先進的仮想化ネットワークのための効率的なマルチリソース割当戦略

Efficient Multi-resource Allocation Strategies for Emerging Virtualized Networks

Wuyunzhaola

Department of Sciences and Informatics
Muroran Institute of Technology

This dissertation is submitted for the degree of
Doctor of Philosophy of Engineering

March 2022

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Wuyunzhaola
March 2022

Acknowledgements

Five years of studying for a PhD are coming to an end in this season, but I reached another stage in my life. I would like to express my gratitude to all those who helped me during the writing of this thesis.

First of all, I would like to extend my sincere gratitude to my supervisor, Prof. Dong and Prof. Ota, for their instructive advice and useful suggestions on my thesis. Without them consistent and illuminating instruction, the thesis could not have reached its present form.

Next, I would like to express my heartfelt gratitude to Prof. Li, who have instructed and helped me a lot in the past five years. He has spent much time reading through each draft and provided me with inspiring advice. I also owe a special debt of gratitude to the rest of my thesis committee: Prof. Watanabe and Prof. Jin, for many suggestive comments to improve the quality of this thesis.

In addition, I deeply appreciate the contribution to this thesis made in varios ways by my friends and ENeS lab members.

Last but not least, my gratitute also extends to my family who have been assisting, supporting and caring for me all of my life.

Wuyunzhaola
March 2022

Abstract

Network function virtualization (NFV) is an emerging scheme to provide virtualized network function (VNF) services for next-generation networks. However, finding an efficientway to distribute different resources to customers is difficult. In the first half of this thesis, we deal with the resource allocation from two aspects. First, we develop a new double-auction approachnamed DARA that is used for both service function chain (SFC) routing and NFV priceadjustment to maximize the profits of all participants.To the best of ourknowledge, this is the first work to adopt a double-auction strategy in thisarea. The objective of the proposed approach is to maximize the profits ofthree types of participants: NFV broker, customers and serviceproviders. Second, multiple Walrasian Auction Graphic Model of different bundled tree nodes is proposed to maximise the social effectiveness. It is the first work to define the virtualized service as a tree valuation in NFV market. Moreover, We propose the relevant proof of the algorithms to ensure the correctness of the method. With all theory taken into account, we conducted a comprehensive simulation to evaluate the proposed method.

In the second the thesis, we schedule other resources including deep learning as a service and coflow scheduling for Data Center. Firstly, deep learning combining with cloud computing is a surging technology recently which is a new paradigm called DLAS (deep learning as a service). We formulate a competitive market between a provider and users in cloud computing and propose two efficient decision and pricing strategies called Dealer strategies for users and the provider, respectively. Secondly, we schedule coflow which is a collection of parallel flows, while a job consists of a set of coflows. We take the dependency of coflows into consideration. To guarantee job completion for performance, we formulate a deadline and dependency-based model called MTF scheduler model with the constraints of deadline and network capacity. We consider the dependent coflows as an entirety and propose a valuable coflow scheduling first MTF algorithm. Finally, we conduct extensive simulations to evaluate our methods for deep learning as a service and coflow scheduling in Data Center.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

In this chapter, we will introduce our research background by three aspects including double auction based NFV market, multi-walrasian tree valuation and scheduling other resources.

1.1 Double auction based NFV market

Network function virtualization (NFV) is a new scheme for enhancing network scalability and flexibility [1, 48, 6, 74]. Due to the high costs of providing storage space and computational resources for traditional network methods, it is becoming difficult to provide new full services in the current networks [102, 12]. The development of NFV introduces new approaches, along with other emerging technologies, such as software-defined networks (SDNs) [62, 57, 61] and cloud computing [94], to design, schedule and manage network resources. NFV changes the traditional rules for how network operators manage their infrastructure as software instances separate from the hardware platform by using the proven virtualization technology [7]. For example, one example of an open platform for NFV (OPNFV), the Huawei E9000 blade server, is widely applied in industry and is intended to facilitate the commercial adoption of NFV applications [42].

Virtualized network functions (VNFs) defined by NFV are virtualized tasks that are separated from network hardware, which is provided by network service providers. In fact, NFV distributes VNFs, including firewall, storage and routing, executed on commodity hardware, as shown in Figure 1.1, because a single VNF instance is not enough to providing a valid service to the required customers [100]. Multiple VNFs, instantiated without delay and equipment installation, can be connected to obtain chains of network services, called a service function chain (SFC) [11], which is made-to-order for different use cases [95, 106, 28]. Accompanying SFCs, an NFV market is an emerging scheme in which the SFC broker has geo-distributed information of SFCs and then sells them to users on demand, while SFC suppliers provide one or more SFCs. The participants, including customers, the SFC broker and service suppliers, play a game with each other for their own benefits.

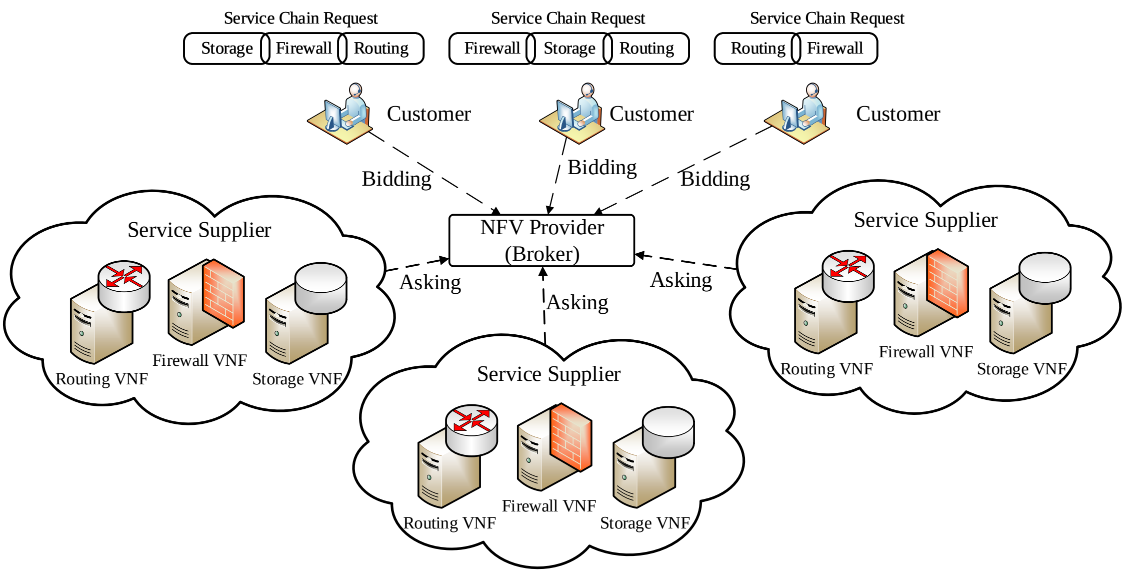


Fig. 1.1 Double auction between service suppliers and customers in an NFV market

NFV provider SFC provides the capability to dynamically functions in a network processing path in NFV market. The concept is shown in Figure 1.2. Each circle represents a different service network function that is connected to other services via a network. The arrows represent SFCs that comprise a particular set of service functions connected in order. For example, when a user's demand is arranged according to the red arrow in Figure 1.2, the NFV provider can provide it to the user according to the user's demand and charge a reasonable price. And the price of the payment is the critical value for both sides to get the maximum return. On this basis, VNFs may have many different topologies, such as a service function tree. This VNF topology is very useful in solving service function sharing and two-way service function chain in different scenarios. According to different functions, the topological state of VNF can be fine-grained or coarse-grained.

The development of NFV faces also several technical issues in handling VNF, which is the most important component of an SFC. Previous works [96, 31] have shown that although the underlying network is lightly utilized, virtualization may still lead to performance problems such as abnormal latency variations and severe throughput jitter. Therefore, the first problem of NFV is that the hardware and software may be supported by different service providers, resulting in skewed utilization, increased latency or unstable throughput. The second problem is that when multiple suppliers manage the virtualized resources in the network, it is difficult to coordinate with suppliers to provide good service performance [8, 44].

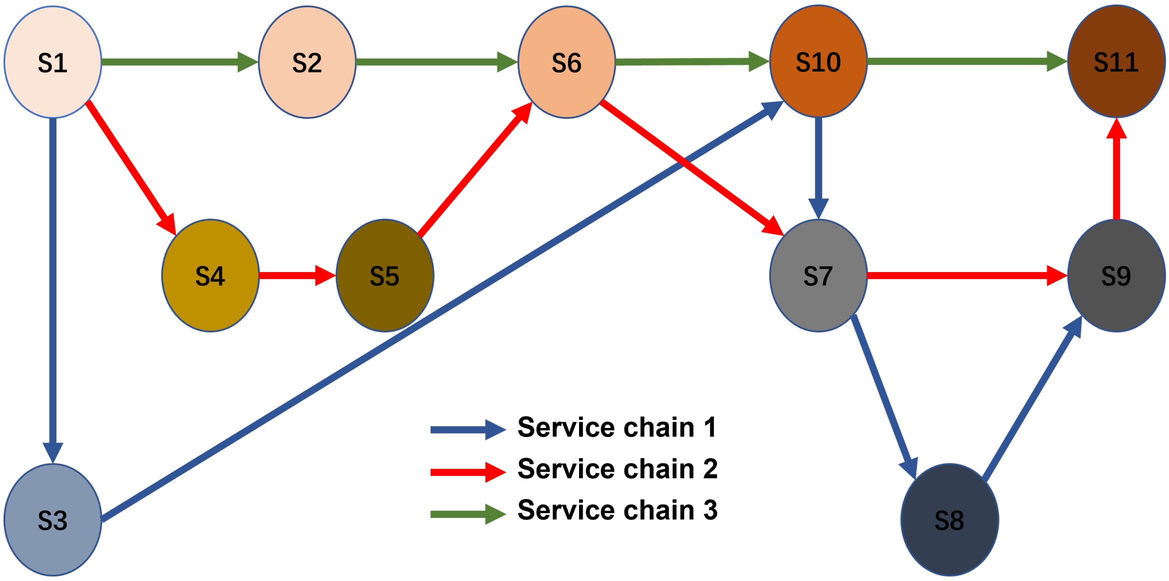


Fig. 1.2 Service function chain concept

To solve these problems, some industrial projects from commercial companies attempt to define standards for the coordination of suppliers [46, 114, 103]. However, the resource utilization and service performance of these methods are not sufficient to support the distribution of SFC in NFV market due to inefficient scheduling strategies. Finding an efficient method for scheduling available SFCs among independent suppliers and to calculate the applicable price in the NFV market is challenging. In general, an auction-based method can improve the efficiency of resource scheduling in a competitive environment [30]. Compared with simple allocation using fixed pricing, an auction mechanism provides more economical efficiency for suppliers according to customer demands, flexible allocation of SFCs and finer targeting of customers.

A double-auction mechanism can model the interaction of two or three parties well, where buyers request SFCs with the bidding price, suppliers provide their services with the asking price, and the broker decides the transaction value [92]. Through competitive bidding and asking, the profit in the double-auction method is higher than that in the single-auction method.

The main contributions of this subsection can be summarized as follows.

1) First, we propose the concept of the double-auction NFV market and characterize the mutual effect between the SFC broker, customers and resource suppliers. With this concept, we can combine a large number of VNFs into different service chains and then schedule them separately for a customer in an NFV market.

2) We formulate a double-auction model with constraints of customers and sellers to maximize the profits of the three participants. To solve this model, we use three algorithms, including auction, price adjustment and payment strategy, to schedule network resources. We combine the normal distribution element and the price adjustment to control the auction progress. We also theoretically analyze the effectiveness of our proposed method.

3) We conduct a comprehensive simulation to evaluate the DARA resource scheduling method. The results confirm that the DARA method outperforms the single-auction model with respect to the profits of both participants.

1.2 Multiple-Walrasian tree valuation market

As one of emerging methods the strength of Network Function Virtualization (NFV) lies in to schedule network service resource and improve the expandability and flexibility of such resources[38, 6, 74, 58]. In a traditional network, it is difficult to improve service quality due to the limited condition in regards to storage and computing power [102, 12]. In the recent years, many researchers studying NFV discovered a new method to design and schedule network resource, namely by separating hardware platform and software[94, 7].

In NFV network, network service providers provide the smallest functional unit called virtualized network function (VNF) to handle virtualized tasks from network hardware.There are three types of network service resources including firewall, storage, and routing. The providers sell the bundled VNFs to the required users [100]. Based on these different requests from users, VNFs would be combined with each other for users, which is made-to-order for different tasks [95, 28]. In an NFV auction market, there are three parties including an auctioneer, service provider and users. The auctioneer has geo-distributed information of service providers and users. Service providers have both service resources as bundled or separated VNFs to sell. Users demand tasks handled by the number of appropriate VNFs. The three parties are actively involved to sustain their own profit.

The emergence of NFV technologies faces several challenges to schedule VNFs, which is the smallest unit in the NFV market. Selecting VNF suppliers among service providers[106, 67], allocating resources with respect to maximizing profit of the VNF sales[73, 60], as well as finding the best routing path Data Center to minimize traffic time [65, 8, 44] are three of the major challenges in the industry.

In response to these challenges, some previous works proposed that the efficient iterative auction based on the valuation is an effective method to schedule resources, such as gross substitutes or complementaries and tree valuation. Auction-based on Walrasian price is an efficient auction mechanism with the same supply and demand from providers and customers [46, 114, 32]. However, the previous methods can not efficiently schedule VNFs in NFV market because the traditional Walrasion auction is a single auction mechanism in which an item has multiple competitors in the single auction to buy it. However, in our Multiple-Walrasian model, in which multiple items has multiple competitors, we have combined the the traditional Walrasian auction and double auction. A bundled VNFs based on the tree valuation iterative double auction is applied in this . Tree valuation is defined by non-cycled tree graph in which the valuation of the bidder is considered as a node in a tree [77]. In general, a tree valuation method based on the double-auction can effectively improve the VNF distribution in a fair condition.

We can summarize main contributions in this subsection as follows.

1) We introduce definitions and basic assumptions of Multipl-Walrasian auction mechanism and our model to guarantee the mechanism based on the tree valuation in the graph. Then three assumptions are proposed to simplify the conception of the tree valuation and to meet the rule of Multipl-Walrasian auction through VCG payment. With the concept taken into consideration, different VNFs are applied to the bundled tree nodes and priced for each users.

2) We apply Multipl-Walrasian double-auction model based on the graphical theory with constraints of the decision of the participants to maximize the social welfare. Novel algorithms are proposed to schedule tree valuations. Then we prove the feasibility and dominance of new algorithms.

3) We propose the numerical results are in agreement with the analysis through comprehensive simulations. From the results, our method is more effective than Backpack model and Reserve model to maximize the social welfare.

1.3 Scheduling other resources

In this subsection, we will introduce background about deep learning as a service and scheduling coflow in Data Center.

1.3.1 Deep learning as a service

Recent years, with the rapid development of cloud computing, SAAS (software as a service) applications are gaining more and more tendency. SAAS have changed the pattern of utilization of the traditional software greatly [4, 13]. With the coming of SAAS, the performance of the cloud computing is enhanced with the great cost-saving, the time-saving with startup, the less OPEX (operating expense) and so on [41]. SAAS also brings the chance with the higher revenue of the cloud provider based on users' increasing requests for the SAAS application services and the decreasing costs for the resource allocation.

Deep learning combining with cloud computing is a surging technology recently, which is a new paradigm called DLAS (deep learning as a service) [41, 104]. To guarantee the utilization of resource allocation and the lower costs of the application, a cloud provider provides multiple resources in physical servers, such as CPU (central processing unit), GPU (graphics processing unit) and TPU (tensor processing unit). To achieve the high performance of the independence and the security, more and more providers use the VM (virtual machine) in cloud computing for the virtual resources utilization. Therefore, a large number of virtual processors including VCPU (virtual CPU) and VGPU (virtual GPU) is coming and running on the VMs[57, 63].

The Graphics Processing Unit (GPU) is an important microprocessor to process figure for high performance computing in deep learning. GPU can accelerates efficiently the computation including Artificial Intelligence (AI) and cloud graphics. Recently, GPU virtualization is an emerging scheme to optimize the utilization of GPU resources for good performance. With the approach of GPU virtualization, virtual machines (VMs) can schedule the VGPU into applications. It is important to schedule VGPUs and VCPUs into applications to guarantee the high utilization.

Google first deploys the concept of TPU in 2015 in Data Centers [34]. TPU can accelerate the speed of the logical deduction of NN (neural networks) in machine learning [79]. With the increasing virtualization of CPU and GPU, the users are demanding more and more virtualization of professional computing chip including TPU and FPGA (field programmable gate array).

The main contributions of this part can be summarized as follows.

1) First, we propose the concept of the competitive market including leader and followers. We characterize the mutual effect between the provider and users. With this concept, we consider the provider and users as a leader and followers, and then the provider schedule resources separately for users in market.

2) We formulate a multi-objective model called MRAP model with constraints of the provider and users. To solve this model, we propose two pricing algorithms called Dealer, including the decision strategy for users and the pricing strategy for the provider.

3) We conduct a comprehensive simulation to evaluate the Dealer pricing strategy. The results confirm that MRAP model outperforms the Elastic pricing strategy model and Amazon EC2 with respect to the revenues of both participants.

1.3.2 Coflow scheduling for Data Center

Data Centers have been used as the critical computing platforms for cloud computing, such as MapReduce[26, 57], Spark[108] and Dryad[51]. These frameworks have communication patterns that a large number of flows transfer through a set of computation nodes. The completion of a job is considered as all of the coflows have finished before their deadlines[64]. Application performance is largely affected by the job completion time. The completion time of coflow affects the job performance which is important for service quality. The service quality limits their profit performance directly, even a minor improvement of the completion time will increase the revenue of providers. In this case, improving the job completion time is important for these frameworks. However, such frameworks mainly focus on flow level requirement guarantee while ignoring the entirety of a job. Therefore, the joint of dependency consideration among coflows is becoming increasing important for providing deadline basic performance guarantee.

Coflow is an abstraction of a set of flows which are brought in to ignore the gap between flow and job[21]. Coflow can be seen as a set of flows between two stages of a job such as a shuffle in MapReduce. To improve service performance, we need to satisfy the job requirement in a dependent way instead of individual flows. For example, in MapReduce framework, we take the first stage as Map phase, and the Reduce phase as the second stage. Hence the Reduce phase can not start before stage one finished. In addition, what if one flow in the coflow lags the other completion time, the whole job will be affected by this lag flow. Therefore, we take the coflows as an entirety to improve overall completion time. The research focuses on scheduling dependent coflows in Data Center network and mainly makes these following contributions:

- We address the dependency among coflows to guarantee service performance on job deadline. With the dependency of coflows, we can divide coflows into different collections and schedule them separately to provide deadline guarantee.
- We formulate a dependent coflow model with constrains of deadline and network capacity. To solve this model, we resort to MTF method to schedule the valuable coflow first. Our proposed method turns out to be effective by theoretical analysis.

- We conduct a comprehensive simulation to evaluate MTF coflow scheduling method. We compare MTF method with short job first method by constraints of deadline. The results confirm that MTF method outperforms on job completion time and the ratio of job completion.

In summary, in this thesis, we will work from these above parts one by one to solve each challenge.

Chapter 2

In Broker We Trust: A Double-auction Approach for Resource Allocation

In this chapter, we present an auction-based resource scheduling method for guaranteeing resource utilization and service performance in the NFV market. Although the traditional single-auction method can improve the resource utilization to schedule SFCs, it cannot guarantee the profits of service suppliers. Rather than using single-auction methods, a double-auction model can achieve a higher efficiency with competitive bidding between customers and service suppliers. Consequently, we first design an efficient double-auction model in our resource scheduling method in which both service suppliers and customers can participate in the auction market. The main goal of our model is to maximize the profits of the three participants when the auction mechanism is incentive compatibility (IC).

The remainder of this chapter is organized as follows. The related works are discussed in Section 4.1. We present the related preliminaries in Section 2.2. We introduce the resource allocation problem in Data Center networks. We also propose the DARA resource scheduling and pricing method with an NFV performance constraint in Section 4.2. In Section 4.3, we design a double-auction mechanism containing the DARA algorithm and a price adjustment to guarantee availability. We present some numerical results in Section 4.4, and then we demonstrate the feasibility of the proposed DARA method through a mathematical analysis and comparison with the single-auction model.

2.1 Related Work

Several large industrial projects, such as the European Telecommunications Standards Institute (ETSI) and the Internet Engineering Task Force (IETF), propose industry standards on

NFV in the form of white papers [2]. To solve the problem of data traffic, the SFC working group of IETF [11] finds a dynamic approach with a series of network functions to guide the physical or virtualized data traffic. The resource scheduling in this chapter follows the framework and assumptions in these white papers.

Gember et al. [43] propose programming a network-aware orchestration layer called Stratos to deploy middleboxes in the cloud for a virtual middlebox appliance. Stratos's process consists of three phases: determining the number of VNFs in each type, deciding a better position for each type of VNF in the cloud, and guiding the data traffic through service chains. To solve the placement problem of VNFs, Ming Xia et al. [101, 100] find a heuristic algorithm that can be efficiently operated with binary integer programming (BIP). Moreover, in their study, it is possible to minimize the cost of optical-electronic-optical (O-E-O) conversions by using NFV chains in optical Data Center.

Prior works mainly focus on the VNF deployment problem from the perspective of resource allocation. In fact, social welfare and resource market are the other mechanisms for providing good service performance to competing customers, and auctions have been regarded as a primary method.

Bari et al. [5] present two methods to solve the VNF orchestration problem (VNF-OP). The first method is an integer linear programming (ILP) formulation with an implementation in IBM ILOG CPLEX Optimization Studio (CPLEX) for small-scale networks. Second, for large-scale networks, they also propose a heuristic algorithm based on dynamic programming. Double-auction methods are widely used in distributing resources between competing customers and resource suppliers. In [91], the authors propose an intelligent resource allocation mechanism by building a double combinatorial auction model based on a reputation system to avoid malicious behavior. They also present a price decision mechanism based on a backpropagation (BP) neural network to make decisions scientifically. In [9], the authors introduce new resource allocation mechanisms for three type of participants: providers, tenants and end users.

To clearly minimize the costs of capital and operation, SDN is first proposed based on the wireless virtualization architecture, which can solve multiple flow transmission problems when there are multiple infrastructure providers and multiple mobile virtual network operators [112]. Moreover, by using a virtual resource allocation algorithm, the authors also solve an optimization problem of social welfare, improving the quality of service (QoS) requirements while reducing transaction costs. Dong et al. [29] find a new caching device named SRCMN to enhance the network performance under constrained conditions in an SDN-enabled network. In [46], the authors first propose an efficient and truthful auction method to distribute resources dynamically and to price the unit of transaction. To connect atomic network functions and provide integrated services, they define NFV service chains in a Data Center.

VNF orchestration and capital expense problems can be solved as an auction model, and the double-auction model is also an effective way to solve resource allocation problems. In [56], the authors present a novel double-auction scheme to protect the privacy of electric vehicles (EVs) and meet the requirement of demand response. In the double-auction market, the auctioneer matches buyers to sellers to achieve the maximum social welfare. As an auctioneer, the cloud protects the privacy between bidders and the auctioneer. Rather than the traditional double-auction methods, the authors [59] first adopt a game theory method to analyze the profits of the cloud provider and customers, as well as state the profit functions. Dou An et al. [3] propose a novel weakly dominant strategy-based on-line double-auction (SODA) method in the smart grid system to address the energy management issues with microgrids. The theoretical analysis proves that SODA can achieve high performance with (weak) budget balance and computational efficiency.

Fu et al. present a new type of core-selecting virtual machine (VM) combinatorial auction-based allocations [40] that can economically and efficiently calculate bidder charges from the core of the price vector space. In [111], the authors formulate the problem of allocating virtual resources as an optimization problem to maximize the total utility of the system. Then, they transform the transaction cost problem into an iterative double-auction problem. In this process, the bidding prices are changed by the iterative computation according to their own utility until the deal is closed. To solve the problems of SFC positioning and pricing, Zhang et al. [114] propose a novel auction mechanism in which the NFV provider owns resource information and customers can bid stochastically online. This mechanism significantly enhances the performance of existing techniques, while both sellers and buyers occupy or supply the VNF service chain in a limited time. M. Nazif Faqiry et al. [37] introduce a general double-auction scheme to solve the energy distribution problem among competing buyers, sellers and agents in a microgrid. They create a suitable projection objective function to maximize the total welfare of participants, while the agents can sell or procure energy with free bids in a selfish manner. By formulating a double-auction mechanism, our main objective is to maximize the profit of the SFC broker as an auctioneer in the NFV market. In the auction mechanism, we also strive to guarantee the profits of customers and resource suppliers.

2.2 Preliminaries

In this chapter, we consider a centralized SFC broker who collects resources from distributed service suppliers to obtain maximum overall profit. As shown in Figure 1.1, customers request a certain ordered chain of VNFs; meanwhile, more than one seller supplies resources for the required service chain. Therefore, the customers are capable of choosing the price for their requirements, while the SFC broker also desires a higher profit. The SFC broker collects all the available service chains and supplies them to the customers. Accordingly, we can formulate a double-auction problem in this market.

We present our basic definitions and assumptions to describe the formulation of the auction problem. We assume that there are two or more service suppliers who can provide some objects in a single cloud. Let $\mathscr{J}$ denote the set of sellers, numbered $1,2,...,j,...J$, and $\mathscr{J} = \{1,...,J\}$. Sellers face I buyers or potential buyers, numbered $1,2,...,i,...,I$. Let $\mathscr{I}$ represent the set of buyers, and $\mathscr{I} = \{1,...,I\}$. We use i to denote typical buyers in $\mathscr{I}$. In the auction, we assume that all sellers do not know the bidding and asking prices of other buyers and sellers. We provide a precise definition of the double-auction problem as follows.

Definition 1 *A market consists of sellers, buyers, and a broker, while a single auction consists of an auctioneer and many buyers.*

In a single-auction market, such as the English auction market, bidding prices are increasing, and each subsequent bidding price is greater than the previous one. If no buyer is willing to continue bidding, the buyer with the highest bidding price pays the bidding price and the auction ends. In a double-auction market, buyers first present their bidding price, and sellers submit their asking price to the auctioneer. Then, the auctioneer chooses the hammer price, denoted by p, which is decided by the asking price and bidding price. Finally, price p must satisfy the rule that the hammer price is higher than the bidding price and less than the asking price. Thus, we consider a double-auction approach for scheduling resources in the NFV market. We assume that the double-auction model in our chapter is the truthful auction model based on IC.

