Ph.D. degree from Australian National University in 2017, all in computer science. She is currently an Associate Professor with the Dalian University of Technology. Her research interests include mobile cloud computing, query evaluation, big data analytics, big data management in distributed clouds, and cloud computing.

Wanlei Zhou (Member, IEEE) received the B.E. and M.E. degrees in computer science and engineering from the Harbin Institute of Technology, Harbin, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science and engineering from Australian National University, Canberra, Australia. He is currently the Vice Rector (Academic Affairs) and the Dean of the Institute of Data Science, City University of Macau, Macau, China. His research interests include distributed systems, networks security, and privacy preservation.

Pan Zhou (Senior Member, IEEE) received the B.S. degree from the Advanced Class of the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, the M.S. degree from the Department of Electronics and Information Engineering, HUST, in 2008, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, USA, in 2011. He was a Senior Technical Member at Oracle Inc., USA, during 2011 to 2013; and worked on Hadoop and distributed storage systems for big data analytics at Oracle Cloud Platform. He is currently a Full Professor and a Ph.D. Advisor with the Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, HUST. His current research interests include security and privacy, big data analytics, machine learning, and information networks. He held honorary degree in his bachelor's degree and Merit Research Award of HUST in his master's study. He received the "Rising Star in Science and Technology of HUST" in 2017. He is currently an Associate Editor of IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.



Wenzheng Xu (Member, IEEE) received the B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He was a Visitor at both the Australian National University, Australia; and The Chinese University of Hong Kong, Hong Kong. He is currently an Associate Professor with Sichuan University, China. His research interests include wireless *ad-hoc* and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Guowei Wu received the Ph.D. degree from Harbin Engineering University, China, in 2003. He is currently a Professor with the School of Software, Dalian University of Technology (DUT), China. He has published over 100 papers in journals and conferences. His research interests include embedded real-time systems, cyber-physical systems (CPS), and smart edge computing.

Mingchu Li received the B.S. degree in mathematics from Jiangxi Normal University, Nanchang, China, in 1983, the M.S. degree in applied science from the University of Science and Technology Beijing, Beijing, China, in 1989, and the Ph.D. degree in mathematics from the University of Toronto, Toronto, ON, Canada, in 1998. He was an Associate Professor with the University of Science and Technology Beijing from 1989 to 1994. From 2002 to 2004, he was a Full Professor with the School of Software, Tianjin University, Tianjin, China. Since 2004, he has been a Full Professor with the School of Software, Dalian University of Technology, Dalian, China. His main research interests include mobile edge computing, theoretical computer science and information security, trust models, and game theory.