Definition 2 *An auction mechanism is IC if the dominant strategy for all customers is to reveal its true valuation, regardless of other buyers' bidding [45].*

Let b_{-i} denote the bidding price of a given buyer except buyer i. We use $P(b_i) : R \rightarrow [0,1]$ to denote the cumulative distribution function corresponding to the density $f_i(\cdot)$. Hence,

$$P(b_i) = \int_{-\infty}^{b_i} f_i(s_i) ds_i. \tag{2.1}$$

Theorem 1 *The equivalent condition of the truthful auction market is*

1. *The probability $P(b_i)$ of buyer i with price b_i is monotonically nondecreasing in b_i;*

2. *The charge p_i of buyer i is equal to*

$$p_i = b_i P(b_i) - \int_0^{b_i} P_{-i}(b)db, \tag{2.2}$$

where $P(b_i)$ is the probability that buyer i obtains the instance and $P_{-i}(b)$ is the winning probability of a given buyer except b_i [76, 14].

Definition 3 *A strategy for player $i \in \mathscr{I}$ is a map $s_i : \mathscr{B} \to S_i$, where s_i denotes an action for each player i, $\mathscr{B}$ denotes the set of bidding prices, and S_i is the strategy sequence set [86].*

Theorem 2 *The strategy function $s(\cdot)$ is a Bayesian Nash equilibrium (BNE) if, for all $i \in \mathscr{I}$ and for all $b_i \in \mathscr{B}$, we have the following equation:*

$$s_i(b_i) \in \arg\max_{s'_i \in S_i} \sum_{b_{-i}} f(b_{-i}|b_i) u_i(s'_i, s_{-i}(b_{-i}), b_i, b_{-i}), \tag{2.3}$$

where u_i is the utilization function of buyer i [90].

2.3 Double-auction Model

In this section, we describe the double-auction model for the NFV market. Participants in the market are customers, resource suppliers, and the SFC broker. Customers have one or more independent tasks for execution, and resource suppliers have available resources. The SFC broker possesses geodistributed information of SFCs and distributes and sells SFCs based on the demands of the customer.

We formulate the SFC distribution in the NFV network with a double-auction market to satisfy IC. There are K types of SFCs, such as routing, firewall, and storage. There are three assumptions in our model, as follows:

1. Each seller has enough SFCs for all buyers' requests in the truthful NFV auction market and sells the same SFC to different buyers.

2. The SFC broker only places those SFCs from sellers that satisfy requests of buyers. Therefore, in the auction process, the constraints of SFCs, such as delay or service capability, are always satisfied.

3. Buyers can purchase all required SFCs from the NFV market after the auction process. Meanwhile, each buyer only obtains one SFC from one seller.

Our model is formulated depending on Definition 2 and Theorem 1. Let b_i^k and a_j^k denote the bidding price and the asking price for the $k-$th SFC, respectively. Every buyer i has a private valuation denoted by d_i^k and a hammer price denoted by p_i^k for the $k-$th SFC that satisfy

$$b_i^k \leq p_i^k \leq d_i^k. \tag{2.4}$$

The profit of buyer i for $k-$th SFC is $u_i^k = d_i^k - p_i^k$; thus, the profit of buyer i is $u_i = \sum_{k=1}^{K} u_i^k$. Therefore, the total benefit of buyers is

$$U_{Buyer} = \sum_{i=1}^{I} \sum_{k=1}^{K} (d_i^k - p_i^k). \tag{2.5}$$

Seller j has a cost price denoted by c_j^k and a hammer price denoted by p_j^k for the $k-$th SFC that satisfy

$$c_j^k \leq p_j^k \leq a_j^k. \tag{2.6}$$

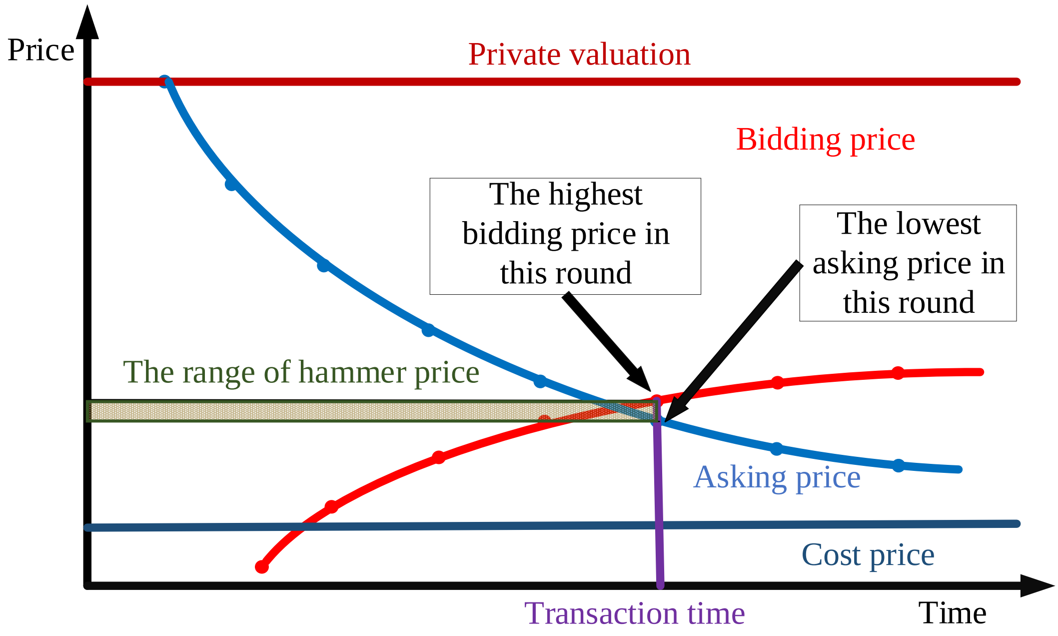
The profit of seller j for $k-$th SFC is $u_j^k = p_i^k - c_i^k$; thus, the profit of seller j is $u_j = \sum_{k=1}^{K} u_j^k$. Therefore, the total benefit of sellers is

$$U_{Seller} = \sum_{j=1}^{J} \sum_{k=1}^{K} (p_j^k - c_j^k). \tag{2.7}$$

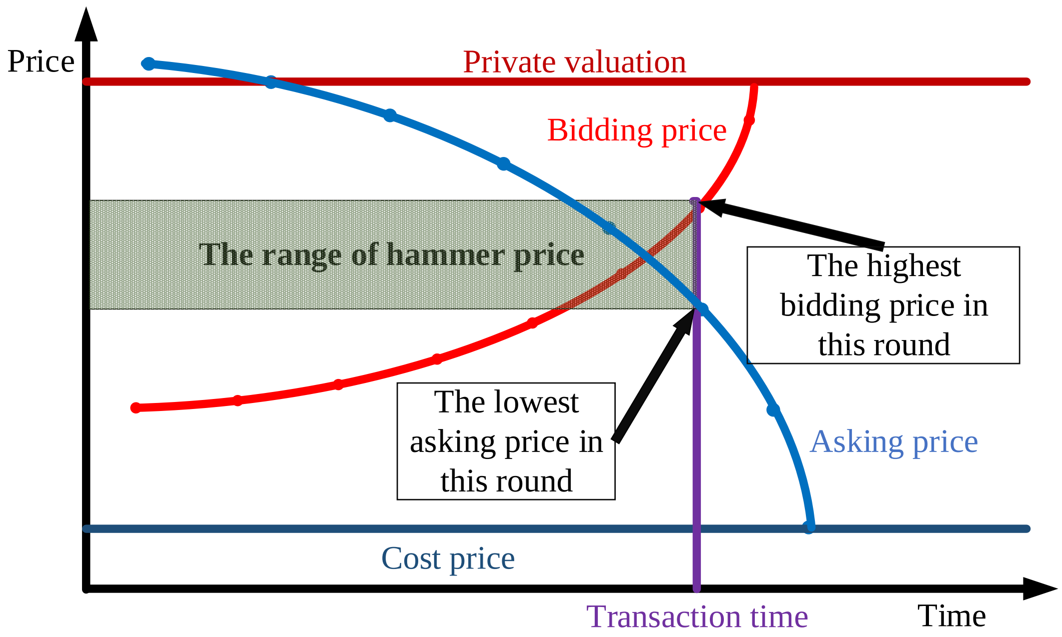
Consider the case in which buyer i submits their first bidding price b_i^k for the $k-$th SFC and seller j simultaneously submits their first asking price b_i^k for the $k-$th SFC. If $b_i^k \geq a_j^k$, buyer i and seller j make a deal for the $k-$th SFC with price p_{ij}^k decided by the SFC broker in the range of $[a_j^k, b_i^k]$. If $b_i^k < a_j^k$, buyer i and seller j have to adjust the initial price.

Buyer i has to increase their bidding price b_i^k, and seller j has to bring the price down until the deal ends, as shown in Figure 2.1a and Figure 2.1b, respectively. The rate of price adjustment in Figure 2.1a changes more slowly than the rate in Figure 2.1b. Thus, in these two price adjustments, the range of hammer price decided by the SFC broker is different. Clearly, the fast price adjustment is better for the broker to select the appropriate hammer price from the larger range. Therefore, we choose a fast price adjustment function from the normal distribution as

$$\tilde{a}_j^k(n+1) = a_j^k(n)(1 + \int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt), \text{ and} \tag{2.8}$$



(a) Slow price adjustment in double auction



(b) Fast price adjustment in double auction

Fig. 2.1 Price adjustment in double auction

$$\tilde{b_i^k}(n+1) = b_i^k(n)(1 - \int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt), \tag{2.9}$$

where $\tilde{a_j^k}$ and $\tilde{b_i^k}$ are the prices after the price adjustments, respectively. The number of auction rounds is denoted as n. Let μ denote a constant parameter, and when $t = \mu$, we can obtain the maximum number of trades. With a suitable parameter, we can achieve a tradeoff between the time of the trade and the profits of the three participants in the market.

Let r_{ij}^k denote whether buyer i and seller j make a deal ($r_{ij}^k = 1$) or do not make a deal ($r_{ij}^k = 0$). The profit of the SFC broker from buyer i and seller j for the $k-$th SFC is $(\tilde{b_i^k} - \tilde{a_j^k})r_{ij}^k$. Thus, the profit of the SFC broker is as follows:

$$U_{NFV} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} (b_i^k - a_j^k) r_{ij}^k. \tag{2.10}$$

Let γ denote the service charge per auction round to prevent the price adjustment of buyers or sellers from having an endless loop. When we acquire an appropriate service charge, buyers and sellers can only choose to adjust the original price to avoid overpaying for γ. Therefore, our objective is to maximize the profits of the three participants in this chapter as follows:

$$\max \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} (b_i^k - a_j^k) r_{ij}^k + 2N\gamma, \tag{2.11}$$

$$\max \sum_{k=1}^{K} \sum_{i=1}^{I} (d_i^k - p_{ij}^k) r_{ij}^k, \text{ and} \tag{2.12}$$

$$\max \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{i=1}^{I} r_{ij}^k}{J}, \tag{2.13}$$

where N is the total number of auction rounds.

Equation (2.12) is to maximize the profit of buyers, and Equation (2.13) is to maximize the trading ratio for sellers. From Equations (2.5) and (2.7), we can obtain the profits of buyers and sellers as

$$U_{Buyer} = \sum_{j=1}^{J} \sum_{k=1}^{K} (d_i^k - p_{ij}^k) r_{ij}^k - N\gamma, \text{ and} \tag{2.14}$$

$$U_{Seller} = \sum_{i=1}^{I} \sum_{k=1}^{K} (p_{ij}^k - c_j^k) r_{ij}^k - N\gamma. \tag{2.15}$$

Table 2.1 Summary of notations

Symbols	Description
I	The number of buyers.
J	The number of servers.
K	The number of SFCs.
$\mathscr{I}$	The set of buyer i.
$\mathscr{J}$	The set of seller j.
$\mathscr{K}$	The set of SFCs
b_i^k	The bidding price of the $i-$th buyer for the $k-$th SFC.
a_j^k	The asking price of the $j-$th seller for the $k-$th SFC.
d_i^k	The private valuation of the $i-$th buyer for the $k-$th SFC.
c_i^k	The cost price of the $j-$th seller for the $k-$th SFC.
p_{ij}^k	The hammer price of the $k-$th SFC between buyer i and seller j.
r_{ij}^k	r_{ij}^k equals 1 when buyer i and seller j have a deal for the $k-$th SFC, otherwise 0.
N	The total number of auction rounds.
γ	The service charge per auction round.
$P_i(b_i)$	The probability that buyer i achieves the resource with price b_i.
$P_{-i}(b)$	The probability that buyers except i achieve the resource with price b.
$f_i(\cdot)$	The probability density function of buyer i achieving resource.

To satisfy all requests from buyers, we assume that at least one buyer provides a bidding price that is higher than the minimum asking price, given by

$$\max_i b_i^k \geq \min_j a_j^k. \tag{2.16}$$

Finally, the profit maximization problem in the double-auction model is given by

$$\max \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} (b_i^k - a_j^k) r_{ij}^k + 2N\gamma, \tag{17.1}$$

$$\max \sum_{k=1}^{K} \sum_{i=1}^{I} (d_i^k - p_{ij}^k) r_{ij}^k, \qquad \forall j \in \mathscr{J} \tag{17.2}$$

$$\max \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{i=1}^{I} r_{ij}^k}{J}. \tag{17.3}$$

$s.t.$

$$\sum_{j=1}^{J} \sum_{k=1}^{K} (d_i^k - p_{ij}^k) r_{ij}^k - N\gamma \geq 0, \quad \forall i \in \mathscr{I}; \tag{17a}$$

$$\sum_{i=1}^{I} \sum_{k=1}^{K} (p_{ij}^k - c_j^k) r_{ij}^k - N\gamma \geq 0, \quad \forall j \in \mathscr{J}; \tag{17b}$$

$$r_{ij}^k \in \{0,1\}, \qquad \forall i, \in \mathscr{I}, j \in \mathscr{J}, k \in \mathscr{K}; \tag{17c}$$

$$b_i^k \leq p_i^k \leq d_i^k, \quad \forall i, \in \mathscr{I}, j \in \mathscr{J}, k \in \mathscr{K}; \tag{17d}$$

$$c_j^k \leq p_j^k \leq a_j^k, \quad \forall i, \in \mathscr{I}, j \in \mathscr{J}, k \in \mathscr{K}. \tag{17e}$$

2.4 Algorithm

In this section, we present three algorithms to optimize the profit in Data Center networks by adjusting the bidding price and asking price. In our model, customers require SFCs with corresponding VNFs, while service suppliers want the maximum profit. This problem is considered as a noncooperative game, which is proven to be a weakly dominant strategy in our algorithms. In Algorithm 1, we present the details of the double-auction process. Algorithm 2 is the process of price adjustment, which can output a feasible solution for guaranteeing the profits of every buyer and seller. Algorithm 3 illustrates the process of searching the real value of SFCs, and it is able to calculate the appropriate price to attract customers. We also prove that these three algorithms guarantee the performance of the auction process.

2.4.1 DARA Algorithm for ILP Problem (2.17)

Algorithm 1 DARA auction algorithm for ILP problem (2.17)

Input: The numbers of buyers, sellers and SFCs; the demand of buyers; the set of SFCs for sellers.

Output: The total profit of SFC.

for k in range(K) **do**
  $\mathscr{I} = 0$;
  for i in range (I) **do**
    if customer_demand(i,k) is truth **then**
      $\mathscr{I} = \mathscr{I} + i$;
    else
      $\mathscr{I} = \mathscr{I} + 0$;
    end if
  end for
  while any($\mathscr{I}$) **do**
    if max $b_i^k \geq$ min a_j^k **then**
      $i^* = \arg\max b_i^k$;
      $j^* = \arg\min a_j^k$;
      $price_{i^*j^*}^k \in [a(k,i^*), b(k,j^*)]$ decided by SFC broker ;
      $r_{i^*j^*}^k = 1$;
      $customer\_price_{i^*}^k = price_{i^*j^*}^k r_{i^*j^*}^k$;
      if $seller\_price_{j^*}^k \neq c_{j^*}^k$ **then**
        $seller\_price_{i^*j^*}^k += price_{i^*j^*}^k r_{i^*j^*}^k$;
      else
        $seller\_price_{j^*}^k = price_{i^*j^*}^k r_{i^*j^*}^k$;
      end if
      Profit of SFC broker is $(b_{i^*}^k - a_{j^*}^k) * r_{i^*j^*}^k + (1 - charge) * price_{i^*j^*}^k$;
      $\mathscr{I} = \mathscr{I}/i$;
    else
      The sellers and buyers adjusting price within their limits;
    end if
  end while
end for

We propose the DARA auction algorithm to solve the ILP problem (2.17). First, we prove the following theorem that states that there is no polynomial-time dynamic algorithm for the ILP problem (2.17).

Theorem 3 *The SFC broker profit maximization problem, as shown in the ILP problem (2.17), is NP-hard.*

Proof 1 *An example of an NP-hard problems is the* $0-1$ *knapsack problem. As a typical optimization problem, it is proven to be an NP-hard problem, given by*

$$\max \sum_{i=1}^{n} w_i x_i$$

$$s.t. \sum_{i=1}^{n} w_i x_i \leq W, \quad \forall x_i \in \{0,1\}.$$

From the above equations, the $0-1$ *knapsack problem is a special form of the ILP problem (2.17) with one constraint. Thus, the SFC broker profit maximization problem is an NP-hard problem.*

From Theorem 3, we apply a game theory method to find the equilibrium solution as Algorithm 1. First, we initialize the demands of customers for different SFCs. As shown on Lines 3-10, the algorithm resets set $\mathscr{I}$ of buyers for different instances. Then, as the auctioneer in the market, the SFC broker distinguishes buyers who want to buy the $k-$th SFC as the while loop in Algorithm 1. In the distinguishing process, when the SFC broker finds that one buyer bids a price greater than the minimum asking price, the process of the auction will continue, and two participants will be chosen to stop the auction.

In the auction process, the customer needs to ask the SFC broker for the geographical information of VNFs to know the corresponding servers for the required SFCs. Only if the bidding price is higher than other bidding prices, the lowest asking price and the private value can the customer buy the required SFC from the NFV market. At the same time, the SFC broker records this purchase as Line 15 in Algorithm 1. Otherwise, the customer enters the second round or the process of price adjustment as Line 25. From Lines 23 to 24, the algorithm calculates the total profit of the SFC broker and updates the set of buyers. Next, we propose Theorem 4 to prove that the output of Algorithm 1 includes at least one feasible solution.

Theorem 4 *The output of Algorithm 1 is a feasible solution to the ILP problem (2.17).*

Proof 2 *In Algorithm 1,* r_{ij}^{k} *is a binary variable and initialized to* 0*. Thus, there is no conflict with Equation (17c). Next, we prove that the adjustment of price in Line 23 satisfies Equations (17d) and (17e). Our adjustment rules follow Equations (5.8) and (2.9), and we need to prove that there must be an intersection point between adjusting the functions of bidding and* *selling prices. From Equation (5.8), we know that*

$$\frac{\tilde{a}_j^k(n+1)}{a_j^k(n)} = 1 + \int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt, \tag{2.19}$$

where $\int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt$ *is the cumulative function of the Gaussian distribution, which is always greater than* 0. *Therefore, the price adjusting function of sellers is monotonically decreasing. Similarly, the adjusting function of buyers is monotonically increasing. Because of the assumption in Equation (2.16), we conclude that there must be an intersection point between the price adjusting functions for bidding and selling.*

Therefore, the total profit function of buyers without charge, given by

$$U_{buyer} = \sum_{j=1}^{J} \sum_{k=1}^{K} (d_i^k - p_{ij}^k) r_{ij}^k, \tag{2.20}$$

is strictly greater than 0. *Thus, there is at least one* $\varepsilon > 0$ *satisfying* $U_{buyer} \geq \varepsilon$. *If* ε *does not exist, then* $U_{buyer} < \varepsilon$ *is true for all* ε. *When we choose* ε *given by*

$$\varepsilon = \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} (d_i^k - p_{ij}^k) r_{ij}^k}{2}, \tag{2.21}$$

we can find the contradiction as

$$\sum_{j=1}^{J} \sum_{k=1}^{K} (d_i^k - p_{ij}^k) r_{ij}^k < 0. \tag{2.22}$$

Therefore, there is at least one value to satisfy Equation (17a), which is a charge for buyers in our model. For the same reason, we also find the charge for sellers from Equation (17b).

Then, we prove that our algorithm can find a feasible solution in polynomial time.

Theorem 5 *Algorithm 1 can find a feasible solution in polynomial time.*

Proof 3 *The loop from Line 1 to Line 28 in Algorithm 1 has K iterations. During the iteration, the loop from 12 to 27 has at most I rounds. Thus, the complexity of Algorithm 1 is* $O(n^2)$.

From Theorem 5, it is easy to know the time complexity of Algorithm 1 based on the double-auction is $O(n^2)$. Compared with the time complexity $O(n^2)$ of the single-auction-based algorithm [46], Algorithm 1 is a slightly slower than methods based on single auction in a general scale market. Therefore, the increased profit with Algorithm 1 is able to cover the additional computational cost.

2.4.2 Price Adjustment Algorithm for Algorithm 1

Algorithm 2 DARA price adjustment algorithm

Input: The bidding (asking) price of sellers and buyers
Output: The adjustment price of sellers and buyers
1: **for** j in range (J) **do**
2:   **if** $a_j^k \geq c_j^k$ and a_j^k is not true **then**
3:     $a_j^k = a_j^k(1 + \int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt) + \gamma$;
4:     **if** $a_j^k < c_j^k - \gamma$ **then**
5:       $a_j^k = c_j^k$;
6:     **end if**
7:   **end if**
8: **end for**
9: **for** i in range (I) **do**
10:   **if** $b_i^k \geq d_i^k$ and b_i^k is not true **then**
11:     $b_i^k = b_i^k(1 - \int_{-\infty}^{n} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2} dt) + \gamma$;
12:     **if** $b_i^k < d_i^k + \gamma$ **then**
13:       $b_i^k = d_i^k$;
14:     **end if**
15:   **end if**
16: **end for**

We propose a price adjustment algorithm to guarantee the profits of all participants. In Algorithm 2, Line 3 and Line 11 are the main adjustment functions from Equation (5.8) and Equation (2.9), considering the service charge γ. Due to the significance of the normal distribution in statistics with random variables [70], we utilize the normal distribution function to improve the flexibility of the price adjustment. If the bidding price exceeds the sum of the private valuation and the service charge, the SFC broker will terminate the transaction. Similarly, if the asking price exceeds the difference between the cost price and the service charge, the transaction will be terminated.

We use an example to illustrate Algorithm 2 for a better understanding. The bidding prices of buyers are presented in Table 2.2, where " – " denotes that the buyer has no request for the corresponding resource.

Similarly, the asking prices are shown in Table 2.3, where the container $\{1,2,0,4\}$ for resource 1 means that the buyers numbered $\{1,2,4\}$ need to buy resource 1. Because the

Table 2.2 Requests of buyers

b_i	Buyer 1	Buyer 2	Buyer 3	Buyer 4
Resource 1	13295	12874	-	13671
Resource 2	-	-	-	-
Resource 3	-	-	-	13366
Resource 4	13072	-	12617	-

Table 2.3 Resources of sellers

a_j	Supplier 1	Supplier 2
Resource 1	19106	15845
Resource 2	15756	18682
Resource 3	19668	-
Resource 4	-	-

bidding prices are less than the lowest asking price, buyers and sellers have to enter the price adjustment process. Then, the buyers adjust the original prices of resource 1, as shown in Table 2.4. Similarly, the asking price for resource 1 is shown in Table 2.5. Thus, buyer 4 and

Table 2.4 Requests for Resource 1

b_i	Buyer 1	Buyer 2	Buyer 3	Buyer 4
Resource1	20000	19000	-	21000

resource supplier 2 reach an agreement for resource 1.

Then, we prove the correctness of the price adjustment strategy.

Theorem 6 *For customers and resource suppliers, the price adjustment strategy is the weakly dominant strategy.*

Proof 4 *In this game, the SFC broker has a simultaneous price adjustment strategy that includes cooperation (C) and defection (D) for buyers and sellers. Cooperation means that the player decides to accept the given strategy, and defection means that the given strategy is rejected. If all buyers and sellers choose cooperation, they both earn profits with probabilities of* $P(C_{buyer}|C_{seller})$ *and* $P(C_{seller}|C_{buyer})$*, respectively. If buyers and sellers reject the strategy, the probabilities of the profits for buyers and sellers are given by* $P(D_{buyer}|D_{seller})$ *and* $P(D_{seller}|D_{buyer})$*, respectively. If one participant accepts the strategy while the other rejects*

Table 2.5 Resource 1 for supplier

a_j	Supplier 1	Supplier 2
Resource 1	9600	7900

the strategy, then the probability that the cooperative participant can obtain the profit is given by $P(C_{buyer}|D_{seller})$ *or* $P(C_{seller}|D_{buyer})$*. Similarly, the probability that the defective participant can obtain the profit is* $P(D_{buyer}|C_{seller})$ *or* $P(D_{seller}|C_{buyer})$*.*

According to the prisoner's dilemma game, we need to prove the following aspects:

(a) *The probability that a buyer accepts the pricing adjustment strategy is always higher than the probability of rejection, given by*

$$\begin{aligned} P(C_{buyer}|D_{seller}) &\geq P(C_{buyer}|C_{seller}) \\ &\geq P(D_{buyer}|D_{seller}) \geq P(D_{buyer}|C_{seller}). \end{aligned} \tag{2.23}$$

(b) *The probability that a seller accepts the pricing adjustment strategy is always higher than the probability of rejection, given by*

$$\begin{aligned} P(C_{seller}|D_{buyer}) &\geq P(C_{seller}|C_{buyer}) \\ &\geq P(D_{seller}|D_{buyer}) \geq P(D_{seller}|C_{buyer}). \end{aligned} \tag{2.24}$$

(c) *The probability that a cooperative buyer can obtain the profit with a cooperative seller or a cooperative seller with a cooperative buyer is higher than the probability of a cooperative buyer with a noncooperative seller or a noncooperative seller with a cooperative buyer, give by*

$$\begin{aligned} P(C_{buyer}|C_{seller}) &+ P(C_{seller}|C_{buyer}) \\ &\geq P(C_{buyer}|D_{seller}) + P(D_{seller}|C_{buyer}), \end{aligned} \tag{2.25}$$

The probability that a cooperative buyer can obtain the profit with a cooperative seller or a cooperative seller with a cooperative buyer is higher than the probability of a noncooperative buyer with a cooperative seller or a cooperative seller with a noncooperative buyer, given by

$$\begin{aligned} P(C_{buyer}|C_{seller}) &+ P(C_{seller}|C_{buyer}) \\ &\geq P(D_{buyer}|C_{seller}) + P(C_{seller}|D_{buyer}). \end{aligned} \tag{2.26}$$

Let $\mathscr{A} = \{C_{buyer}, D_{buyer}\}$ and $\mathscr{G} = \{C_{seller}, D_{seller}\}$ denote the sets of strategies for the buyer and seller, respectively. From Bayes formula [39], we know that

$$P(C_{buyer}|D_{seller}) = \frac{P(C_{buyer})P(D_{seller}|C_{buyer})}{P(D_{seller})}, \tag{2.27}$$

where

$$\begin{aligned} P(D_{seller}) =& P(C_{buyer})P(D_{seller}|C_{buyer}) \\ &+ P(D_{buyer})P(D_{seller}|D_{buyer}). \end{aligned} \tag{2.28}$$

Accordingly, we show the probability matrix in Table 2.6. For all buyers and sellers, the

Table 2.6 Probability matrix

(Buyer, Seller)	Cooperation (C)	Defection (D)
Cooperation (C)	$(P(C_{buyer}|C_{seller})$, $P(C_{seller}|C_{buyer}))$	$(P(D_{buyer}|C_{seller})$, $P(C_{seller}|D_{buyer}))$
Defection (D)	$(P(C_{buyer}|D_{seller})$, $P(D_{seller}|C_{buyer}))$	$(P(D_{buyer}|D_{seller})$, $P(D_{seller}|D_{buyer}))$

auction success is not only based on their own prices but also others' prices. Thus, every participant adjusts the price in the dominant strategy. We determine the equilibrium strategy of every participant based on Bayes game theory.

In the model, because buyers and sellers only have a common knowledge, our model is based on a symmetric independent private value (SIPV) [18] model with a typical static Bayesian game. Hence, the probability of buyer i^* with bidding price $b_{i^*}^k$ for the $k-th$ SFC is based on price b_i or a_j, $i \neq i^*$, $i = 1, ..., I; j = 1, ..., J$. Generally, we need to prove items (a) and (c).

We propose the expectation profit function of buyer i as

$$U_i = (d_i - b_i)Pr(b_i \geq b_j, j \neq i), \tag{2.29}$$

where $Pr(.)$ is the extreme probability.

Thus, we can calculate that the probability that a cooperative buyer can obtain the profit and the probability that a cooperative seller can obtain the profit as

$$P(C_{buyer}) = \prod_{i=1}^{I} Pr(b_i \geq b_{i^*}, i^* \neq i) = \prod_{i=1}^{I} \frac{U_i(d_i)}{(d_i - b_i)}, \tag{2.30}$$

$$P(C_{seller}) = \prod_{i=j}^{J} Pr(a_j \leq a_{j^*}, j^* \neq j) = \prod_{j=1}^{J} \frac{U_j(a_j)}{(c_j - a_j)}, \tag{2.31}$$

$$U_i(d_i) = [d_i - p_i]P^{I-1}(d_i), \tag{2.32}$$

$$U_j(c_j) = [p_j - a_j]P^{J-1}(a_j), \tag{2.33}$$

where $P^{I-1}(\cdot)$ is the probability distribution function.

Then, we compute the value of $P(\cdot)$ as

$$P(d_i) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(d_i - d_{i^*})^2}{2\pi\delta^2}}, \tag{2.34}$$

$$P(c_j) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(c_j - c_{j^*})^2}{2\pi\delta^2}}, \tag{2.35}$$

where δ is a constant value. Thus, we know that

$$\frac{P(C_{buyer})}{P(C_{seller})} \approx 1. \tag{2.36}$$

Accordingly, we change item (c) as follows:

$$\begin{aligned} 2P(C_{buyer}|C_{seller}) \geq & P(C_{buyer}|D_{seller}) \\ & + P(D_{seller}|C_{buyer}). \end{aligned} \tag{2.37}$$

Since the bidding prices of buyers are generally different, buyer i has a bidding function $b_i = B_i(d_i)$, where d_i is the private valuation. Thus, the strategy of buyer i is $\{B_1(d_1), B_2(d_2), ..., B_I(d_I)\}$. According to the BNE theorem, if buyer i knows that other competitors adopt strategy $B_j^*(d_j) \in \{B_1^*(d_1), B_2^*(d_2), ..., B_I^*(d_I)\}$, $j \neq i, j = 1, 2, ..., I$, buyer i will also adopt strategy $B_i^*(d_i)$. We can obtain $P(D_{buyer}|C_{seller}) = 0$ and $P(C_{buyer}|D_{seller}) = 1$. Thus, Theorem 6 is proven based on Equations (2.37) and (2.36).

2.4.3 DARA Payment Strategy Algorithm

Next, we are supposed to guarantee that our method is able to obtain the accurate price of the SFC for every buyer. This algorithm focuses on the real value of the SFC. From Equation (5.2), we just need to find a threshold value b_i^* in our scheduling.

$$p_i = \begin{cases} b^*, & b_i \geq b_i^*; \\ 0, & others. \end{cases} \tag{2.38}$$

Algorithm 3 DARA payment strategy algorithm

Input: The highest prices of sellers and buyers
Output: The truthful prices of sellers and buyers

all of i, j, k in their limits;
$p_{ij}^k = 0$;
for all $k \in [K]$ **do**
  for all $i \in [I]$ **do**
    for all $j \in [J]$ **do**
      if $r_{ij}^k == 1$ **then**
        $s\_b_{i*}^k = \min_i b_i^k$;
        $t\_b_{i*}^k = \max_i b_i^k$;
        $s\_s_{j*}^k = \min_j a_j^k$;
        $t\_s_{j*}^k = \max_j a_j^k$;
        while $(s\_b_i^k - t\_b_i^k > \varepsilon)$ **do**
          Run Algorithm 1 with $\frac{(s\_b_{i*}^k + t\_b_{i*}^k)}{2}$;
          if Buyer i wins **then**
            $t\_b_{i*}^k = \frac{(s\_b_{i*}^k + t\_b_{i*}^k)}{2}$;
          else
            $s\_b_{i*}^k = \frac{(s\_b_{i*}^k + t\_b_{i*}^k)}{2}$;
          end if
          if Seller j wins **then**
            $t\_s_{j*}^k \frac{=(s\_s_{j*}^k + t\_s_{j*}^k)}{2}$;
          else
            $s\_s_{i*}^k = \frac{(s\_s_{i*}^k + t\_s_{i*}^k)}{2}$;
          end if
        end while
        $p_{i^* j^*}^k = \frac{r_{ij}^k (t\_b_{i*}^k + s\_b_{i*}^k + t\_s_{j*}^k + s\_s_{j*}^k)}{4}$;
      end if
    end for
  end for
end for

We compute the real value based on the following Algorithm 3. Because the buyer only makes a requirement in the truthful auction mechanism, the SFC broker should find the real value of the SFC for attracting customers. First, we compute the difference between the high value and the low value in the NFV market. If the difference is greater than a constant value ε, we need to halve the price and then run Algorithm 1 until we find the real value of the SFC.

2.5 Performance Evaluation

In this section, we study the NFV service chain in the double-auction market where customers can share SFCs. Each SFC contains three types of resources: storage, routing and firewall. We generate the matrix of bidding prices and asking prices following a random distribution.

First, we compare the resource allocation performance between the DARA model and the single-auction model with different numbers of buyers. Figure 2.2 shows the profit of the SFC broker when the number of buyers is changed with different auction models. It shows the profit of the SFC broker with 100 service suppliers and 100 types of SFCs. From the results, the profit of the SFC broker is increased with more buyers. This result occurs because the profit of the SFC broker mainly comes from the buyers. The solid blue line shows the optimal solution of the profit of the SFC broker maximization problem, and the solid red line is the profit of the SFC broker calculated by the single-auction model. As shown, the profit of the SFC broker in the DARA model is always higher than the profit in the single-auction model.

In addition to the results shown in Figure 2.2, we also compare the resource allocation performance between the DARA model and the single-auction model with different numbers of sellers in Figure 2.3. In contrast to the previous results, the profit of the SFC broker changes erratically when we increase the number of sellers. This result occurs because the sellers are not the main factor of the SFC broker profit. It also shows that the profit of the SFC broker in the DARA model is always higher than the profit in the single-auction model.

As shown in Figure 2.4, the profit of the SFC broker is increased by more VNFs. This figure shows the profit of the SFC broker in the DARA model is always higher than the profit in the single-auction model.

Figure 2.5, Figure 2.6 and Figure 2.7 show the proportionality factors between the DARA model and the single-auction model. In Figure 2.5 and Figure 2.7, the line is irregular due to the random training data, while the value of proportionality is always greater than 1. From Figure 2.6, compared to the results in the other figures, the value of proportionality is decreased with more sellers. This result occurs because the number of sellers cannot impact the profit of the SFC broker.

We finally compare the profits of sellers and buyers between the DARA model and the single-auction model. The empirical cumulative distribution function (CDF) of the profit of sellers and buyers is shown in figures. From the results shown in Figure 2.8, the proposed scheme outperforms the single-auction model. Furthermore, we observe that the probability of the profits for sellers who earn nothing in the DARA model is approximately equal to 0.1. Moreover, the proportion of the seller profit increases gradually, and approximately 90% of sellers obtain 2.5×10^6 profit. Thus, every seller can obtain some benefits in the DARA

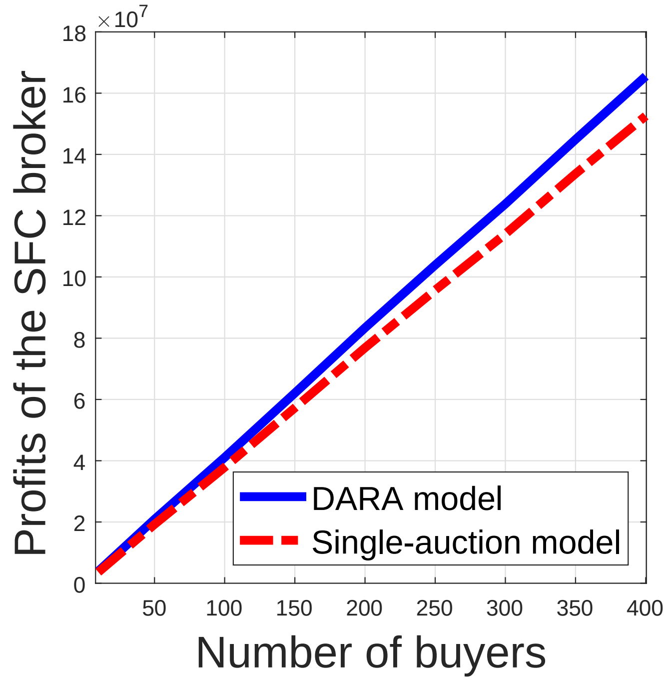


Fig. 2.2 Profits of the SFC broker with impact of buyers

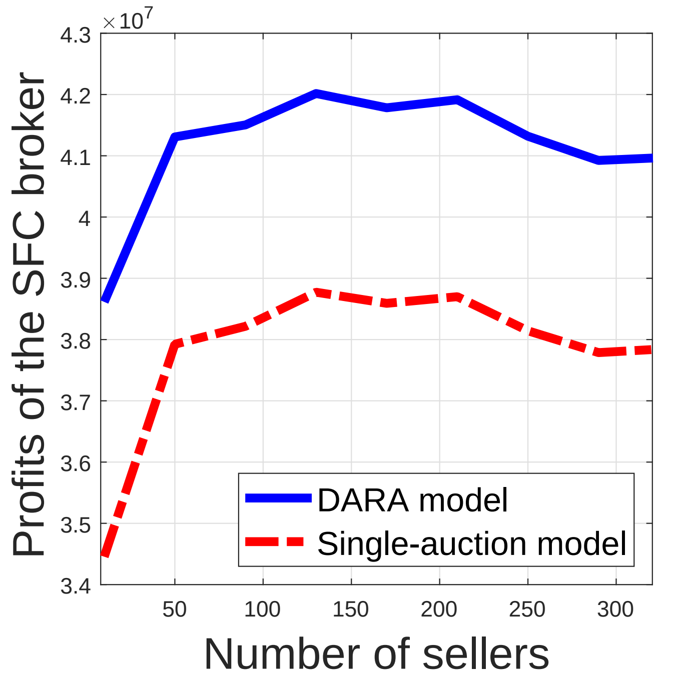


Fig. 2.3 Profits of the SFC broker with impact of sellers

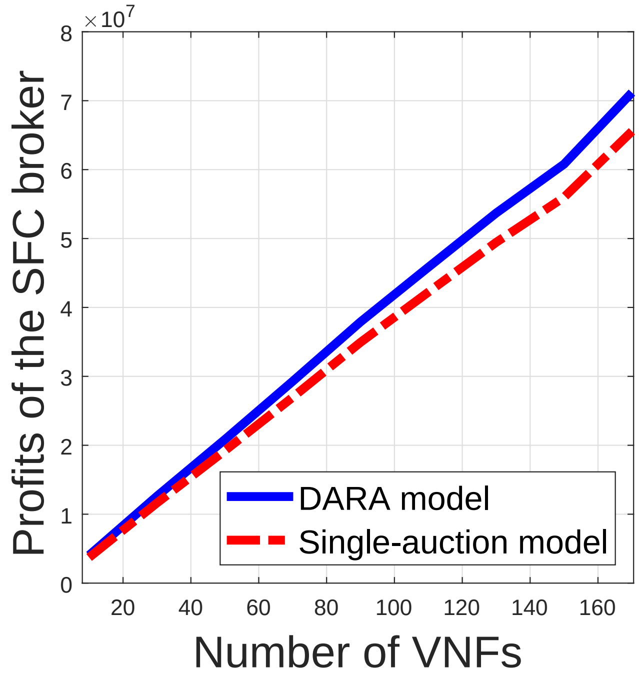


Fig. 2.4 Profits of the SFC broker with impact of VNFs

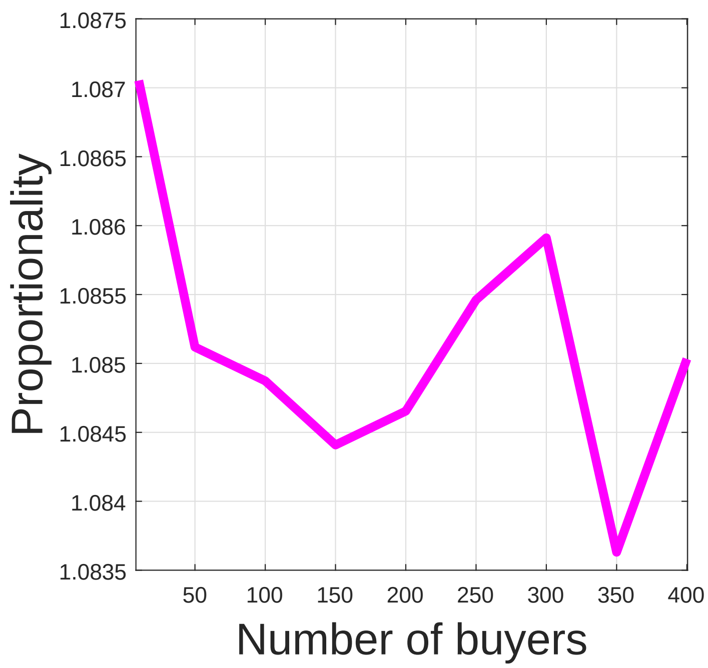


Fig. 2.5 The VNF profit proportionality affected by buyers compared between DARA model and single-auction model

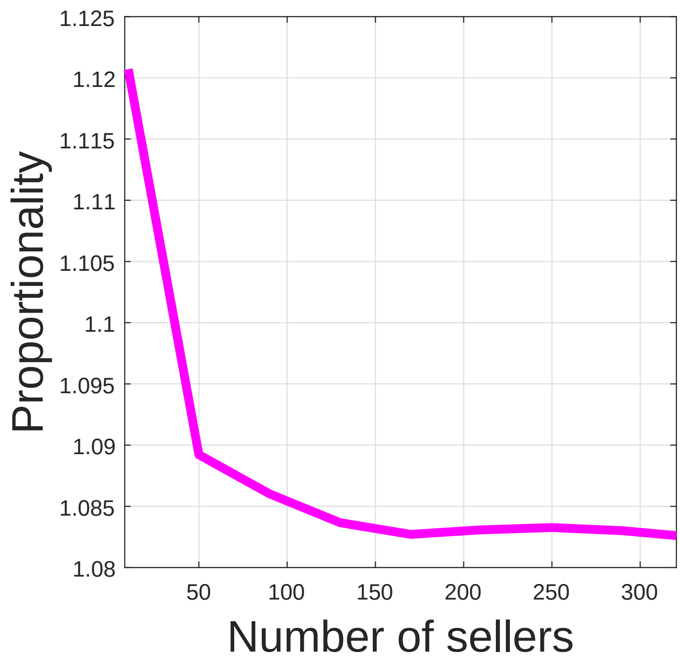


Fig. 2.6 The VNF profit proportionality affected by sellers compared between DARA model and single-auction model

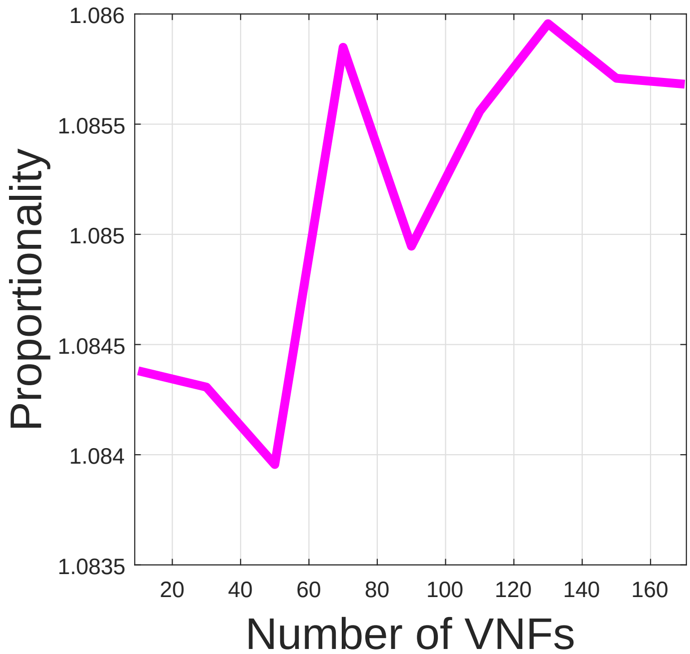


Fig. 2.7 The VNF profit proportionality affected by VNFs compared between DARA model and single-auction model

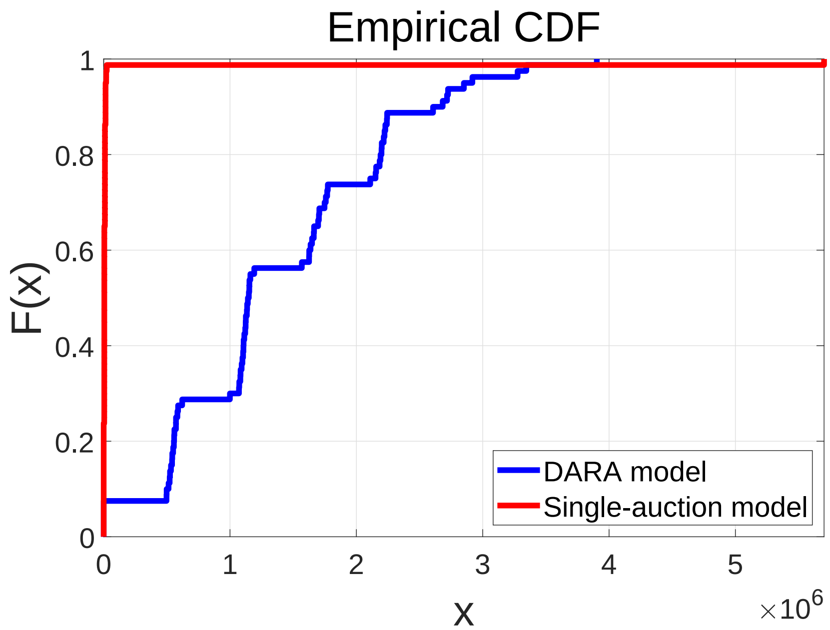


Fig. 2.8 Empirical CDF of the seller profit

model. However, in the single-auction model, the proportion of low-income sellers accounts for up to 90% of the population. Because we assume that one seller can provide resources to a different customer, the blue line is like a "stair-step" graph. Similar to the seller profit analysis, in Figure 2.9, we also present the CDF of buyer profit to show that the profit of more than 80% buyers is less than 1×10^6 in the single-auction model, while more than 95% buyers can achieve more than 1×10^6 profit in the DARA model.

In general, we find that the DARA model performs better than the single-auction model due to the limitation of the single-auction model in which only customers can change the price. Our results show that we can achieve the main goal, which is to maximize the profits of the three participants.

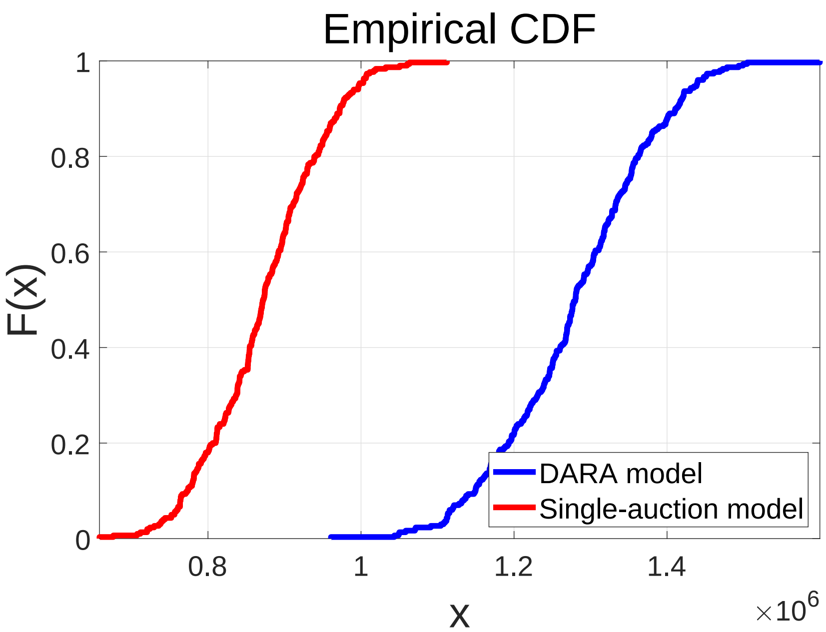


Fig. 2.9 Empirical CDF of the buyer profit

Chapter 3

Multiple-Walrasian Auction Mechanism for Tree Valuation Service in NFV Market

This chapter introduces the Multiple-Walrasian Auction Graphic Model combined with Virtual Network Functions (VNFs) of different bundled tree nodes based on Vickrey-Clarke-Grove (VCG) payment to maximise the social effectiveness of NFV. It is the first work to define the virtualized service as a tree valuation in NFV market. To solve this problem, novel algorithms including the valuation structure algorithm and auction strategy are implemented along with the NFV to schedule network resources. With all theory taken into account, we conducted a comprehensive simulation to verify the tree valuation mechanism. The results confirmed that the tree valuation mechanism outperforms Backpack auction model and Reserve auction moedel in respect to social welfare.

3.1 Motivation

We present a service allocation mechanism through the double-auction method combined with tree valuation for improving the efficiency of utilizing and the public service performance in the NFV market. In a traditional auction as single English auction, the set of results is finite, and each element corresponds to a winner who may win the item. In a bundled auction, there are multiple items for sale, and each item can not be divided. Bidders' preferences for different sets of items may be complex. There are two different types of items in the bundled auction. One is that items are substitutes for each other and the other is that items are complementary for each other[55]. It is very complicated to design an effective bundled

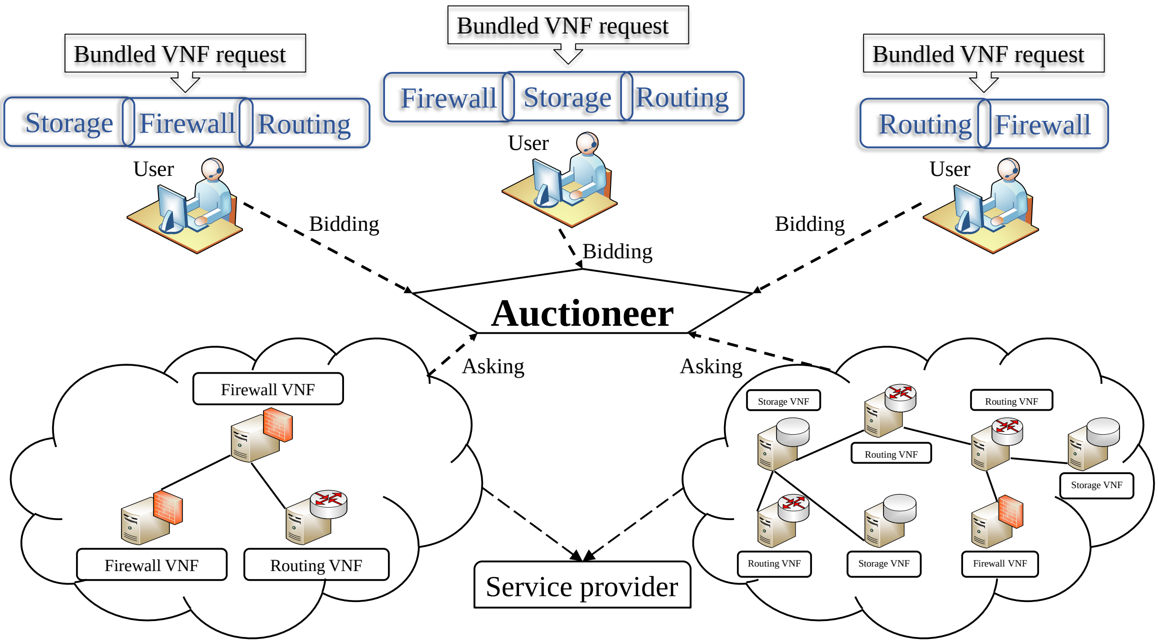


Fig. 3.1 Double auction with tree valuation in the NFV market

auction because the allocation rules and payment rules need to be carefully combined. We can maximize the social welfare through the VCG payment mechanism which is based on the bidding rule and the winner will pay the value of loss of losers caused by the winner. When winners achieve the item, they must pay the social welfare for losers because of its existence.

Instead of using a single English auction, we propose a bundled service as a tree valuation auction method to guarantee the profit of both parties by bidding and asking. An iterative auction based Multiple-Walrasian auction can guarantee the demand equals to the supply. Consequently, an efficient tree valuation model based on Multipl-Walrasian auction through VCG payment rule is designed to provide in which service provider and users submit the asking and bidding price. The objectives of our model are to maximize the social welfare and to guarantee the demand is equal to the supply. For example, as shown in Figure 3.1, there are three types of network service resources including firewall, storage, and routing.

The numerical results are in agreement with the analysis ones. The rest of this chapter is structured as follows. In Section 3.2, we discussed the previous works about the VNF resource allocation, the double-auction, the tree valuation and VCG payment. Section 3.3 consists of two parts. The related preliminaries and some assumptions are introduced in Subsection 3.3.1. Then, we propose the new resource scheduling method based on the tree valuation Multipl-Walrasian equilibrium in Section 3.3.2. Section 3.4 present novel algorithms based on the Multipl-Walrasian double-auction including tree valuation structure and auction process. Next, we prove the availability and efficiency of tree valuation auction by comparison with Backpack auction and Reserve auction in Section 3.5.

3.2 Related Work

We introduce relevant works from three aspects: VNF resource allocation, tree valuation, and double-auction.

3.2.1 VNF resource allocation

The topic of VNF resource allocation is a challenge in research about NFV market. To address the challenge, some researchers including VNF working group of IETF propose a dynamic way to schedule VNFs as a sequence of functions [11, 53]. Prior works [43] are mainly used by placing VNF resources to solve the resource allocation problem. However, social welfare and resource market are two important and meaningful method to improve the service performance combined with competing customers. Lin et al. [68] propose a new scheme in which they schedule resources as a multi-tree to minimize the response time of user obtaining service.

3.2.2 Double-auction

To consider the balance between the demand and the supply, there are some researchers focus on applying auction-based pricing method. Zaman et al. [109] propose a truthful auction-based mechanism to dynamically schedule VMs with consideration of the user demand. Wei et al. [89] propose a truthful online combinational auction mechanism which can optimize system efficiency with continuous time. Zhang et al. [113] employ an α-approximation algorithm to design a random combinatorially effective auction to guarantee the social welfare. In [112], authors design a truthful online auction market with a novel bidding language to reflect the relation of supply and demand on time. Li et al. [61] propose an auction which is truthful to maximize the revenue of the provider by pricing the resources and the payment rule. Wei et al. [98] apply Stackelberg game and Markov model to design a cloud allocation model to maximize the revenue of the provider and the users. Li et al. [62] conduct the pricing strategy of multi-users competition for resource allocation in which each user share the information. Borjigin et al. propose a new double-auction mechanism for network service chain and price strategy to maximize the total profit [10]. Peng et al. propose an double-auction mechanisim based on milti-atribute in fog computing cloud [80]. Li et al. also propose the double auction mechanism including a truthful auction and a efficient auction, in which multi-sellers services to multi-buyers to establish a two-sided relationship, to improve the efficiency of the trade and social welfare [66].

3.2.3 Tree valuation

Some researchers study the iterative auction problem as valuations of the subgraph and focus on common pricing rules. Candogan et al. analyze how to graph valuations solve iterative auctions with simple rules. In this work, they study the maximum welfare of bidders problem with hypergraph valuations [16, 17, 15]. The valuations of bidders describe as nodes value of subgraphs, then the authors propose a truthful and approximate algorithm. Diestel et al. [27] propose systematically the introduction of emerging graph valuation and combine with an analysis of mathematical proof in detail. In our chapter, we based on the content of the above standard and combined the tree valuation to schedule our resource.

The definition of the VCG payment mechanism is from Clarke who generalized the single item auction proposed by Vickrey. Groves proposes a more general mechanism in which each bidder's payment rule adds an independent bidding key item [81]. Wang et al. propose a truthful VCG payment auction for maximizing the profit in Multi-Area power system[97]. VCG payment is also important in the green Data Center. Zhou et al. achieve a social welfare maximization and truthfulness auction mechanism based on VCG payment[117]. Ma et al. show a VCG payment auction guarantee the incentive compatibility and the budget balance[71]. Ma et al. propose a truthful double auction mechanism with the pricing strategy using VCG payment to enhance the property in Mobile Edge Computing environment [72].

We design a real-time bidding strategy through the user's bidding behavior to predict the user's bidding strategy for any specific auction. This chapter focuses on NFV market and dynamic strategies designed for tree valuation through VCG payment auction to maximize the social welfare.

3.3 Model and Preliminaries

In this section, we introduce our basic definitions and assumptions in Subsection 3.3.1 and present our model in subsection 3.3.2. We first design a resource scheduling model to maximize the social welfare based on Multipl-Walrasian auction market. In Multipl-Walrasian auction market, each user and NFV provider submit bidding and asking price in private, and calculate their demand and supply for the bundled service resource in the first round. Then all parties can know all prices from other parties, and resubmit the new price to the auctioneer and rotate around. When the demand and supply are balanced, the auctioneer gives the market-clearing price for VNFs. We present the bundled tree valuation VNFs scheduling by Multipl-Walrasian auction in the NFV market.

3.3.1 Definition and Assumption

We consider settings where three parties engaged in the market are the auctioneer, users and NFV providers in NFV market. Users give the bidding price and the demand to buy a sequence of VNFs and NFV providers give the asking price and the supply to selling all of VNFs. In order to guarantee users and NFV providers giving the truthful price in the market, we assume that the market is in accordance. Then we introduce the definition of a Multipl-Walrasian auction mechanism and propose the first assumption.

Definition 4 (Multiple-Walrasian double-auction mechanism) *In Multiple-Walrasian double-auction mechanism, the value of demand equals to the supply after users submit the demand function based on every possible bidding price and providers submit the supply, given by*

$$\sum_{n=1}^{N}\sum_{i=1}^{M} p_{ni}D_{ni} = \sum_{n=1}^{N}\sum_{j=1}^{L} p_{nj}S_{nj}, \tag{3.1}$$

where p_n is the market-clearing price of n-th item. D_{ni} and S_{nj} denote the demand and supply of the n-th item for the i-th user and j-th provider, respectively.

From the assumption, we know there is a partial equilibrium between users and providers in Walrasian auction mechanism. We assume that there are L number of providers selling N number of item to M number of users. Let $\mathscr{L}$, $\mathscr{N}$ and $\mathscr{M}$ are the set of providers, items and users, respectively. Each user $i \in \mathscr{M}$ has a value function $v_i : 2^N \to \mathbb{R}^+$ to calculate the profit of users. Let D_i denotes the set of demand from user i, and $v_i(D_i)$ is the value function of user i. We assume that the value functions of each user are private, i.e., each user can not know the value function of others. On the other hand, each provider has own value function r_j for provider j, given by

$$r_j(n) = p_{nj} - c_{nj}, \forall j \in \mathscr{L}, \tag{3.2}$$

where $p_j(n)$ and $c_j(n)$ is the market-clearing price and cost of item n, respectively.

According to the above definition 4, we assume that the value functions for users are monotone and may not be a linear function. We propose our first assumption as follows:

Assumption 1 *We assume the following three assumptions in our model:*

1. *If the set of item is empty, the value function is equal to zero, i.e., $v(\emptyset) = 0$.*
2. *The value function v is monotone, i.e., $v(\{n_1,n_2\}) \le v(\{n_1,n_2,n_3\}), \forall n_1,n_2,n_3 \in D$, where D is the demand set of user.*

3. *The value function is not always linear, i.e.,* $v(\{i,j\}) \neq v(n_1)+v(n_2)$*, for all* $n_1, n_2 \in D$*.*

When the value function of user i meet $v_i(\{n_1,n_2\}) \geq v_i(n_1)+v_i(n_2)$, we call item n_1 and item n_2 can be bought together, and the value of bundled item $v_i(\{n_1,n_2\})$ for user i is greater than the sum of value function for each item $v_i(n_1)+v_i(n_2)$ for user i. When the value function of user i meet $v_i(\{n_1,n_2\}) \leq v_i(n_1)+v_i(n_2)$, we call item n_1 and item n_2 can not be bought together, the value of buying separated is greater than the value of buying bundled.

Definition 5 (Graphical valuation) *We denote a graph by* $G = (\mathscr{N}, \mathscr{E})$*, where* $\mathscr{N}$ *and* $\mathscr{E}$ *are the set of nodes and edges, respectively. Let* w_{n_1} *and* $w_{n_1 n_2}$ *denote the weight of node* n_1 *and the weight of edge between node* n_1 *and* n_2*, respectively. If the value function* v *meet* $v(\{n_1,n_2\}) = w_{n_1}+w_{n_2}+w_{n_1 n_2}, \forall n_1, n_2 \in \mathscr{N}$*, we call the value function is graphical with graph* G*.*

From the definition, we can see if weight $w_{n_1 n_2} \geq 0$, item n_1 and item n_2 can be sold together and the value of bundled items is greater than the sum of value per unit, i.e.,

$$\sum_{n \in B} v(\{n\}) = \sum_{n \in B} w_n \leq \sum_{n \in B} w_n + \sum_{\{n_1,n_2\} \in B} w_{n_1 n_2} = v(B). \tag{3.3}$$

If weight $w_{n_1 n_2} \leq 0$, it is better to sell separately than sell bundled of item n_1 and item n_2, i.e.,

$$\sum_{n \in B} v(\{n\}) = \sum_{n \in B} w_n \geq \sum_{n \in B} w_n + \sum_{\{n_1,n_2\} \in B} w_{n_1 n_2} = v(B). \tag{3.4}$$

In our model, we apply graphical theory with $G = (\mathscr{N}, \mathscr{E})$ to all value function. There are node weights and edge weights which can represent the value of bundle B. The sum of the weights for both nodes and edges is equal to the value of bundle $B \in \mathscr{G}$ as shown in Figure 3.2. We propose our second assumption as follows,

Assumption 2 *We assume that*

- *For each user i and provider j, there exist a graph* $G = (\mathscr{N}, \mathscr{E})$*, in which* $\mathscr{N}$ *is the set of items and* $\mathscr{E}$ *is the set of favor weight for the bundled items.*
- *If no any edge between* n_1 *and* n_2*, we denote weight* $w_{n_1 n_2} = 0$*. In other words, item* n_1 *and* n_2 *can not be sell bundled. i.e.,* $\sum_{(n_1,n_2) \in \mathscr{E}} w_{n_1 n_2} = \sum_{(n_1,n_2) \in \mathscr{E}} w_{(n_1 n_2)}$.

From Assumption 2, we can denote both bundled set by graph $G = (\mathscr{N}, \mathscr{E})$. In Multiple-Walrasian double-auction mechanism, the auctioneer price items based on the graphical

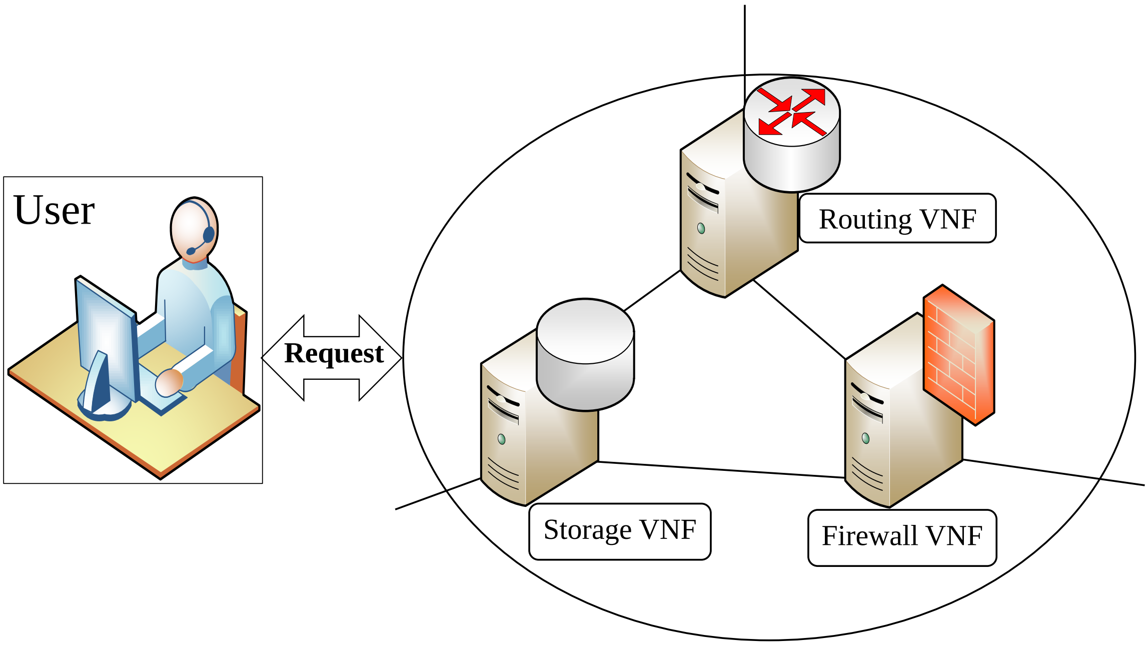


Fig. 3.2 User's request with tree valuation in NFV market

value, in which $w_{n_1n_2}$ denote the favor value between item n_1 and item n_2. When $w_{n_1n_2} = 0$, value function $v(\{n_1,n_2\}) = v(n_1) + v(n_2)$.

Then we give a definition about an efficient allocation. From Definition 4, we know the demands are equal to the supplies. We denote that an efficient allocation is to optimize the social welfare. In our model, the total social welfare is as follows,

$$\max \sum_{i=1}^{M} \sum_{j=1}^{L} \sum_{\{n_1,n_2\}\in\mathcal{N}} (v_i(\{n_1,n_2\}) + r_j(\{n_1,n_2\})). \tag{3.5}$$

Combined with the definition of graphical, we denote the definition of competitive equilibrium in Multiple-Walrasian double-auction as follows,

Definition 6 (Competitive equilibrium in Multiple-Walrasian double-auction) *We call a sequence* $[\{p_n^*\}_n, \{D_{ni}^*\}_i, \{S_{nj}^*\}_j], \forall n \in \mathcal{N}, i \in \mathcal{M}, j \in \mathcal{L}$, *is a Multiple-Walrasian equilibrium if*

1. *Both clearing price* $p_n^*, \forall n \in \mathcal{N}$ *is greater than zero, i.e.,* $p_n^* > 0$.
2. *Total demands are equal to total supplies,no item will be destroyed or created, i.e.,* $\sum_{n=1}^{N} p_n^*(\sum_{i=1}^{M} D_{ni}^* - \sum_{j=1}^{L} S*_{nj}) = 0$.
3. *Both users and providers have profit, i.e.,* $v(D_{ni}^*), r^*(S_{nj}) \geq 0$.

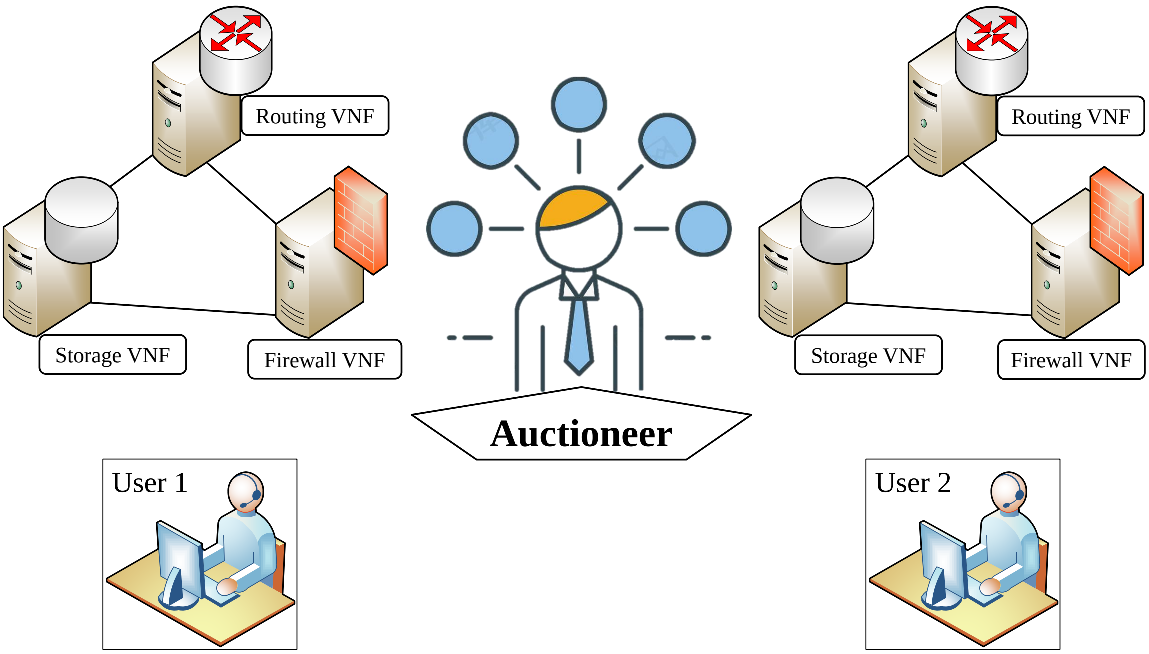


Fig. 3.3 Consistent tree valuation

As shown in Figure 3.3, for user 1 and user 2, the provider sells the same type of bundled items with the same price. In this chapter, we focus on tree valuation based on the graphical valuation. Hence, we propose the following assumption.

Assumption 3 (Consistent tree valuation) *We assume that our graph model is a Consistent tree if it meets the following conditions:*

1. *Graph* $G = (\mathscr{N}, \mathscr{E})$ *is a tree, i.e.,* $N-1$ *is equal to the number of edge* $|\mathscr{E}|$.
2. *Tree valuation graph meet that if* $w^i_{n_1n_2} > 0$, *for other users* $w^{i'}_{n_1n_2} > 0$ *for all* $n_1, n_2 \in \mathscr{N}$ *and* $i' \in \mathscr{M}/\{i\}$. *Similarly, if* $w^i_{n_1n_2} < 0$, *we can achieve inequality* $w^{i'}_{n_1n_2} < 0$ *for all* $n_1, n_2 \in \mathscr{N}$ *and* $i' \in \mathscr{M}/\{i\}$.

From Assumption 3, we can avoid that providers sell bundled items in circle. For example, if items n_1, n_2 and n_3 can be sold as $v(\{n_1, n_2\}) = 1$, $v(\{n_1, n_3\}) = 2$ and $v(\{n_2, n_3\}) = 3$, we will delete edge (n_1, n_2) to guarantee the profit of users. Note that Assumption 3 also guarantee the justice among users and providers. For each user, providers only can sell bundled items S with same price.

Definition 7 (Dominant-Strategy Incentive Compatible (DSIC)) *In an auction mechanism, if the bidding price of every bidder is based on its true valuation, the strategy is a dominant strategy and the utility of the bidder is not negative, we call the auction is Dominant-Strategy Incentive Compatible[47].*

Definition 8 (Vickrey-Clarke-Grove (VCG) payment) *The paying price of the winner i is equal to the loss of the loser $-i$ caused by the winner i.*

In this section, we propose three assumptions and five definitions. In 3.3.2, we will establish a novel model in the NFV market based on the consistent tree valuation graph.

3.3.2 Tree Valuation Model

In this part, we will model our auction mechanism to maximize the social welfare in the NFV market, given by in Equation 3.5. In the NFV market, each provider sells VNFs resource as either bundled or separated to users. When providers submit their selling price and supplies which is constant value to the auctioneer, users also submit their bidding price and demands which is based on the bidding price. When the total demands are equal to the total supplies, the auctioneer gives the clearing price for both bundled and separated items. Let we denote the decision variable by $x^{ij}_{n_1n_2} \in \{0,1\}$. If value $x^{ij}_{n_1n_2} = 1$, it means service provider j sell bundled VNFs n_1 and n_2 to user i. If value $x^{ij}_{n_1n_2} = 0$, service provider do not sell bundled item n_1 and n_2. Let $y^{ij}_n \in \{0,1\}$ denote the decision of separated VNFs. If $y^{ij}_n = 1$, service provider j sell their VNF n to user i. If $y^{ij}_n = 0$, service provider can not sell single item n to user i.

Then our objective function can be written as follows:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{L}\sum_{\{n_1,n_2\}\in\mathcal{N}} (w^i_n y^{ij}_n + w^i_{n_1n_2} x^{ij}_{n_1n_2} + r_j(\{n_1,n_2\})). \tag{3.6}$$

Equation 3.6 can change into as follows:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{L}\sum_{\{n_1,n_2\}\in\mathcal{N}} (w^i_n y^{ij}_n + w^i_{n_1n_2} x^{ij}_{n_1n_2} + p_{nj} y^{ij}_n - c_{nj} y^{ij}_n). \tag{3.7}$$

From the additional conditions, our binary optimization problem can be written as:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{L}\sum_{\{n_1,n_2\}\in\mathcal{N}} ((w^i_n + p_{nj} - c_{nj}) y^{ij}_n + w^i_{n_1n_2} x^{ij}_{n_1n_2}). \tag{8}$$

$s.t.$

$$x^{ij}_{n_1n_2}, y^{ij}_n \in \{0,1\}, \tag{3.8a}$$

$$y^{ij}_{n_1} + y^{ij}_{n_2} - 1 \le x^{ij}_{n_1n_2} \le y^{ij}_{n_1}, y^{ij}_{n_2}, \tag{3.8b}$$

$$\sum_{n=1}^{N} p_n \Big(\sum_{i=1}^{M} D_{ni}(\{y^{ij}_n\}_i) - \sum_{j=1}^{L} S_{jn}\Big) = 0. \tag{3.8c}$$

Table 3.1 Summary of notations

Symbols	Description
L	Number of providers.
M	Number of users.
N	Number of VNFs.
$\mathscr{L}$	Set of providers.
$\mathscr{M}$	Set of users.
$\mathscr{N}$	Set of VNFs.
$\mathscr{E}$	Set of edges between two nodes.
$v(n)$	Value function of item n for users.
$r_j(n)$	Revenue of provider j for item n.
w_n	Weight of node n.
$w_{n_1n_2}$	Weight of edge between node n_1 and n_2.
D_{ni}	Demand of the $n-$th item for user i.
c_{nj}	Cost of item n for provider j.
p_n	Price of item n.
$w^i_{n_1n_2}$	Weight of edge between node n_1 and n_2 for user i.
S_{nj}	Supply of provider j for item n.
$x^{ij}_{n_1n_2}$	Decision of the bundled VNFs $\{n_1,n_2\}$ between user i and provider j.
y^{ij}_n	Decision of the separated VNF n between user i and provider j.
$b_i(w)$	Bidder price of user i for item w.
ω	Set of distribution result.

We can know variable $x^{ij}_{n_1n_2}$ as follows:

$$x^{ij}_{n_1n_2} = y^{ij}_{n_1} y^{ij}_{n_2}. \tag{3.9}$$

Equation 3.9 means $x^{ij}_{n_1n_2} = 1$ if and only if user i buy items n_1 and n_2 from service provider j with bundled or separated type. Hence, if user do not buy item n_1 or n_2 from provider j, variable $x^{ij}_{n_1n_2}$ is equal to 0.

Because problem 3.8 is a binary problem, we relax the problem to a linear problem, given by

$$\max \sum_{i=1}^{M} \sum_{j=1}^{L} \sum_{\{n_1,n_2\}\in\mathcal{N}} ((w_n^i + p_{nj} - c_{nj}) y_n^{ij} + w_{n_1 n_2}^i y_{n_1}^{ij} y_{n_2}^{ij}). \tag{10}$$

$s.t.$

$$y_n^{ij} \in (0,1), \tag{3.10a}$$

$$y_{n_1}^{ij} + y_{n_2}^{ij} - 1 \leq y_{n_1}^{ij} y_{n_2}^{ij}, \tag{3.10b}$$

$$\sum_{n=1}^{N} p_n (\sum_{i=1}^{M} D_{ni}(\{y_n^{ij}\}_i) - \sum_{j=1}^{L} S_{nj}) = 0. \tag{3.10c}$$

3.4 Algorithm

In this section, we design three algorithms which consists of the tree valuation structure algorithm, the auction algorithm and the payment algorithm to maximize the social welfare in the NFV market.

Algorithm 4 Tree valuation structure algorithm

Input: The numbers of users, providers, and VNFs; the demand of users; the supply of providers;

Output: The graph of bundled VNFs;

1: **while** Price p is available **do**
2: Initialize the price of users and service providers;
3: Collect the demand function and the supply function for users and service providers;
4: State the pricing rate R_p and step-size α. Update the price with $p + \alpha R_p$. Go to step 1;
5: **end while**

In our model, users require bundled VNFs as a tree valuation, while service providers sell the corresponding VNFs. This problem is considered as a tree valuation Multiple-Walrasian double-auction problem. In Algorithm 1, we introduce the details of the tree valuation structure of bundled VNFs. Algorithm 2 is the process of the double-auction to output a feasible solution. Algorithm 3 is a payment mechanism based on VCG mechanism which is DSIC. In algorithm 1, we can achieve bundled VNFs as tree valuations. When the price of VNF is available, the VNF can be bundled as a tree. When the price of VNF is not available, we define the VNF as the substitute. The loop from line 1 to line 5 has N iterationm, the complexity of Algorithm 1 is $O(n)$. In algorithm 2, we can achieve the decision of bidders and the provider in a double auction mechanism. The loop from line 1 to line 16 has L

iteration, in which line 3 and line 4 has at most M iteration. Hence, the complexity of Algorithm is $O(n^2)$.

The key point of Algorithm 6 is to sell the items one by one as the price goes up. In addition to the current price, the auction keeps track of the current supply S and the surplus budget B_i^* of bidder i. There are two cases in the inner loop. In the first case, When the total surplus demand exceeds surplus supply, however, the total surplus demand of other bidders except bidder i is less than the total surplus supply of other bidders expect bidder i, bidder i will get the item and upload the new budget of bidder i. The total surplus demand and supply will minus 1. The second case happens only when the total demand $\sum_{j=1}^{n} D_j^+(p)$ is reduced by two or more at a price p. Assuming that all $\frac{B_i^*}{k}$ are marked with different values, this will only happen when price p is equal to valuation v_l of bidder l. In this case, when the demand of bidder l drops to 0, there is no competition among the remaining s items, so the remaining demand of all bidders is met at the same time. After satisfying the bidder's surplus demand, there may be surplus goods. At this time, we assign them to bidder i in price $p = v_l$ who have no discrimination. The loop from line 2 to line 19 has N iteration, during the iteration the loop from line 5 to line 18 has at most M iterations. Hence, the complexity of Algorithm is $O(n^2)$.

Algorithm 5 Multiple-Walrasian double-auction algorithm

Input: The numbers of users and service providers; the demand of users; Graph of VNFs for providers;

Output: The set of distribution results;

```
1: while the user is available do
2:   if the maximum bidding price is greater than the minimum asking price then
3:     Find the user whose price is the maximum value;
4:     Find the provider whose price is the minimum value;
5:     The hander price is in limit;
6:     Upload the decision of users;
7:     Upload the price of users;
8:     if the price of the provider is in limit then
9:       Upload the price of the provider;
10:    else
11:      Keep the price of the provider;
12:    end if
13:    Remove user i;
14:  end if
15: end while
```

Theorem 7 *In any multi-parameter auction, there is a social welfare maximization mechanism.*

Proof 5 *First, every bidder submits the true valuation and chooses the distribution result. Due to the social welfare maximization, the bidder price $b_1, b_2, ..., b_n$is equal to the private valuation.We define distribution rules $x(b)$ as follows:*

$$x(b) = \arg\max_{w \in \Omega} \sum_{i=1}^{n} b_i(w), \tag{3.11}$$

Ω is the distribution result.

Second, under the above distribution rule, we define a payment rule as follows:

$$p_i(b) = \max_{w \in \Omega} \sum_{j \neq i} b_j(w) - \sum_{j \neq i} b_j(w^*), \tag{3.12}$$

$w^* = x(b)$ is from Equation (3.11).

Third, we select a generalized mechanism in design environment, and (x, p) meets Equation (3.11) and (3.12). We can change Equation (3.12) into,

$$p_i(b) = b_i(w^*) - [\sum_{j=1}^{n} b_j(w^*) - \max_{w \in \Omega} \sum_{j \neq i} b_j(w)]. \tag{3.13}$$

The above equation means the payment of bidder i is equal to the bidding price deducts the partial refund. The partial refund is from the welfare increment of bidder i. For example, there are two bidders in second-price auction, if the winner is bidder 1, the payment p_1 of the bidder 1 is equal to the bidding price b_1 deducts the partial refund $b_1 - b_2$ (b_2 is the second bidding price) . The partial refund also means the welfare increment bidder 1 brings to the system. When the bidder price is not negative, the partial refund in Equation (3.13) is also not negative,

$$p_i(b) \geq b_i(w^*). \tag{3.14}$$

Therefore,truthful auction can ensure non negative utility and maximization social welfare. Acording to the defination 7, we only need to proof that bidder i can maximize the quasilinear utility $v_i(x(b)) - p_i(b)$ by seting up $b_i = v_i$ for any bidder i and any other bidder price b_{-i}.

When we seleced i and b_{-i}, choose $x(b)$ as w^*, the utility function can be written as follows:

$$v_i(w^*) - p_i(b) = [v_i(w^*) + \sum_{j \neq i} b_j(w^*)] - [\max_{w \in \Omega} \sum_{i \neq j} b_j(w)]. \tag{3.15}$$

In the above funtion, $\max_{w \in \Omega} \sum_{i \neq j} b_j(w)$ of independent b_i is a constant. Therefore, utility function maximization probelm can change into $v_i(w^*) + \sum_{j \neq i} b_j(w^*)$ maximization problem. If bidder i can select the outcome w^* directly without bidding price b_i, bidder i is bound to *choose the maximize of* $v_i(w^*) + \sum_{j \neq i} b_j(w^*)$. *If bidder i set up* $b_i = v_i$, *the maximization Equation (3.11) is same as* $v_i(w^*) + \sum_{j \neq i} b_j(w^*)$ *maximization problem. Therefore, the truthful bidding can maximize the utility function of bidder i.*

Algorithm 6 VCG payment algorithm

Input: The set of distribution results and the surplus budget;
Output: The set of payment price p;

$p = 0$, $S = 0$, $b_i = B_i$;
while $S > 0$ **do**
  Increase p to v_i;
  Find out i of the maximum surplus demand $D_i^*(p)$;
  while $\sum_{i \neq j} D_i^*(p) < S$ **do**
    if $\sum_{j=1}^{n} D_j^*(p) > S$ **then**
      $p_i = p$;
      $\{b_i\} = \{b_i\}/b_i$;
      $S = S - 1$;
      Based on price p, find out i of the maximum surplus demand $D_i^*(p)$;
    else
      $\sum_{j=1}^{n} D_j^*(p) \leq S$;
      $p_j = p$;
      ASSIGN
      Assign surplus items to bidder l who meet the condition $v_l = p$;
      $S = 0$;
    end if
  end while
end while

Theorem 8 *Algorithm 6 is the DSIC if bidders have a common budget constraint.*

Proof 6 *When the bidding price* b_i *and the other bidding price* b_{-i} *are fixed value, we define the surplus demand* $D_i^*(p)$ *of bidder i as follows:*

$$D_i^*(p) = \begin{cases} \min\{\lfloor \frac{B_i^*}{p} \rfloor, S\}, & p < v_i; \\ 0, & p > v_i. \end{cases} \tag{3.16}$$

We define the extremum $D_i^+(p)$ *of the surplus demand* $D_i^*(p)$ *as follows:*

$$D_i^+(p) = \lim_{q \to p} D_i^*(q).$$

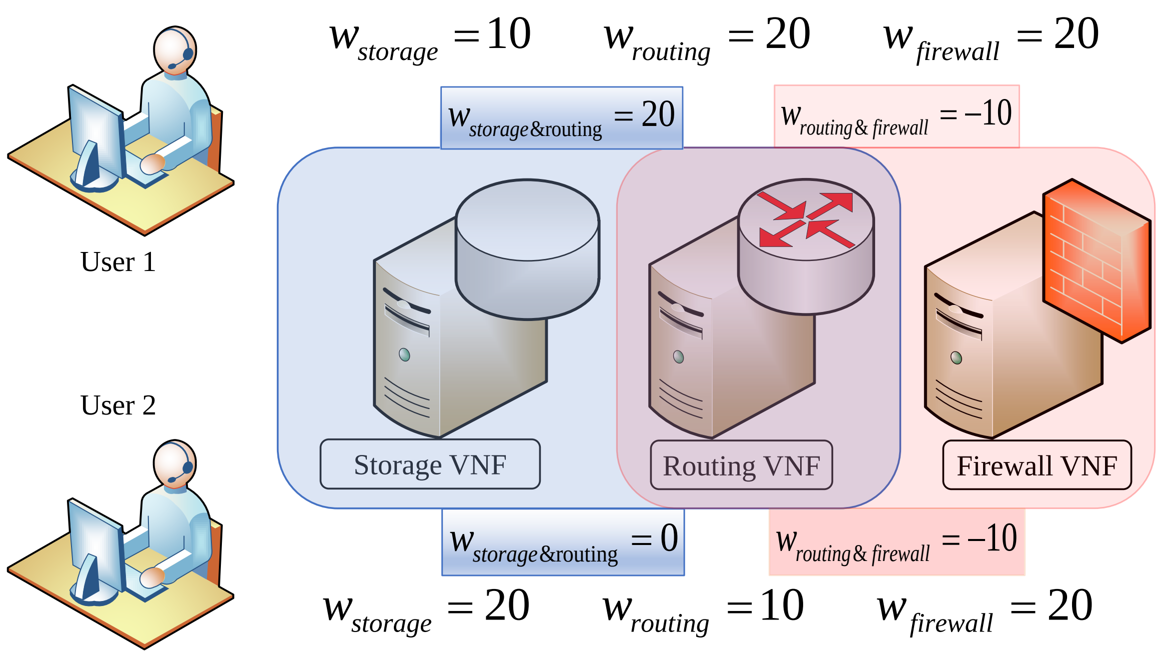


Fig. 3.4 Example for tree valuation request of users

Because of the budget of bidder i is a commen message, bidder i can not affect the value $\lfloor \frac{B_i^*}{p} \rfloor$ *of surplus demand* $D_i^*(p)$*. Bidder i only can decide the time to launch the auction which is the time to* $D_i^+(p) = 0$*. When* $p < v_i$*, each item of bidder i increase the profit of bidder i. When* $p > v_i$*, each item of bidder i decrease the profit of bidder i. The truthful auction ensure the non negative profit.*

Then we compare the profit of bidding price v_i *and bidding price* $b_i < v_i$*. When the price p goes up from* 0 *to* b_i *in the same execution process, the bidder misses some item of which price p is in* $[b_i, v_i]$*. Similarly, if the bidding price* $b_i > v_i$*, the bidder gets some item of which price p is in* $[v_i, b_i]$*, and these items only bring non-positive benefits. Therefore, no false auction will make bidder i get more profit than the truthful auction.*

Example 1 *Consider the tree valuation request of users as Figure 3.4, in which there are two users and three VNFs. The best choices for user 1 and user 2 are as following,*

$$(y^1_{sto}, y^1_{rou}, y^1_{fire}, x^1_{s\&r}, x^1_{r\&f}) = (1, 1, 0, 1, 0), \tag{3.17}$$

$$(y^2_{sto}, y^2_{rou}, y^2_{fire}, x^2_{s\&r}, x^2_{r\&f}) = (0, 0, 1, 0, 0). \tag{3.18}$$

From the example, it is an efficient allocation that user 1 buys the storage and routing while user 2 buys the firewall. From $y^1_{sto} = y^1_{rou} = 1$ *and Equation (3.9), we can get* $x^1_{a\&r} = 1$*. When computing the objective value, we consider the weight of edge and the obtimal objective value is 70. For this solution, the optimal welfare of the efficient distribution. The set of all possible* *results for user 1 and user 2 are {(1,1,0,1,0), (1,0,1,0,0), (0,1,1,0,1), (1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0)} and {(0,0,1,0,0), (0,1,0,0,0), (1,0,0,0,0), (0,1,1,0,1), (1,0,1,0,0), (1,1,0,1,0)}, respectively. Thus the social welfare of the above results is {70, 40, 50, 30, 60, 50}. It is obviously that the best choice is the result of our method.*

3.5 Numeral Experiment

In the section, we consider bundled resources including storage, routing, and firewall in Multiple-Walrasian tree valuation auction market. We generate prices as a random distribution matrix including the price and demand function. In our numeral experiment, we choose the different number of customers in $(0,150)$,VNFs in $(0,1500)$ and 100 providers to compare the performance among Tree valuation auction model, Backpack auction model and Reserve auction model. We choose the pricing rate $R_p = 10$ and the step size $\alpha = 0.7$ in Algorithm 1. Figure 3.5 shows the algorithm time with 100 providers and 1500 VNFs is increased with more customers. Figure 3.6 shows the algorithm time with 100 providers and 150 customers is increased with more customers. The red, blue and green lines indicate the auction time calculated by Tree valuation auction model and Backpack auction model and Reserve auction model, respectively. As shown in Figure 3.5, the auction time of Backpack auction model and Reserve auction model exceed the value in Tree valuation auction model. In Figure 3.5, the auction time of Backpack auction and Reserve auction model exceed the value from 600 VMFs in Tree valuation auction model.

Next, we compare the total social welfare among Tree valuation auction, Backpack auction and Reserve auction changing the customers and NVF number. As shown in Figure 3.7, the social welfare in Tree valuation auction is greater than the value of Backpack auction and Reserve auction. Because our model is based on VCG payment in which the winner will pay the value of loss of the losers caused by the winner. As shown in Figure 3.8, the total social welfare in Tree valuation auction is greater than the value of Backpack auction and Reserve auction. With the increase of customer number and VNF number are clearly on the rise. Because of the VCG payment, the number of customers and VNFs cannot affect the total social welfare.

Finally, we compare the profit of providers and customers among Tree valuation auction, Backpack auction and Reserve auction by changing customer number and VNF number. In Figure 3.9 and 3.10, with the increase of customer number and VNF number, the profit of customers in Tree valuation auction model is greater than the values in Backpack auction model and Reserve auction. Besides, there is the volatility in Backpack auction. Figure 3.10 shows that the profit of customers is clearly on the rise. We find that the performance of

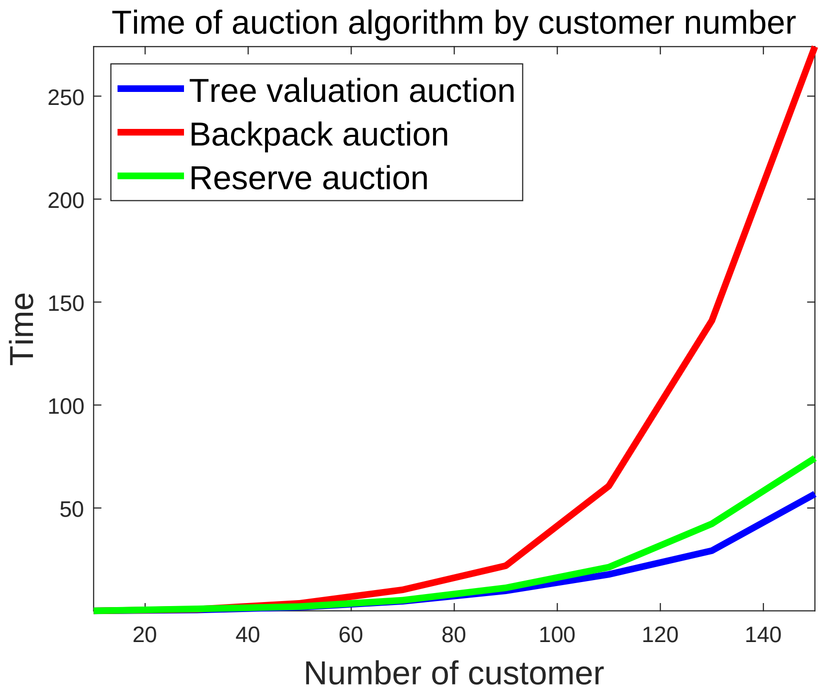


Fig. 3.5 Time of auction algorithm by customer number

the customers' profit in Tree valuation auction is better than the performance in Backpack auction and Reserve auction. The profit of provider is clearly on the rise with the increase of the number of customers and VNFs in Figure 3.11 and 3.12. We also find the profit of the provider in Tree valuation auction is better than the value in the Backpack auction and Reserve auction.

In general, the tree valuation double-auction mechanism is more effective than Backpack auction mechanism and Reserve auction because VNFs are more focused and relevant for customers in bundled double-auction model. From the result, we obtain the optimal solution is to maximize the total social welfare.

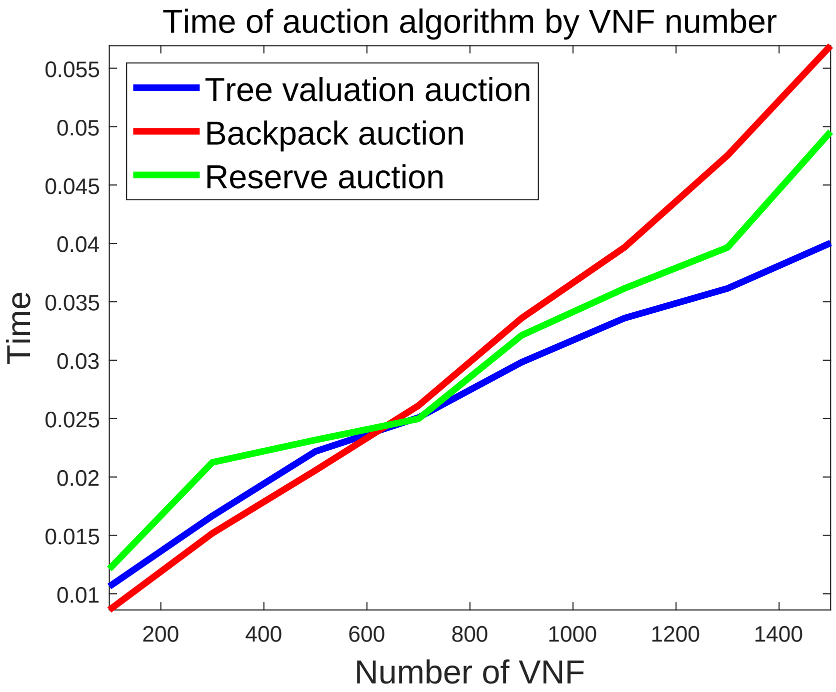


Fig. 3.6 Time of auction algorithm by VNF number

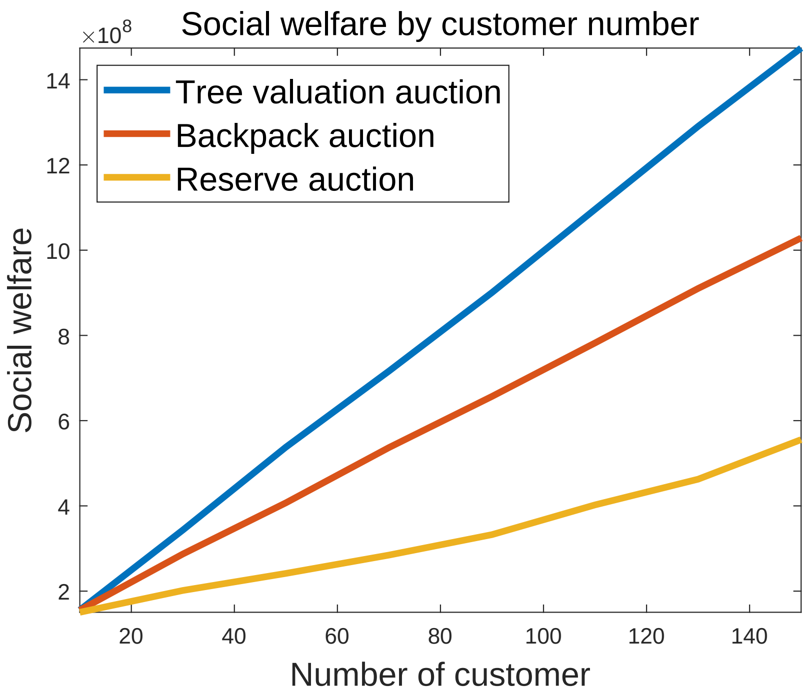


Fig. 3.7 Social welfare affected by customers

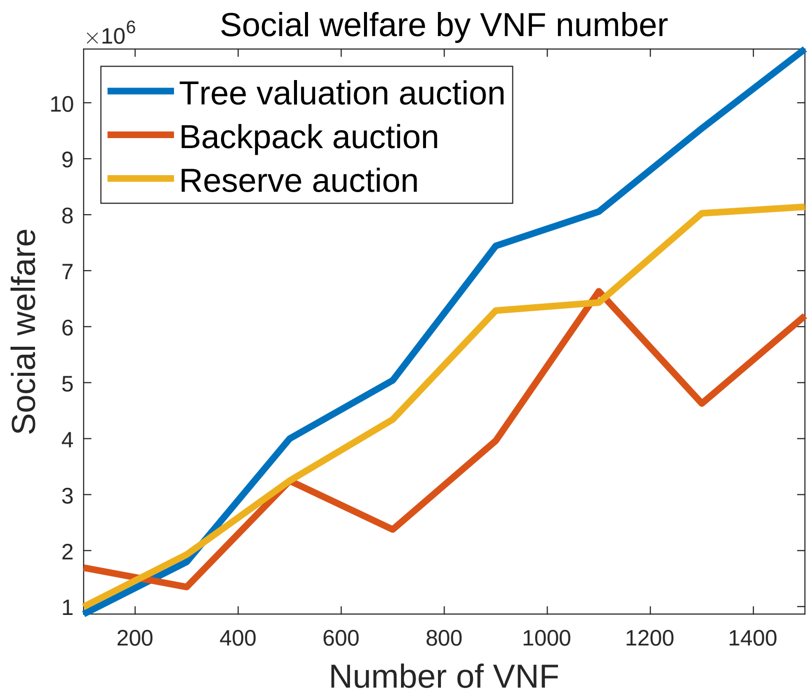


Fig. 3.8 Social welfare affected by VNFs

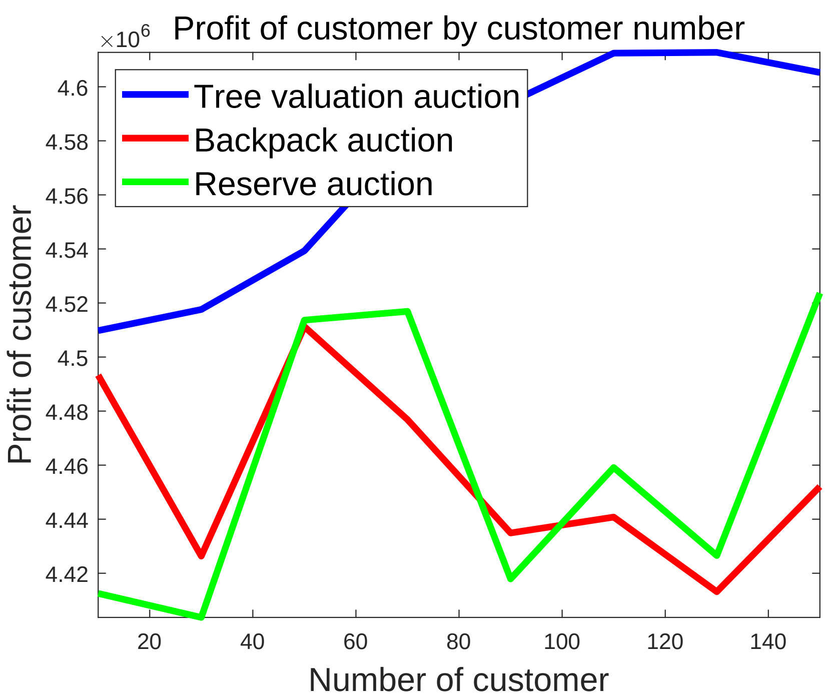


Fig. 3.9 Profit of customer affected by customers

‘

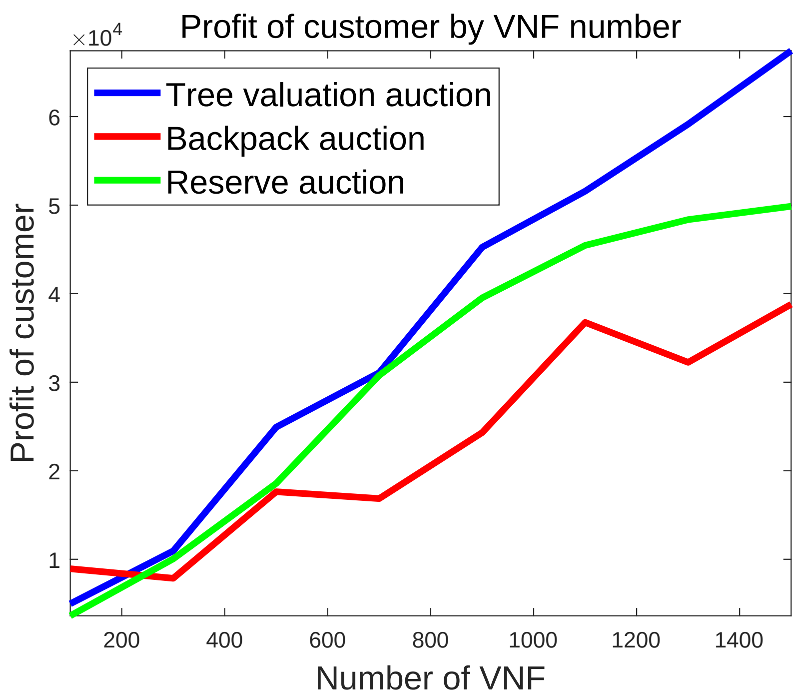


Fig. 3.10 Profit of customer affected by VNFs

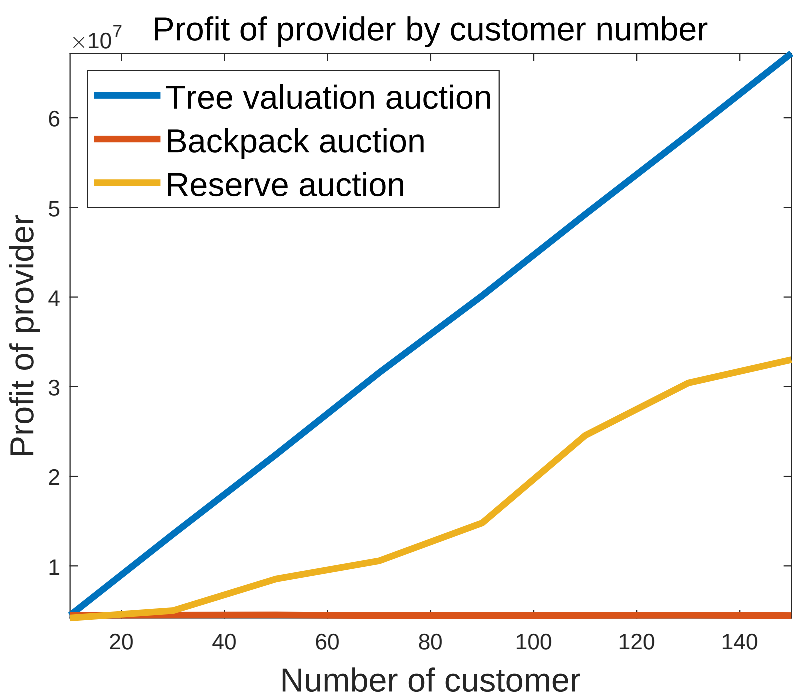


Fig. 3.11 Profit of provider affected by customers

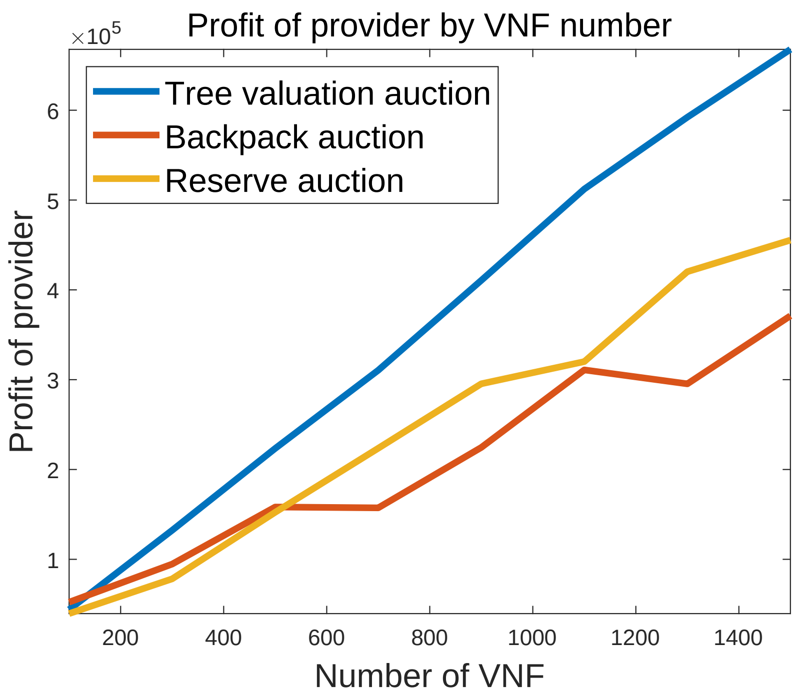


Fig. 3.12 Profit of provider affected by VNFs

Chapter 4

Dealer: An Efficient Pricing Strategy for Deep-Learning-as-a-Service

In this chapter, we present a pricing strategy called Dealer based on a game theory with virtualized resources {VCPU, VGPU, VTPU} allocation for guaranteeing resource utilization and service performance in cloud computing. Although the traditional pricing method can improve the resource utilization to schedule virtualization resources, it cannot guarantee the revenues of service providers [25, 83, 87]. Stackelberg-game-theory-based Dealer pricing strategy can achieve a higher efficiency with competitive bidding between the provider and users. Stackelberg competition game is non-cooperation game, in which the competitors are the leader and followers. Through Dealer pricing strategy, the revenue is higher than that in other methods. Consequently, we first design an efficient game theory model in our resource scheduling problem in which both users know all information of competitors. The main goal of our model is to maximize the revenues of the provider and users.

The remainder of this chapter is organized as follows. The related works are discussed in Section 4.1. We introduce the resource scheduling problem in cloud computing and also propose the MRAP resource scheduling model with multi-objective optimization function in Section 4.2. In Section 4.3, we design Stackelberg game theory mechanism containing Dealer decision algorithm and Dealer price adjustment algorithm to guarantee availability. We present some numerical results in Section 4.4.

4.1 Related Work

Recently, there is an increasing number of researchers to access the powerful computational resources [84, 104, 62, 69]. Wu et al. propose a parallel soft real-time scheduler called Poris which is to schedule VCPUs according to soft real-time applications in hypervisors. To guarantee the accomplishment ratio of tasks, the authors present parallel scheduler, group scheduler and communication-driven scheduler [99]. C. Prakash et al. propose a efficient scheduler to manage multi-type of resource including VCPU and memory. They also do some simulation to prove their scheduler have the higher performance [82].

Hong et al. propose a K-means algorithm which is very famous non-leader algorithm to schedule GPUs [50] Li et al. propose an optimal pricing strategy called Elastic pricing strategy to maximize the revenues of both cloud provider and users. They first analyze the differences between payments from users and costs of the VGPU resources [56]. Yao et al. propose a sharing scheduler with automated resource as VGPU called Auto-vGPU with configuring by themselves to reduce manual cost and to guarantee the service quality [105]. Zhao et al. propose a fine grained scheduler with VGPU resources for the high performance to share a GPU among multi-VMs. They also present a new strategy called VM scheduling strategy which can efficiently schedule workloads to different VGPU [115].

With the increasing virtualization of CPU and GPU, the users request more and more virtualization of professional computing chip including TPU. Jouppi et al. introduce the concept of TPU developed by Google, which can greatly accelerate the speed of logical inference process of CNN. However, TPU can not enhance the precision of training [34]. Eric et al. propose RNS TPU for the higher performance of neural networks [78].

According to the above content, we first consider three types of virtualized resources to schedule based on the pricing model. Recently, some researchers propose some pricing model. Sharma et al. develop a pricing strategy model for cloud resources allocation, in which they focus on the QoS (quality of service) from users and the revenue of the provider. They also employ the financial option theory to provide a high performance of resource allocation [88]. In 2015, Sharma et al. also propose a general formula called compound-Moore Law to maximize the revenue of both participants by using genetic algorithm and fuzzy logic[87]. Fang et al. propose a new pricing strategy to efficiently resource allocation for the maximal revenue in two monopoly IAAS cloud market including a provider and a broker [36]. Chiu et al. use a game theory in pricing situation, in which retailers compete with each other like a price war. Their model is a non-cooperative competitive model between retailers based on the owning demands [20]. Yuan et al. employ a zonal pricing strategy in distribution network based on the demand response [107]. Fang et al. solve a resource allocation problem considered as a non-convex optimization problem [35]. We combined the game theory method and pricing model to find an optimal solution for scheduling resources.

Table 4.1 Summary of notations

Symbols	Description
N	The number of users.
G^w	The generality of resources.
Q^w	The service quality of resources.
E^w	The energy consumption of resources.
D^w	The depreciation of resources j.
$\mathscr{T}$	The set of tasks fo user i.
d_{im_i}	The demand of user i.
T_{im_i}	The m_ith task of user i.
M_i	The number of tasks for user i.
$p^w_{im_i}$	The private price of user i of task T_{im_i} for resource w.
$r^w_{im_i}$	The ratio of resource w for task T_{im_i}.
R^w_i	The revenue of user i buying resource w.
p^w	The selling price of the provider for resource w.
c^w	The cost of the provider for resource w.
S^w	The URS for resource w.
R	The total revenue of the provider.

4.2 System Model

In this section, we model three types of resources allocation including VCPU, VGPU and VTPU in the cloud computing. We consider a cloud cluster as a provider in a market, in which resources {VTPU, VGPU, VCPU} are infinitely divisible. Three types of resources have difference properties and functions. We use G^w, Q^w, E^w and D^w denote the generality, the service quality, the energy consumption and the depreciation, respectively. Let w denotes the type of resource like C denoting VCPU, G denoting VGPU and T denoting VTPU. For example, as shown in Fig 4.1, resource VCPUs handle in turn any task, and resource VGPUs are capable of the processing parallel tasks. Resource VTPUs only deal with tasks related with the neural networks in machine learning. Therefore, the generality of resources meets $G^C > G^G > G^T$. Because the maximal performance of resource VGPUs per unit equipment is higher than the maximal performance of other resources, the service quality meets $Q^G > Q^C > Q^T$.

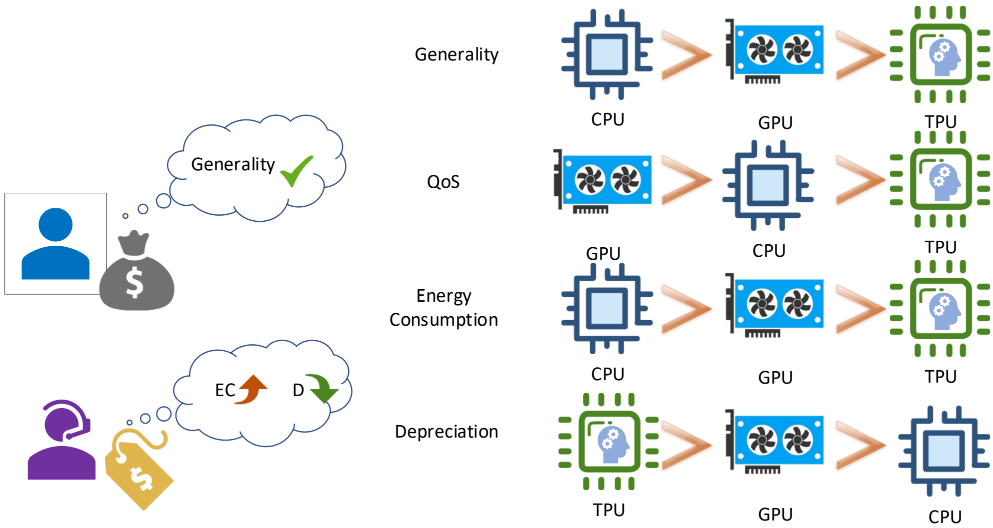


Fig. 4.1 Generality of three resources

Let S^w denotes URS (User request standard) as follows:

$$S^w = G^w + Q^w. \tag{4.1}$$

Similarly, there are two inequalities about the energy consumption and the depreciation as $E^T < E^G < E^C$ and $D^T > D^G > D^C$. We can calculate the cost of resource w given by:

$$c^w = E^w + D^w. \tag{4.2}$$

The provider sells resources with price p^w to users and meets $p^w > c^w = E^w + D^w$. Therefore, our first objective is to maximize the revenue of the provider is following:

$$\max_{p^w} R = \max_{p^w} \sum_w (p^w - c^w). \tag{4.3}$$

Users choose which type of resources to handle their tasks based on the service quality. We assume that each users have different number of tasks to complete. Let $\mathscr{U}$ denote the set of users numbered $1, \ldots, N$. We denote the set of tasks for user i by $\mathscr{T} = \{T_{i1}, T_{i2}, \ldots, T_{iM_i}\}$, where M_i is the total number of tasks for user i. Let $r^w_{im_i} \in [0,1]$ denotes the ratio of resource w for task T_{im_i}. User i pays for resource w to handle task T_{im_i} with private price denoted by $p^w_{im_i}$. Task T_{im_i} has the demand of resource w denoted by $d^w_{im_i}$ from the provider as follows:

$$
\begin{aligned}
d^w_{im_i} = \alpha_i - \beta_i p^w_{im_i} r^w_{im_i} \\
+ \sum_{j \neq i} (\beta_j r^w_{im_i} r^w_{jm_i} (p^w_{jm_i} - p^w_{im_i})) + \beta_0 S^w,
\end{aligned}
\tag{4.4}
$$

where $\alpha_i > 0$ is the scale of the market for user i as a market base. When both users give the private price at 0 and no further accompanying price α_i is the demand of user i. $\beta_i > 0, i = 0, 1, ..., N$ is the parameter which act as experience points.

We use R^w_i denote the revenue of user i. Therefore,

$$
R^w_i = \sum_{m_i=1}^{M_i} (p^w_{im_i} r^w_{im_i} - p^w) \cdot d^w_{im_i} + \eta_i S^w/2,
\tag{4.5}
$$

where $\eta_i > 0$ is the parameter. Our second objective function is to maximize the total revenue of users as follows:

$$
\max_{r^w_{im_i}} \sum_{w,i} R^w_i.
\tag{4.6}
$$

Finally, the revenue maximization problem in our model is given by,

$$
\max_{p^w} R; \tag{4.7a}
$$

$$
\max_{p^w, r^w_{im_i}} \sum_{w,i} R^w_i \tag{4.7b}
$$

$s.t.$

$$
p^w > E^w + D^w, \quad \forall w \in \{C, G, T\}; \tag{4.7c}
$$

$$
\sum_w r^w_{im_i} = 1, \qquad \forall i, \in \mathscr{U}, w \in \{C, G, T\}. \tag{4.7d}
$$

4.3 Algorithm

In this section, we propose Stackelberg competition game in which both users know all information of competitors. We consider the multi-objective equation (4.7) as a two level Stackelberg game between users and the provider. A Stackelberg game is a non-cooperation game, in which the competitors are leaders and the followers. In our model, we consider the provider and users as a leader and followers, respectively. In the game, the provider decides the price of resources to maximize its revenue. Then users make decisions to choose resources according to their tasks based on the prices.

4.3.1 Optimal decision for users

Algorithm 7 Dealer strategy of users

Input: Price of users and the provider for three type of resources
Output: Strategy of users

$r^{w}_{im_i} \leftarrow 0$;
$r^{C}_{im_i} \leftarrow 1$;
for $i \in \mathscr{U}$ **do**
  for $w \in \{G,T\}$ **do**
    while $r^{w}_{jm_i} \in [0,1]$ **do**
      Calculate the partial derivation of R^{w}_{i};
      if $\frac{\partial R^{w}_{i}}{\partial r^{w}_{i} j} > 0$ **then**
        $r^{w}_{im_i} \leftarrow 1$;
      else
        if $\frac{\partial R^{w}_{i}}{\partial r^{w}_{i} j} < 0$ **then**
          $r^{w}_{im_i} \leftarrow 0$;
        else
          Calculate $r^{w}_{oim_i}$ meets $\frac{\partial R^{w}_{i}}{\partial r^{w}_{im_i}} = 0$;
        end if
      end if
      if $r^{w}_{oim_i} < 0$ **then**
        $r^{w}_{oim_i} \leftarrow 0$;
      end if
      if $r^{w}_{oim_i} > 1$ **then**
        $r^{w}_{oim_i} \leftarrow 1$;
      end if
    end while
    $\tilde{r}^{C}_{im_i} \leftarrow 1 - \sum_{w \in G,T} r^{w}_{im_i}$;
  end for
end for

We use the backward induction method to solve the optimal decision for users. We assume that the provider have a best strategy p^{w}_{o} and user i can decide the optimal strategy $r^{w}_{oim_i}$.

$$\max_{p^{w}, r^{w}_{im_i}} \sum_{w,i} R^{w}_{i} \tag{4.8a}$$

$$s.t. \tag{4.8b}$$

$$p^{w} > E^{w} + D^{w}, \quad \forall w \in \{C,G,T\}; \tag{4.8c}$$

$$r^{w}_{im_i} \in [0,1], \qquad \forall i \in \mathscr{U}, w \in \{C,G,T\}. \tag{4.8d}$$

From Equation (4.5), we can see the function of revenue for user i is a continuous differentiable function. Next, we take the revenue function R_i^w partial respect to $r_{im_i}^w$.

$$\begin{aligned}\frac{\partial \sum_{w,i} R_i^w}{\partial r_{im_i}^w} = \frac{dR_i^w}{dr_{im_i}^w} = \frac{\partial((p_{im_i}^w r_{im_i}^w - p^w)\cdot d_{im_i}^w)}{\partial r_{im_i}^w} \\ = \frac{\partial d_{im_i}^w}{\partial r_{im_i}^w}(p_{im_i}^w r_{im_i}^w - p^w) + d_{im_i}^w(p_{im_i}^w - p^w),\end{aligned} \tag{4.9}$$

where,

$$\begin{aligned}d_{im_i}^w = &\sum_{j\neq i} \beta_j r_{im_i}^w r_{jm_i}^w (p_{jm_i}^w - p_{im_i}^w) + \beta_0 S^w + \alpha_i \\ &- \beta_i p_{im_i}^w r_{im_i}^w,\end{aligned} \tag{4.10}$$

and

$$\frac{\partial d_{im_i}^w}{\partial r_{im_i}^w} = \sum_{j\neq i} \beta_j r_{jm_i}^w (p_{jm_i}^w - p_{im_i}^w) - \beta_i p_{im_i}^w. \tag{4.11}$$

To achieve a maximal revenue of the provider, let $\frac{\partial \sum_{w,i} R_i^w}{\partial r_{im_i}^w} = 0$. Then we can achieve Equation (4.12) which is the relation between the optimal decision and the optimal price:

$$\begin{aligned}&\frac{p_o^w(\alpha_i - \beta_i p_{im_i}^w + \sum_{j\neq i}\beta_j p_o^w(p_{jm_i}^w - p_{im_i}^w)) + \beta_0 S^w}{p_{im_i}^w p_o^w - p^w} \\ &= \frac{\beta_i p_{im_i}^w - \sum_{j\neq i}\beta_j p_o^w (p_{jm_i}^w - p_{im_i}^w)}{p_{im_i}^w - p^w}.\end{aligned} \tag{4.12}$$

Therefore, we can obtain expressions based on rational simplification as Equation (4.13), given by,

$$p_o^w = \frac{\mathscr{A}}{\mathscr{B}} \quad \text{and} \quad p_o^w = \frac{\mathscr{C}}{\mathscr{D}}, \tag{4.13}$$

where $\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D}$ represent the following equation, respectively, given by,

$$\begin{aligned}\mathscr{A} = &p_o^w p_{im_i}^w (\sum_{j\neq i} 2\beta_j r_{jm_i}^w (p_{jm_i}^w - p_{im_i}^w) - 2\beta_i p_{im_i}^w \\ &+ \alpha_i + \beta_0 S^w),\end{aligned} \tag{4.14}$$

$$\begin{aligned}\mathscr{B} = &r_{im_i}^w(\alpha_i - \beta_i p_{im_i}^w + \sum_{j\neq i}\beta_j r_{jm_i}^w (p_{jm_i}^w - p_{im_i}^w)) \\ &+ \beta_0 S^w - \beta_i p_{im_i}^w + \sum_{j\neq i}\beta_j r_{jm_i}^w (p_{jm_i}^w - p_{im_i}^w),\end{aligned} \tag{4.15}$$

$$\begin{aligned}\mathscr{C} =& p_o^w \beta_i p_{im_i}^w - \sum_{j \neq i} \beta_j r_{jm_i}^w p_o^w (p_{jm_i}^w - p_{im_i}^w) \\ &+ p_{im_i}^w \beta_0 S^w - p_o^w \beta_0 S^w,\end{aligned} \tag{4.16}$$

and

$$\begin{aligned}\mathscr{D} =& \sum_{j \neq i} \beta_j r_{jm_i}^w (1 - p_{im_i}^w)(p_{jm_i}^w - p_{im_i}^w) + \beta_i p_{im_i}^w \\ &- (p_{im_i}^w - p_o^w)(\alpha_i - \beta_i p_{im_i}^w).\end{aligned} \tag{4.17}$$

4.3.2 Optimal pricing strategy for the provider

From strategies of users, the provider can change the price of resources to maximize its revenue. The provider derive the optimal price $p_o^w(r_{im_i}^w)$ according to the decision of user i for the task $T_{im_i}^w$ with resource w. The problem can be written as:

$$\max_{r_{im_i}^w} P(r_{im_i}^w); \tag{4.18a}$$

$$s.t.$$

$$p^w > E^w + D^w, \quad \forall w \in \{C, G, T\}; \tag{4.18b}$$

$$\sum_{w \in \{C,G,T\}} r_{im_i}^w = 1, \qquad \forall i, \in \mathscr{U} \tag{4.18c}$$

To simplify the problem, we use Lagrange method based on the only variant $r_{im_i}^w$. Therefore, the problem can be change into as follows:

$$\begin{aligned}\max_{r_{im_i}^w} H(r_{im_i}^w) =& P(r_{im_i}^w) + \lambda_1 (\sum r_{im_i}^w - 1) \\ &+ \lambda_2 (p^w - E^w + D^w)\end{aligned} \tag{4.19}$$

Now, we propose Algorithm 8 to achieve a best strategy for the provider.

4.4 Evaluation

In this section, we make some simulations for Dealer pricing strategy evaluation. First, we give a description of the setting in our simulation. Then we discuss the numeral results and the performance based on the simulation.

Algorithm 8 Dealer strategy for the provider

Input: Price of users and the provider for three types of resources
Output: Strategy for the provider
1: Calculate the partial of function $H(r^w_{im_i})$, respectively;
2: Let each partial function meet 0;
3: Extended Simultaneous Equations
4: **while** $(p^w > E^w + D^w)$ **do**
5: Calculate the maximum revenue;
6: **end while**

We do simulation experiment with MATLAB on a computer workstation platform which equips a CoreTM i7-6700 (8M Cache, up to 4GHz) CPU, 16GB installed size and 1×2 TB capacity. We choose the average value from $5 \sim 20$ times tests in each simulation. For comparison, we use the static prices from Amazone Elastic Compute Cloud (EC2) [85] and Elastic pricing model [56].

First, we compare the pricing strategies between Dealer strategy, Elastic pricing strategy and Amazon EC2 with different numbers of users. Figure 4.2 shows the provider's revenue when the number of users is changed with different models. It shows the provider's revenue with two types of resource including {VCPU, VGPU} and average number about $20 \sim 30$ tasks of each user. From the results, the revenue of the provider is increased with more users in both models. The solid green line shows the optimal solution to the provider's revenue maximization problem, and the solid red and blue lines are provider's revenue calculated by the Elastic pricing strategy model and Amazon EC2, respectively. As shown in Figure 4.2, when the number of users is greater than 15, the provider's revenue in the MRAP model is always higher than the revenue in other models.

Then, we also compare the pricing strategy performance with two types of resource including {VCPU, VGPU} between the MRAP model, Elastic pricing model and Amazon EC2 with different numbers of average tasks per user. As shown in Figure 4.3, we can see the provider's revenue changes increasingly when we increase the average number of tasks. The solid green, red and blue lines are provider's revenue calculated by MRAP model, the Elastic pricing strategy model and Amazon EC2, respectively. As shown in Figure 4.3, the provider's revenue in MRAP model is always far higher than the revenue in other models.

Next we compare the profit of users with different number of users when the average task numbers per each user are equal to $3,4,5,6$, respectively. As shown in Figure 4.4, the green, blue, black and red lines represent the profit of users when the average task numbers are equal to $3,4,5,6$, respectively. the profit of users is changed increasingly with the more average number of tasks. It also shows the profit of users is increasing, when the number of

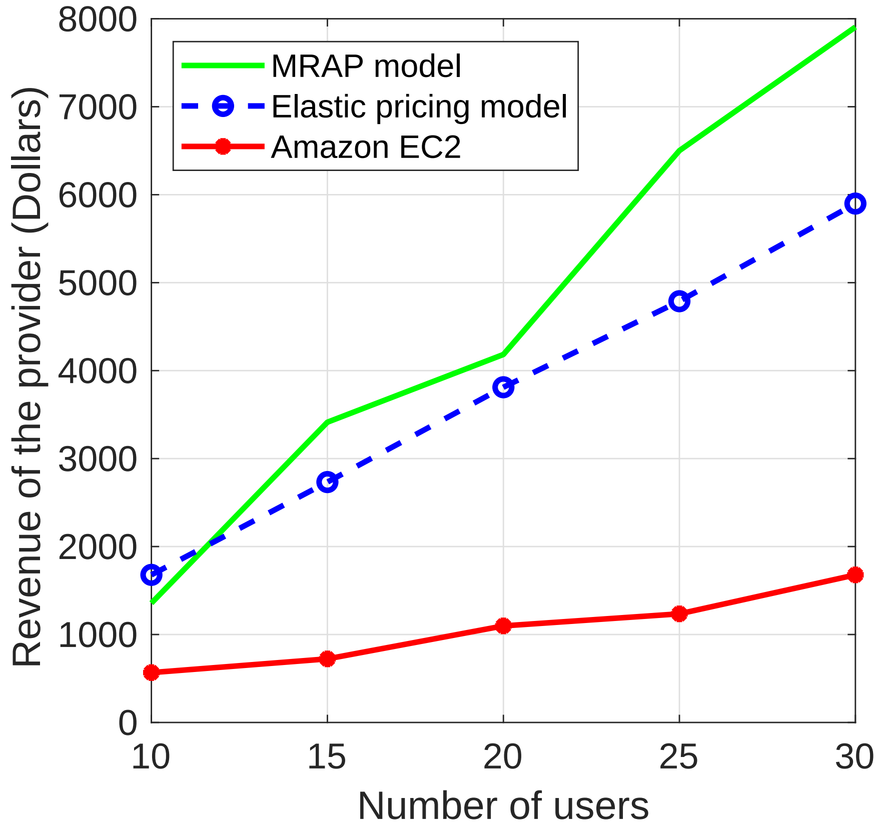


Fig. 4.2 Revenue of the provider with increasing number of users

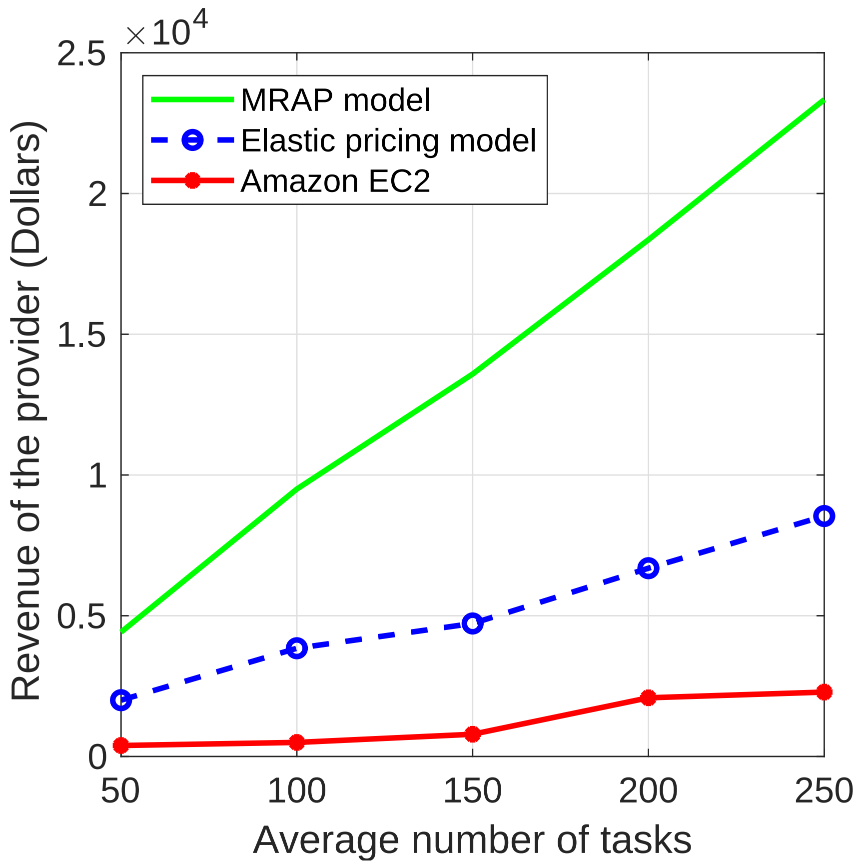


Fig. 4.3 Revenue of the provider

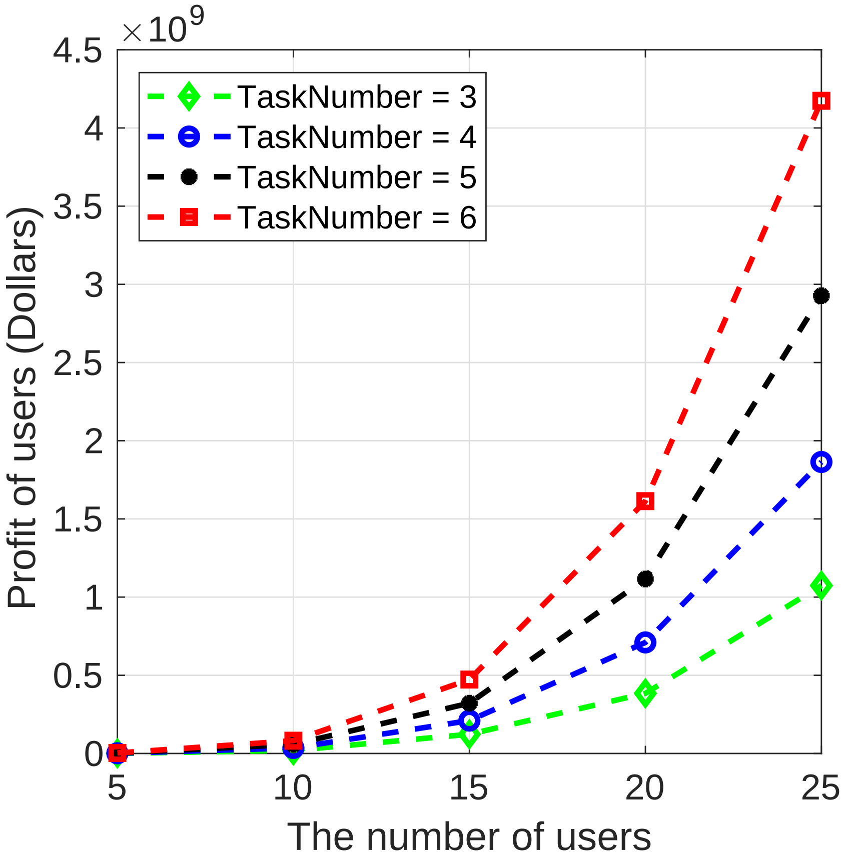


Fig. 4.4 User Profit

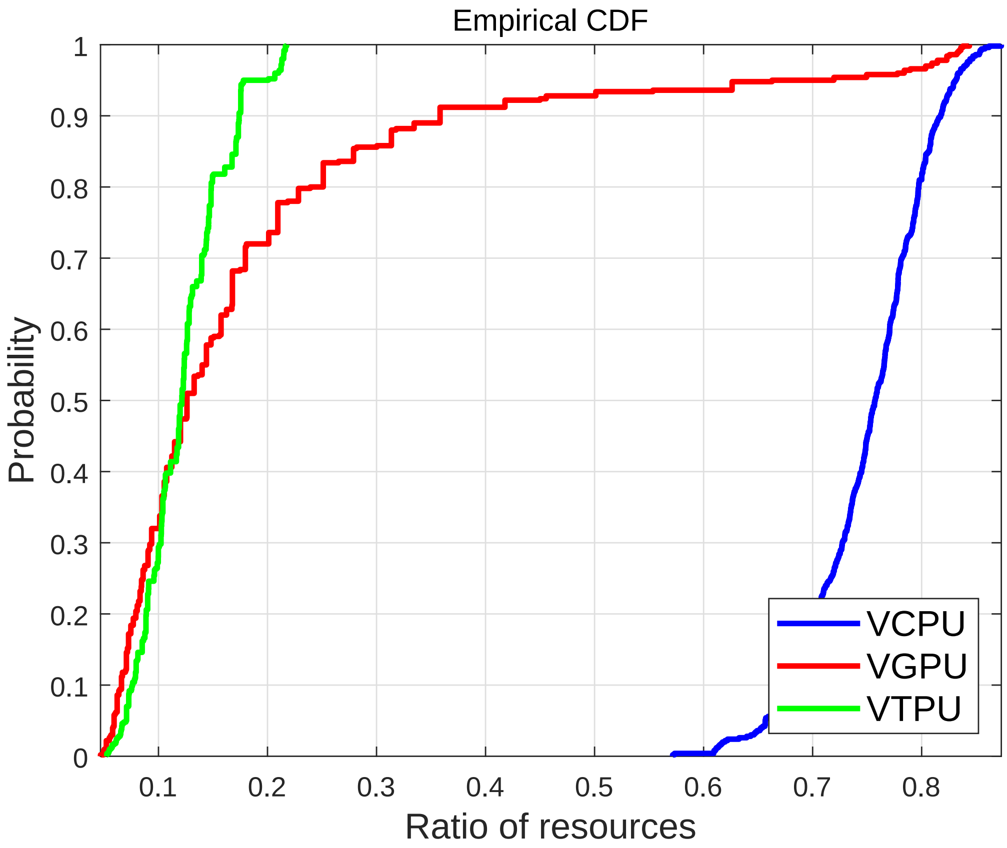


Fig. 4.5 Cumulative distribution function

users is growing. Especially, the rate of the profit is also increasing with the number of users to increase.

We finally compare the ratio of three types of resources include {VCPU, VGPU, VTPU} in the form of empirical CDF's (cumulative distribution function) analysis as shown in Figure 4.5. We choose users' average ratio of three types of resources. The green, red and blue line represent the probabilities of the ratio of VTPU, VGPU and VCPU, respectively. From the results, we can see the ratio of VTPUs only took a very small proportion about 22% with probability 1 due to the higher cost and the limited generality of VTPUs. However, because of the higher computational capabilities, there still be a few tasks request VTPU to with saving-time. VCPU as a popular option to users, at least about 55% users choose to buy VCPU to complete their tasks because of the lower cost and the service ability to do any tasks. The red line means the ratio of VGPU, and about 40% of all users, which is somewhere between VCPU and VTPU, choose to buy VGPU to fulfill the tasks.

In general, we find that the MRAP model performs better than other models. Our results show that we can achieve the main goal, which is to maximize the revenues of the three participants.

Chapter 5

Time-saving First: Coflow Scheduling for Data Center Networks

In this chapter, we focus on scheduling coflows in a dependent method with a constraint on guaranteeing deadline of jobs. We build a new coflow dependent model to present the entirety of coflows in a job. Specifically, we formulate a dependent and performance-guaranteed coflow optimization model. The objective of our proposed model is to minimize the overall completion time with the constraint of deadline guarantee. We take the dependency and job requirement into consideration and divide coflows into different stages in order to schedule coflows uniformly. To solve this problem, we present a heuristic Muli-objective Time-saving First (MTF) method to guarantee the job deadline. We first model coflows on their requirement and stages. After that we schedule coflows in a seamlessly most valuable way by a prioritized method. MTF method can identify the dependency of coflows on their requirements. Finally, we conduct extensive simulation to evaluate the performance of our proposed method. The simulation results show that our proposed method can reduce the job completion time. In the meantime, it can improve the ratio of job completion compared with the short job first method.

The rest of the chapter is organized as follows. In section 5.1, we present our related works. We model the coflow problem in Data Center network and propose MTF scheduling method with the constrains of performance guarantee in section 5.2. We conduct simulations in section 5.3.

5.1 Related work

Improving network performance is a significant thing in improving the performance of applications for Data Center networks. Orchetra is perhaps the first work that explains collective behaviors of flows of job-level concepts when optimize the flow transfers in Data Center cluster[23]. Then Hong present Preemptive Distributed Quick OW scheduling, a protocol designed to complete OWs quickly and meet OW deadlines[49]. However, traditional methods cannot improve application performance by relying on improving flow completion time. Hence, Chowdhury first considers coflow scheduling to improve coflow completion time, which can lead to a better application performance. Their works summarize the traffic patterns and coflow dependency in Data Center networks and propose the concept of coflow explicitly[21]. Zats[110] presents a new cross-layer network stack to reduce the long tail of coflow completion times. Then, recent solutions start to apply the coflow concept in their network optimizations. For example, Varys[24] proposes a heuristic method to schedule coflows with satisfaction on deadline. However, authors only focus on scheduling while neglecting an obbligato part-routing, which makes these solutions defective. Aalo[22] tries to strike a better balance and schedule coflows efficiently without prior knowledge. Aalo performs comparative to schedule like Varys. RAPIER[116] tries to reduce coflow completion time in both scheduling and routing. And then, Chen[19] designs a new utility based scheduler to improve the utility in the course of the scheduling. CODA[112] proposes a machine learning basic coflow recognition method. Li[62] derives an algorithm with the online multi-coflow routing and scheduling, called OMCoflow, and proves that it has a comparatively good rival ratio. GRAVITON is a coflow scheduler to improve coflow completion Time [52]. Some researchers optimize networks using multi-objective functions[93, 54]. They study a multi-objective optimization problem for minimizing the response time and the energy consumption while maximizing the profit.

However, none of the above methods considers the coflows as dependent ones, which might help to improve service performance largely. We extend above researches to formulating a coflow model with constrains of deadline and network capacity using multi-objective functions and the penalty function. To solve this model, we lean upon a heuristic MTF algorithm to schedule the valuable coflow.

5.2 System model and problem formulation

In this section, we present our optimization framework using the multi-objective function based on a new hypothesis, and propose a heuristic algorithm.

5.2.1 System model

We begin with introducing the MTF model based on the multi-objective mathematical optimization.

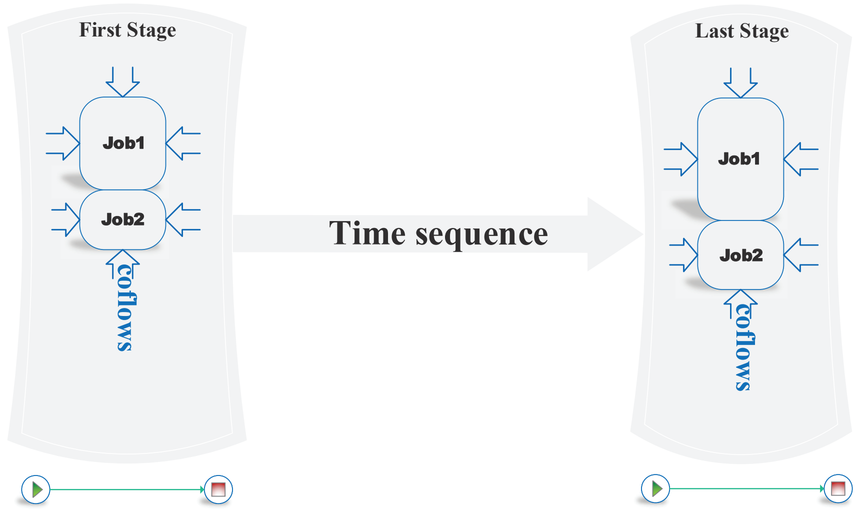


Fig. 5.1 MTF coflow scheduling

In the chapter, we consider a Data Center consisting of N servers, as shown in Fig. 5.1. Let $\mathscr{S} = \{s_1, s_2, ..., s_N\}$ denote the set of servers in the Data Center. To indicate the link of bandwidth capacity, let c_n denote the bandwidth capacity for $n-th$ server. Let $[\mathscr{T}_0, \mathscr{T}_1]$ be a time interval, during which a set $\mathscr{J} = \{J_1, J_2, ..., J_M\}$ of jobs have to be finished. It is worth noting that each job in clouds has a self-deadline to draw a clear distinction between the primary job and the secondary job. To distinguish such deadline respectively, let T_m^D denote the deadline by job J_m. Each job $J_m \in \mathscr{J}$ has to be handled within the internal time $[T_m^B, T_m^D]$. Here, T_m^B represents the beginning time of the $m-th$ job. Similarly, Let T_m^C denote the arrive time of the job J_m. Accordingly, there is a relationship of the size $\mathscr{T}_0 \le T_m^B \le T_m^C \le T_m^D \le \mathscr{T}_1$.

It is known that many jobs include multiple computational processes. Due to the complex constraint between these coflows, multiple coflows can be affected by a job. Accordingly, let $\mathscr{F}_m$ represent the coflow set of job J_m, and $f_{m,i}$ denote the $i-th$ coflow of job J_m in the $\mathscr{F}_m$. Let $D_{m,i}^{n,n'}$ denote the demand of the coflow $f_{m,i}$,which is transfered from server s_n to server $s_{n'}$. Correspondingly, let $B_{m,i}^{n,n'}(t)$ denote the bandwidth of demand $D_{m,i}^{n,n'}$ at time t. Important notations used in the chapter are as show in TABLE I. As we know, every demand has to be finished before the coflow completed in the pre-specified deadline T_m^D. Accordingly, we can denote the time of the coflow $f_{m,i}$ by $T_{m,i}^C$. For purposes of simplifying our model, we propose three hypotheses as follows:

Table 5.1 NOTATIONS

Symbol	Description
N	The number of servers.
M	The number of jobs.
L	The number of stages.
$\mathscr{S}$	The set of servers.
s_n	the $n-th$ server in the set $\mathscr{S}$.
c_n	The bandwidth capacities for $n-th$ sever.
$\mathscr{J}$	The set of jobs.
J_m	The $m-th$ job in the set $\mathscr{J}$.
T_m^D	Deadline of the $m-th$ job.
T_{m,k_l}^C	Completed time of the m-th job for the $k-th$ coflow at the $l-th$ stage.
T_m^B	Begin time of the $m-th$ job.
T_m^C	Completed time of the $m-th$ job.
$T_{m,l}^C$	Completed time of the m-th job for coflows in the $l-th$ stage
$\mathscr{F}_m$	The set of coflows for the $m-th$ job.
f_{m,k_l}	The $i-th$ coflow for the $m-th$ server.
$D_{m,k_l}^{n,n'}$	Demand of $m-th$ job for the k-th coflow in $l-th$ stage from server n to server n'.
$B_{m,k_l}^{n,n'}(t)$	Bandwidth of $k-th$ coflow in $l-th$ stage for the $m-th$ job from server n to server n' at time t.

I. All coflows have to work by stages. Otherwise, coflows are classified L stages according to the time sequence. As shown in Fig.5.1, while every stage contains different coflows, every coflow contains different job.

II. Every coflow at the stage $l \in L$ has to start at the same time. We obtain a strict mathematical expression. There are K_l coflows in the $l \in L$ stage, whose corresponding conditions as follows:

$$\sum_{l=1}^{L}\sum_{k_l=1}^{K_l} k_l = I, \quad \forall f_{m,k_l} \in \mathscr{F}_{m,K_l}, \forall J_m \in \mathscr{J}. \tag{5.1}$$

$$T^C_{m,l} = \max_{k_l \in \{1,\ldots,K_l\}} T^C_{m,k_l}, \quad \forall J_m \in \mathscr{J}, \forall f_{m,l} \in \mathscr{F}_{m,l}. \tag{5.2}$$

$$T^C_m = \sum_{l=1}^{L} T^C_{m,l}, \quad \forall J_m \in \mathscr{J}. \tag{5.3}$$

Here, I represents the number of coflows.

III. Coflows of the same job are independent of each other. In the other words, coflows in different jobs can't influence each other.

5.2.2 Problem Formulation

In this section, our goal is to minimize the average coflow completion time. In addition, we ensure that each coflow at the same stage finishes at the same time by minimizing the variance between T^C_{m,k_l} and $T^C_{m,l}$, for all $J_m \in \mathscr{J}$.

In order to minimize the average completion time, we can definite the main objective function as follow:

$$\min \quad \frac{1}{M}\sum_{m=1}^{M}(T^C_m - T^B_m) \tag{5.4}$$

From Equation (5.2) and (5.3), we can denote the average completion time T^{CC} as following:

$$T^{CC} = \frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{L}(\max_{k\in\{1,\ldots,K\}} T^C_{m,k_l} - LT^B_m), \tag{5.5}$$

The existence of Hypothesis II weakens the first objective. Hence, we need to establish the second objective to limit the average completion time effectively. Mathematical function is as follows:

$$\min \quad \sum_{l=1}^{L}\sum_{k=1}^{K_l}(T^C_{m,k_l} - T^C_{m,l})^2, \quad \forall J_m \in \mathscr{J} \tag{5.6}$$

Changing by the same methods as T^{CC}, let

$$T^V = \sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{k=1}^{K}(\max_{k\in\{1,\ldots,K\}} T^C_{m,k_l} - T^C_{m,k_l})^2, \tag{5.7}$$

denote the variance between T^C_{m,k_l} and $T^C_{m,l}$. The smaller T^V, the better the stability is. Therefore we aim to minimize the value of T^V.

In order to simplify the multi-objective optimization, we can transform the multi-objective function to single-objective function by using the punitive function[75]. So we choose to define a punitive function as follows:

$$P^F = \max_{m\in\{1,\dots,M\}} \{ \sum_{m=1}^{M}\sum_{l=1}^{L} \omega_1 (\max_{k\in\{1,\dots,K\}} T^C_{m,k_l} - LT^B_m), \quad \frac{\omega_2}{K}\sum_{l=1}^{L} (\max_{k\in\{1,\dots,K\}} T^C_{m,k_l} - T^C_{m,k_l})^2 \}. \tag{5.8}$$

Then it holds that

$$\min \omega_1 T^{CC} + \omega_2 T^V + \gamma P^F. \tag{5.9}$$

with a penalty parameter $\gamma \geq 0$, weight parameter ω_1 and ω_2. Penalty parameter can tell us whether the current way is right or not. When γ is infinitely great, every objective of the multi-objectives tends to be infinitely small and vice verse. The weight parameter ω_1 and ω_2 play roles in coinciding magnitude of multiple goals. If one coflow obtains bandwidth in the special stage, it can be known from Hypothesis I that other coflows in the subsequent stage can not obtain any bandwidth. Hence, the first conditional function is

$$B^{n,n'}_{m,l'}(t) = \sum_{l'\leq l\leq L} B^{n,n'}_{m,l}(t), \tag{5.10}$$

From the basic request of coflow scheduling, we know that the cumulative usage amount of bandwidth for each server $s_n \in \mathscr{S}$ ensures not to exceed the corresponding bandwidth capacity c_n. The corresponding condition as follow:

$$\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{\substack{n'=1\\ n'\neq n}}^{N} B^{n,n'}_{m,k_l}(t) \leq c_n, \tag{5.11}$$

In order to ensure that the schedule is executable, which means all coflows in the job J_m should be accomplished in the effective time $[\mathscr{T}_0, \mathscr{T}_1]$, we define

$$\int_{T^B_m}^{T^C_m} B^{n,n'}_{m,k_l}(t)dt = D^{n,n'}_{m,k_l}, \tag{5.12}$$

Let function $A^{n,n'}_{m,k_l}(t)$ denote the original function of the $B^{n,n'}_{m,k_l}(t)$. According to the role of definite integral, we can transform Equation (5.12) to

$$A^{n,n'}_{m,i}(T^C_m) - A^{n,n'}_{m,i}(T^B_m) \geq D^{n,n'}_{m,i}, \tag{5.13}$$

In conclusion, the original optimization problem can be explained as follows:

$$\min \quad \omega_1 T^{CC} + \omega_2 T^V + \gamma P^F \tag{5.14}$$

$$s.t. \quad \sum_{m=1}^{M} \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \sum_{l=1}^{L} \sum_{k=1}^{K} \frac{D_{m,k_l}^{n,n'}}{T_{m,k_l} - T_{m,k_l-1}} \leq c_n; \tag{5.15}$$

As we know, the deadline-aware coflow scheduling is an NP-hard problem[64, 33]. Hence, we can use following algorithm to solve these problems.

5.2.3 Algorithms

In this section, we present the MTF algorithm to minimize coflow completion time by distributing coflow bandwidths. We will discuss how to distribute bandwidth and calculate the minimum completion time for multi-coflow using the heuristic algorithm. As shown in Algorithm 9 that presents the detail of distributing bandwidth process, the mean of N is the number of jobs. From Line 4 to Line 10, it is a little FOR loop with the purpose of finding effective coflows. From Line 14 to Line 20, we calculate the sum of coflows.

Then, we analyze the approximate value of MTF algorithm in section 5.3.

5.3 Evaluation

In this section, we will evaluate our results and compare our results with that of a equal distribution model and a random allocation model.

- Equal distribution model: every coflow for all servers has the same bandwidth to fulfill their jobs. Otherwise, it completes on a share and share alike basis.
- Random allocation model: bandwidths are randomly assigned to all coflows to guarantee the completion of their mission.

As shown in the Fig.5.2, the completion time (about 900s) for more jobs in MTF model is considerably lower than the completion time of equal distribution model and random allocation model (about 8000s and 6000s, respectively). Hence, MTF model is more geared to multi-tasking system than other models. In Fig.5.3, we observe a trend that the completion time of MTF model almost decreases with the coflow bandwidth. However, the time tends towards stability with a high level in other models. When the number of servers is less than about 10, completion time is clearly unstable for all models because coflow is a collection

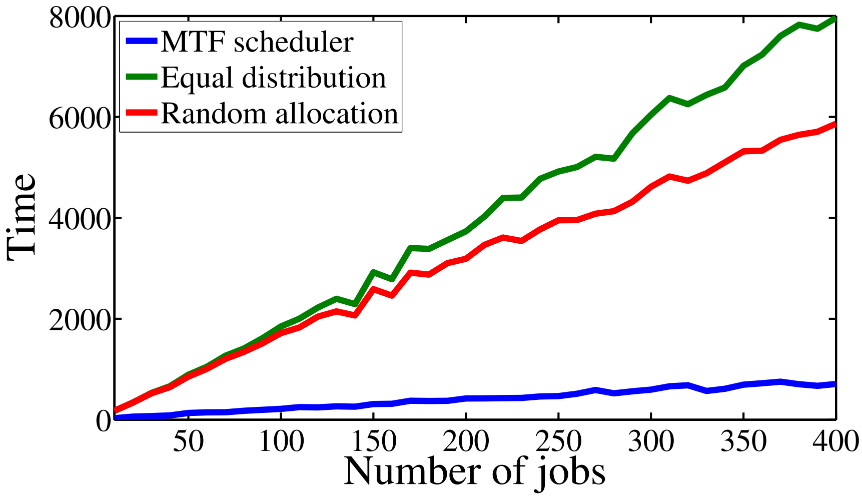


Fig. 5.2 Impact of Job

of some groups of clusters. Compared with equal distributed model and random allocation model, MTF model is more applicable for multi-sever architecture. We have main observation from Fig.5.4 that completion times of equal distribution model and random allocation model are clearly decreasing with bandwidths, where the highest is about 5000s, and the lowest is about 2500s. In MTF model, however, the speed of descent is relatively stable within $500-200$s. Thus, we can obtain the negative correlation between the bandwidth and the completion time.

To evaluate how the performance of MTF model is affected by the coflow size in datacenter clusters. The Fig.5.5 shows the change in completion time over the period from 2 coflow to 50 coflow. Thus, the size of coflow has more effect on distribution model and random allocation model than MTF model. The completion time of other two models is increasing with the increment of the coflow size. However, time condition tends towards stability in MTF model. In other words, MTF model has a better performance in multi-coflow system.

In each round of simulation, we consider the number of coflows in range from 2 to 4 with 10 servers in networks. Fig.5.6 shows our comparison with other two models. In the equal distribution model and the random allocation model, we can see from the figure that the gradient of time increment is about 45 degrees which is three times of our result. Hence MTF model keeps relatively stable performance with different demands. The results show that the MTF algorithm outperforms a well-known algorithm for multi-objective optimization.

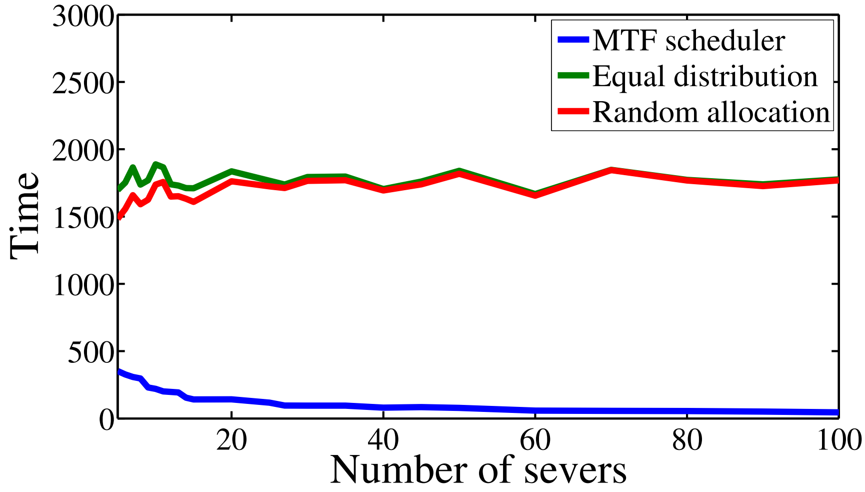


Fig. 5.3 Impact of Server

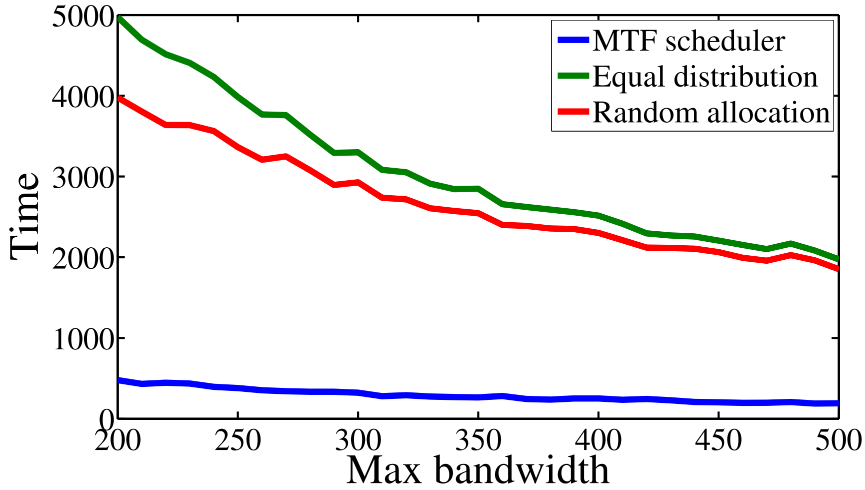


Fig. 5.4 Impact of Bandwidth

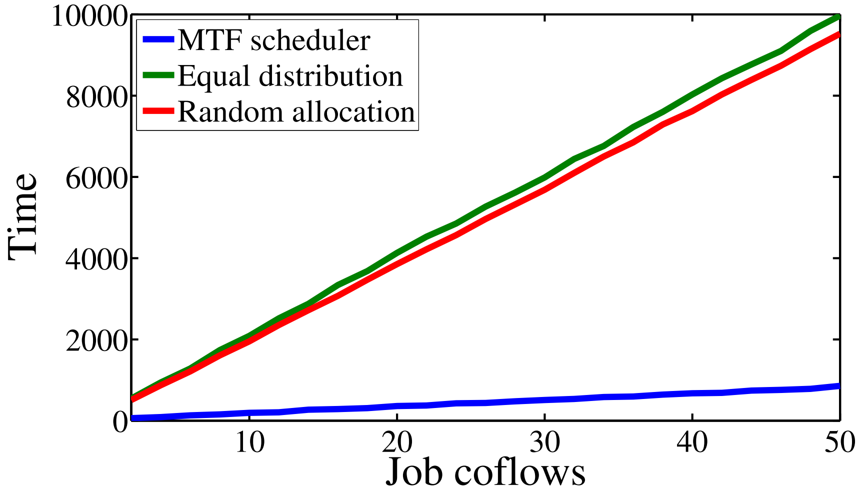


Fig. 5.5 Impact of coflow

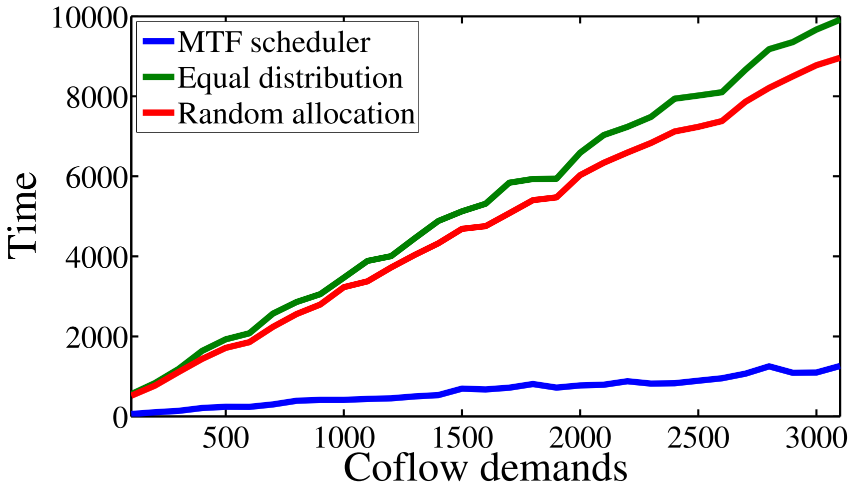


Fig. 5.6 Impact of Demand

Algorithm 9 MTF Scheduler

Input: number of jobs, maximum bandwidth, set of sever, maximum flows, minimum flows, maximum demand, minimum demand.
Output: totaltime.

```
for j in range(N) do
    bandwidth←maxbandwidth;
    while bandwidth do
        for k in range(num_jobs) do
            if jobfinished[k] < len(jobflows[k]) then
                if servers[k][jobfinished[k]] == j then
                    serverflow.append(k);
                end if
            end if
        end for
        if len(flow) ≠ 0 then
            maxsum← 0;
            maxflow← −1;
            for k in flow do
                flowsum ←sum(jobflows[k]);
                if maxsum < flowsum then
                    maxsum ← flowsum;
                    maxflow← k;
                end if
            end for
            if jobflows[maxflow][jobfinished[maxflow]]
            > bandwidth then
                jobflows[maxflow][jobfinished[maxflow]]
                − = bandwidth;
                bandwidth ← 0;
            else
                bandwidth − =
                jobflows[maxflow][jobfinished[maxflow]];
                jobflows[maxflow][jobfinished[maxflow]]← 0;
                jobfinished[maxflow] + = 1;
            end if
        else
            break;
        end if
    end while
end for
for j in unfinishedflow do
    if jobfinished[j] >= len(jobflows[j]) then
        index ← np.argwhere(unfinishedflow= j);
        unfinishedflow ← np.delete(unfinishedflowindex);
    end if
end for
if len(unfinishedflow) == 0 then
    print totaltime;
    break;
end if
```

Chapter 6

Conclusions and Future Directions

This chapter draws the conclusions of this thesis and give some future works. Through Chapter 2~4, we introduce my solutions of applying different emerging technologies in building the next generation disaster response system.

In the first part, we formulate an auction model that connects VNFs to maximize the profits of the three participants. We use a double-auction method called the DARA mechanism to schedule resources in the NFV market. The DARA method is effective according to the theoretical analysis and performance evaluation. Compared with the single-auction model, the DARA model increases the profits of customers and resource suppliers in NFV markets.

In the second part, we first propose a new model with the Multiple-Walrasian auction mechanism based on the tree valuation in a graph. Then we consider different number of VNFs into bundled tree nodes and price them through VCG payment to maximize the social welfare. To solve this problem, we use novel algorithms to schedule and price tree nodes. Moreover, we conduct comprehensive simulations to evaluate our proposed method and the results confirm that our method outperforms Backpack model and Reserve model with respect to the social welfare.

In the third part, first, we formulate a game-theory-based resource allocation model called MRAP to maximize the revenue of users and the provider. In our model, we schedule three types of resources including VCPU, VGPU and VTPU for different kinds of tasks from users. Second, we solve the model by simplifying it into multi-objective optimization problem and a two level Stackelberg competition game. Third, we propose two efficient algorithms called Dealer to find the optimal solution of resources allocation problem. We also evaluate our method compared with Elastic pricing model and Amazon EC2, and our method can bring better revenue than others.

In the forth part, we formulate a dependency coflow model with constrains of deadline and network capacity. To solve the problem, we resort to a heuristic method called MTF scheduler to schedule the valuable coflow. MTF method turns out to be effective by theoretical analysis. Compared with the equal distributed model and the random allocation model, MTF model, which adopts the multi-objective function, have a better performance with multi-servers and multi-tasking system in Data Center clusters. The standard of education of the author, it has still some deficiency to be improved and perfected.

In future research, we will focus on finding a more flexible and cost-saving resource allocation algorithm based on the physical topology of Data Centers. There are two main traffic types in Data Center: One is the association between Data Centers and users, and the other is the association between Data Centers and Data Centers, which are the main traffic of Data Centers today. Therefore, there are two types of virtual network capabilities in Data Center environment: virtual network capabilities within Data Centers and virtual network capabilities between Data Centers. One of the biggest characteristics of Data Center is the regularity of its physical topology, and its typical architecture is a multi-layer tree. In addition, Data Centers are usually homogenous, meaning servers have the same computing, storage, and communication capabilities. Using the function of Data Center, we can design a resource allocation algorithm suitable for Data Center environment. The future research direction is to consider embedding network function chains or trees into Data Center. The main content is based on the tree topology of Data Center network, using different strategies and algorithms to embed more elastic service tree in Data Center, find the optimal allocation scheme.

References

[1] (2012). Framework of network virtualization for future networks.

[2] (2014). Cisco NFV Solution: Enabling Rapid Service Innovation in the Era of Virtualization.

[3] An, D., Yang, Q., Yu, W., Yang, X., Fu, X., and Zhao, W. (2017). SODA: Strategy-Proof Online Double Auction Scheme for Multimicrogrids Bidding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–14.

[4] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.

[5] Bari, F., Chowdhury, S. R., Ahmed, R., Boutaba, R., and Duarte, O. C. M. B. (2016). Orchestrating Virtualized Network Functions. *IEEE Transactions on Network and Service Management*, 13(4):725–739.

[6] Basta, A., Kellerer, W., Hoffmann, M., Morper, H. J., and Hoffmann, K. (2014). Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem. In *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, AllThingsCellular'14, pages 33–38, New York, NY, USA. ACM.

[7] Batalle, J., Riera, J. F., Escalona, E., and Garcia-Espin, J. A. (2013). On the Implementation of NFV over an OpenFlow Infrastructure: Routing Function Virtualization. In *IEEE SDN for Future Networks and Services (SDN4FNS)*, pages 1–6.

[8] Beck, M. T. and Botero, J. F. (2015). Coordinated Allocation of Service Function Chains. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[9] Bega, D., Gramaglia, M., Banchs, A., Sciancalepore, V., Samdanis, K., and Costa-Perez, X. (2017). Optimising 5G infrastructure markets: The business of network slicing. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9.

[10] Borjigin, W., Ota, K., and Dong, M. (2018). In broker we trust: A double-auction approach for resource allocation in NFV markets. *IEEE Transactions on Network and Service Management*, 15(4):1322–1333.

[11] Boucadair, S., Lopez, D., Telefonica, I., Guichard, D., and Pignataro, C. (2014). Service Function Chaining: Framework & Architecture .

[12] Bouras, C., Ntarzanos, P., and Papazois, A. (2016). Cost modeling for SDN/NFV based mobile 5G networks. In *8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 56–61.

[13] Buyya, R. (2009). Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. In *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 1–1.

[14] Cai, Y. and Daskalakis, C. (2011). Extreme-Value Theorems for Optimal Multidimensional Pricing. In *IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 522–531.

[15] Candogan, O., Ozdaglar, A., and Parrilo, P. (2018). Pricing equilibria and graphical valuations. *ACM Trans. Econ. Comput.*, 6(1):2:1–2:26.

[16] Candogan, O., Ozdaglar, A., and Parrilo, P. A. (2015). Iterative auction design for tree valuations. *Operations Research*, 63(4):751–771.

[17] Candogan, O. and Pekeč, S. (2018). Efficient allocation and pricing of multifeatured items. *Management Science*, 64(12):5521–5543.

[18] Chen, J., Chen, X., and Song, X. (2002). Bidder's Strategy Under Group-buying Auction on the Internet. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 32(6):680–690.

[19] Chen, L., Cui, W., Li, B., and Li, B. (2016). Optimizing coflow completion times with utility max-min fairness. In *The 35th Annual IEEE International Conference on Computer Communications(INFOCOM)*, pages 1–9.

[20] Chiu, C.-H., Choi, T.-M., and Li, D. (2009). Price wall or war: The pricing strategies for retailers. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(2):331–343.

[21] Chowdhury, M. and Stoica, I. (2012). Coflow: A networking abstraction for cluster applications. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 31–36, NY, USA.

[22] Chowdhury, M. and Stoica, I. (2015). Efficient coflow scheduling without prior knowledge. *SIGCOMM Comput. Commun. Rev.*, 45(4):393–406.

[23] Chowdhury, M., Zaharia, M., Ma, J., Jordan, M. I., and Stoica, I. (2011). Managing data transfers in computer clusters with orchestra. *SIGCOMM Comput. Commun. Rev.*, 41(4):98–109.

[24] Chowdhury, M., Zhong, Y., and Stoica, I. (2014). Efficient coflow scheduling with varys. *SIGCOMM Comput. Commun. Rev.*, 44(4):443–454.

[25] Chunji, H. and Fucai, W. (2011). A product-set pricing approach based on price sequence matrix. In *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. IEEE.

[26] Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.

[27] Diestel, R. (2017). *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition.

[28] Ding, W., Qi, W., Wang, J., and Chen, B. (2015). OpenSCaaS: an Open Service Chain as a Service Platform toward the Integration of SDN and NFV. *IEEE Network*, 29(3):30–35.

[29] Dong, M., Li, H., Ota, K., and Xiao, J. (2015a). Rule Caching in SDN-enabled Mobile Access Networks. *IEEE Network*, 29(4):40–45.

[30] Dong, M., Liu, X., Qian, Z., Liu, A., and Wang, T. (2015b). QoE-ensured price competition model for emerging mobile networks. *IEEE Wireless Communications*, 22(4):50–57.

[31] Drutskoy, D., Keller, E., and Rexford, J. (2013). Scalable Network Virtualization in Software-Defined Networks. *IEEE Internet Computing*, 17(2):20–27.

[32] Duan, X., Zhao, C., He, S., Cheng, P., and Zhang, J. (2017). Distributed algorithms to compute walrasian equilibrium in mobile crowdsensing. *IEEE Transactions on Industrial Electronics*, 64(5):4048–4057.

[33] El-Rewini, H., Lewis, T. G., and Ali, H. H. (1994). *Task Scheduling in Parallel and Distributed Systems*. Prentice-Hall, Inc., NJ, USA.

[34] et al., N. P. J. (2017). In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12.

[35] Fang, F., Zhang, H., Cheng, J., and Leung, V. C. M. (2017a). Energy-efficient resource scheduling for noma systems with imperfect channel state information. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–5.

[36] Fang, G., Li, X., and Cai, Z. (2017b). Optimal pricing for service provision in IaaS cloud markets. In *2017 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. IEEE.

[37] Faqiry, M. N. and Das, S. (2018). Double Auction With Hidden User Information: Application to Energy Transaction in Microgrid. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–14.

[38] Fischer, A., Botero, J. F., Beck, M. T., de Meer, H., and Hesselbach, X. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906.

[39] Flach, P. A. and Lachiche, N. (2004). Naive Bayesian Classification of Structured Data. *Mach. Learn.*, 57(3):233–269.

[40] Fu, H., Li, Z., Wu, C., and Chu, X. (2014). Core-Selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing. In *IEEE 7th International Conference on Cloud Computing*, pages 152–159.

[41] Gangadharan, G. R. (2017). Open source solutions for cloud computing. *Computer*, 50(1):66–70.

[42] Gasiunas, V., Dominguez-Sal, D., Acker, R., Avitzur, A., Bronshtein, I., Chen, R., Ginot, E., Martinez-Bazan, N., Müller, M., Nozdrin, A., Ou, W., Pachter, N., Sivov, D., and Levy, E. (2017). Fiber-based Architecture for NFV Cloud Databases. *Proc. VLDB Endow.*, 10(12):1682–1693.

[43] Gember, A., Krishnamurthy, A., John, S. S., Grandl, R., Gao, X., Anand, A., Benson, T., Akella, A., and Sekar, V. (2013). Stratos: A Network-Aware Orchestration Layer for Middleboxes in the Cloud. *CoRR*, abs/1305.0209.

[44] Giotis, K., Kryftis, Y., and Maglaris, V. (2015). Policy-based Orchestration of NFV services in Software-Defined Networks. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5.

[45] Gopinathan, A., Li, Z., and Wu, C. (2011). Strategyproof Auctions for Balancing Social Welfare and Fairness in Secondary Spectrum Markets. In *Proceedings IEEE INFOCOM*, pages 3020–3028.

[46] Gu, S., Li, Z., Wu, C., and Huang, C. (2016). An efficient auction mechanism for service chains in the NFV market. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9.

[47] Halabi, T., Bellaiche, M., and Abusitta, A. (2018). Cloud security up for auction: a dsic online mechanism for secure iaas resource allocation. In *2018 2nd Cyber Security in Networking Conference (CSNet)*, pages 1–8.

[48] Hawilo, H., Shami, A., Mirahmadi, M., and Asal, R. (2014). NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC). *IEEE Network*, 28(6):18–26.

[49] Hong, C.-Y., Caesar, M., and Godfrey, P. B. (2012). Finishing flows quickly with preemptive scheduling. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 127–138, NY, USA.

[50] Hong-tao, B., Li-li, H., Dan-tong, O., Zhan-shan, L., and He, L. (2009). K-means on commodity GPUs with CUDA. In *2009 WRI World Congress on Computer Science and Information Engineering*. IEEE.

[51] Isard, M., Budiu, M., Yu, Y., Birrell, A., and Fetterly, D. (2007). Dryad: Distributed data-parallel programs from sequential building blocks. *SIGOPS Oper. Syst. Rev.*, 41(3):59–72.

[52] Jajoo, A., Gandhi, R., and Hu, Y. C. (2016). Graviton: Twisting space and time to speed-up coflows. In *8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)*, Denver, CO.

[53] Khebbache, S., Hadji, M., and Zeghlache, D. (2017). Scalable and cost-efficient algorithms for vnf chaining and placement problem. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 92–99.

[54] Kumrai, T., Ota, K., Dong, M., Kishigami, J., and Sung, D. K. (2017). Multiobjective optimization in cloud brokering systems for connected internet of things. *IEEE Internet of Things Journal*, 4(2):404–413.

[55] Latifi, M., Rastegarnia, A., Khalili, A., Bazzi, W. M., and Sanei, S. (2020). A self-governed online energy management and trading for smart micro/nano-grids. *IEEE Transactions on Industrial Electronics*, 67(9):7484–7498.

[56] Li, D., Yang, Q., Yu, W., An, D., Yang, X., and Zhao, W. (2017a). A Strategy-proof Privacy-preserving Double Auction Mechanism for Electrical Vehicles Demand Response in Microgrids. In *IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8.

[57] Li, H., Dong, M., and Ota, K. (2015). Radio Access Network Virtualization for the Social Internet of Things. *IEEE Cloud Computing*, 2(6):42–50.

[58] Li, H., Ota, K., and Dong, M. (2018). ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network. *IEEE Wireless Communications*, 25(3):88–93.

[59] Li, H., Ota, K., Dong, M., Vasilakos, A., and Nagano, K. (2017b). Multimedia Processing Pricing Strategy in GPU-accelerated Cloud Computing. *IEEE Transactions on Cloud Computing*, PP(99):1–1.

[60] Li, H., Ota, K., Dong, M., Vasilakos, A. V., and Nagano, K. (2020). Multimedia processing pricing strategy in gpu-accelerated cloud computing. *IEEE Trans. Cloud Comput.*, 8(4):1264–1273.

[61] Li, J., Zhu, Y., Yu, J., Long, C., Xue, G., and Qian, S. (2018). Online auction for iaas clouds: Towards elastic user demands and weighted heterogeneous vms. *IEEE Transactions on Parallel and Distributed Systems*, 29(9):2075–2089.

[62] Li, K., Liu, C., Li, K., and Zomaya, A. Y. (2016a). A framework of price bidding configurations for resource usage in cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 27(8):2168–2181.

[63] Li, L., Ota, K., and Dong, M. (2019). Deepnfv: A lightweight framework for intelligent edge network functions virtualization. *IEEE Network*, 33(1):136–141.

[64] Li, S., Hu, S., Wang, S., Su, L., Abdelzaher, T., Gupta, I., and Pace, R. (2014). Woha: Deadline-aware map-reduce workflow scheduling framework over hadoop clusters. In *IEEE 34th International Conference on Distributed Computing Systems*, pages 93–103.

[65] Li, Y., Phan, L. T. X., and Loo, B. T. (2016b). Network functions virtualization with soft real-time guarantees. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9.

[66] Li, Z., Jiang, C., and Kuang, L. (2021). Double auction mechanism for resource allocation in satellite MEC. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1.

[67] Lin, G., Dong, M., Ota, K., Li, J., Yang, W., and Wu, J. (2019). Security function virtualization based moving target defense of sdn-enabled smart grid. In *2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019*, pages 1–6. IEEE.

[68] Lin, Y., Wang, C., Huang, C., and Lai, Y. (2018). Hierarchical cord for nfv datacenters: Resource allocation with cost-latency tradeoff. *IEEE Network*, 32(5):124–130.

[69] Liu, X., Dong, M., Ota, K., Hung, P., and Liu, A. (2016). Service pricing decision in cyber-physical systems: Insights from game theory. *IEEE Transactions on Services Computing*, 9(2):186–198.

[70] Lyon, A. (2014). Why Are Normal Distributions Normal? *British Journal for the Philosophy of Science*, 65(3):621–649.

[71] Ma, K. and Kumar, P. R. (2021). Incentive compatibility in stochastic dynamic systems. *IEEE Transactions on Automatic Control*, 66(2):651–666.

[72] Ma, L., Wang, X., Wang, X., Wang, L., Shi, Y., and Huang, M. (2021). TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things. *IEEE Transactions on Mobile Computing*, pages 1–1.

[73] Ma, Y., Liang, W., Huang, M., and Guo, S. (2018). Profit maximization of nfv-enabled request admissions in sdns. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.

[74] Matias, J., Garay, J., Toledo, N., Unzilla, J., and Jacob, E. (2015). Toward an SDN-enabled NFV Architecture. *IEEE Communications Magazine*, 53(4):187–193.

[75] Murray, W. and Ng, K.-M. (2010). An algorithm for nonlinear optimization problems with binary variables. *Comput. Optim. Appl.*, 47(2):257–288.

[76] Myerson, R. B. (1981). Optimal Auction Design. *Mathematics of Operations Research*, 6(1):58–73.

[77] Nejad, M. Z., Lu, J., and Behbood, V. (2017). Applying dynamic bayesian tree in property sales price estimation. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 1–6.

[78] Olsen, E. B. (2018). RNS hardware matrix multiplier for high precision neural network acceleration: "RNS TPU". In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

[79] Patterson, D. (2018). 50 years of computer architecture: From the mainframe cpu to the domain-specific tpu and the open risc-v instruction set. In *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pages 27–31.

[80] Peng, X., Ota, K., and Dong, M. (2020). Multiattribute-based double auction toward resource allocation in vehicular fog computing. *IEEE Internet Things J.*, 7(4):3094–3103.

[81] Pier Giuseppe Sessa, Neil Walton, M. K. (July 2017). Exploring the vickrey-clarke-groves mechanism for electricity markets. *IFAC-PapersOnLine*.

[82] Prakash, C., Prashanth, P., Bellur, U., and Kulkarni, P. (2018). Deterministic container resource management in derivative clouds. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 79–89.

[83] Radzi, N. H., Iskandar, K., Abdullah, M. N., Kamaruddin, M. S., Jumaat, S. A., and Aziz, R. (2017). Investigation on cost reflective network pricing and modified cost reflective network pricing methods for transmission service charges. In *2017 2nd International Conference Sustainable and Renewable Energy Engineering (ICSREE)*. IEEE.

[84] Senger, W. and de Góis, L. A. (2017). Homogeneous clusters allocation of computational resources. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–4.

[85] Services, I. A. W. (accessed Janurary, 2016). EC2 Instance Pricing Amazon Web Services (AWS).

[86] Shamani, M. J., Rezaei, S., Seneviratne, A., and Kebriaei, H. (2017). Multi-Path TCP Incomplete Information Repeated Bayesian Game. In *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–6.

[87] Sharma, B., Thulasiram, R. K., Thulasiraman, P., and Buyya, R. (2015). Clabacus: A risk-adjusted cloud resources pricing model using financial option theory. *IEEE Transactions on Cloud Computing*, 3(3):332–344.

[88] Sharma, B., Thulasiram, R. K., Thulasiraman, P., Garg, S. K., and Buyya, R. (2012). Pricing cloud compute commodities: A novel financial economic model. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE.

[89] Shi, W., Zhang, L., Wu, C., Li, Z., and Lau, F. C. M. (2016). An online auction framework for dynamic resource provisioning in cloud computing. *IEEE/ACM Transactions on Networking*, 24(4):2060–2073.

[90] Su, Z., Hui, Y., Luan, T. H., and Guo, S. (2017). Engineering a Game Theoretic Access for Urban Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 66(6):4602–4615.

[91] Sun, J., Wang, X., and Huang, M. (2013). An Intelligent Resource Allocation Mechanism in the Cloud Computing Environment. In *IEEE Third International Conference on Information Science and Technology (ICIST)*, pages 744–750.

[92] Sun, W., Liu, J., Yue, Y., and Zhang, H. (2018). Double Auction-based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*.

[93] Tao, M., Dong, M., Ota, K., and He, Z. (2017). Multiobjective network opportunistic access for group mobility in mobile internet. *IEEE Systems Journal*, PP(99):1–10.

[94] Tao, X., Qi, H., Li, W., Li, K., and Jin, Y. (2015). Pota: Maximizing Profit for Task-Level Scheduling for Data Center Networks. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 1230–1237.

[95] Veitch, P., McGrath, M. J., and Bayon, V. (2015). An Instrumentation and Analytics Framework for Optimal and Robust NFV Deployment. *IEEE Communications Magazine*, 53(2):126–133.

[96] Wang, G. and Ng, T. S. E. (2010). The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. In *Proceedings IEEE INFOCOM*, pages 1–9.

[97] Wang, J., Zhong, H., Yang, Z., Lai, X., Xia, Q., and Kang, C. (2020). Incentive mechanism for clearing energy and reserve markets in multi-area power systems. *IEEE Transactions on Sustainable Energy*, 11(4):2470–2482.

[98] Wei, W., Fan, X., Song, H., Fan, X., and Yang, J. (2018). Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing. *IEEE Transactions on Services Computing*, 11(1):78–89.

[99] Wu, S., Zhou, L., Sun, H., Jin, H., and Shi, X. (2016). Poris: A scheduler for parallel soft real-time applications in virtualized environments. *IEEE Transactions on Parallel and Distributed Systems*, 27(3):841–854.

[100] Xia, M., Shirazipour, M., Zhang, Y., Green, H., and Takacs, A. (2014). Network Function Placement for NFV Chaining in Packet/Optical Data Centers. In *The European Conference on Optical Communication (ECOC)*, pages 1–3.

[101] Xia, M., Shirazipour, M., Zhang, Y., Green, H., and Takacs, A. (2015). Network Function Placement for NFV Chaining in Packet/Optical Datacenters. *Journal of Lightwave Technology*, 33(8):1565–1570.

[102] Xu, Z., Liu, F., Wang, T., and Xu, H. (2016). Demystifying the Energy Efficiency of Network Function Virtualization. In *IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pages 1–10.

[103] Yala, L., Frangoudis, P. A., Lucarelli, G., and Ksentini, A. (2018). Cost and availability aware resource allocation and virtual function placement for CDNaaS provision. *IEEE Transactions on Network and Service Management*, pages 1–1.

[104] Yang, T., Hu, Y., Gursoy, M. C., Schmeink, A., and Mathar, R. (2018). Deep reinforcement learning based resource allocation in low latency edge computing networks. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5.

[105] Yao, J., Lu, Q., and Qi, Z. (2017). Automated resource sharing for virtualized GPU with self-configuration. In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. IEEE.

[106] Yu, R., Xue, G., Kilari, V. T., and Zhang, X. (2015). Network Function Virtualization in the Multi-tenant Cloud. *IEEE Network*, 29(3):42–47.

[107] Yuan, Z. and Hesamzadeh, M. R. (2016). Implementing zonal pricing in distribution network: The concept of pricing equivalence. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE.

[108] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, CA, USA.

[109] Zaman, S. and Grosu, D. (2013). A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds. *IEEE Transactions on Cloud Computing*, 1(2):129–141.

[110] Zats, D., Das, T., Mohan, P., Borthakur, D., and Katz, R. (2012). Detail: Reducing the flow completion time tail in datacenter networks. *SIGCOMM Comput. Commun. Rev.*, 42(4):139–150.

[111] Zhang, D., Chang, Z., Hämäläinen, T., and Yu, F. R. (2017a). Double Auction Based Multi-Flow Transmission in Software-Defined and Virtualized Wireless Networks. *IEEE Transactions on Wireless Communications*, 16(12):8390–8404.

[112] Zhang, H., Jiang, H., Li, B., Liu, F., Vasilakos, A. V., and Liu, J. (2016). A framework for truthful online auctions in cloud computing with heterogeneous user demands. *IEEE Transactions on Computers*, 65(3):805–818.

[113] Zhang, L., Li, Z., and Wu, C. (2014). Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 433–441.

[114] Zhang, X., Huang, Z., Wu, C., Li, Z., and Lau, F. C. M. (2017b). Online Stochastic Buy-Sell Mechanism for VNF Chains in the NFV Market. *IEEE Journal on Selected Areas in Communications*, 35(2):392–406.

[115] Zhao, X., Yao, J., Gao, P., and Guan, H. (2018). Efficient sharing and fine-grained scheduling of virtualized GPU resources. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE.

[116] Zhao, Y., Chen, K., Bai, W., Yu, M., Tian, C., Geng, Y., Zhang, Y., Li, D., and Wang, S. (2015). Rapier: Integrating routing and scheduling for coflow-aware data center networks. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 424–432.

[117] Zhou, Z., Liu, F., Chen, S., and Li, Z. (2020). A truthful and efficient incentive mechanism for demand response in green datacenters. *IEEE Transactions on Parallel and Distributed Systems*, 31(1):1–15.

Publications

Journals

1. Wuyunzhaola Borjigin, Kaoru Ota, Mianxiong Dong, "In Broker We Trust: A Double-auction Approach for Resource Allocation in NFV Markets," IEEE Transactions on Network and Service Management (TNSM), vol. 15, no. 4, pp. 1322-1333, December 2018.

2. Wuyunzhaola Borjigin, Kaoru Ota, Mianxiong Dong, "Walrasian Auction Mechanism for Tree Valuation Service in NFV Market," IEEE Transactions on Computational Social Systems(TCSS), 2021, in press.

Proceeding of International Conference

1. Wuyunzhaola Borjigin, Kaoru Ota, Mianxiong Dong, "Dealer: An Efficient Pricing Strategy for Deep-Learning-as-a-Service", IEEE ICC, 2019. (Student Travel Grant Award)

2. Wuyunzhaola Borjigin, Kaoru Ota, Mianxiong Dong, "Time-saving First: Coflow Scheduling for Datacenter Networks", in Proceedings of IEEE 86th Vehicular Technology Conference, 2017. (IEEE VTS Tokyo Chapter 2017 Student Paper Award)