Virtualized Network Function Provisioning in Stochastic Cloud Environment

Yanghao Xie, Binbin Wang, Sheng Wang, Long Luo
School of Information and Communication Engineering
University of Electronic Science and Technology of China, Chengdu, P. R. China
Email: yanghao.xie@std.uestc.edu.cn, wbinbin@std.uestc.edu.cn, wsh_keylab@uestc.edu.cn, longluo.uestc@gmail.com

***Abstract*—Network Function Virtualization (NFV) provides a new paradigm for provisioning network service where network functions are deployed as Virtual Network Functions (VNFs). Due to the advantages of NFV, many Network Function Virtualization Providers (NFVPs) offer their NFV services by deploying VNFs with purchased cloud resources in cloud environment to save the provisioning expense. However, existing VNF provisioning solutions ignore the influences of the dynamics of cloud environment, which may lead to over-provisioning and high deployment expense. In this paper, we study the problem of how should the NFVPs purchase cloud resources to provide NFV services for customers in order to minimize the expense of NFVPs, considering the dynamics of the system. We first abstract the system model of this problem and formulate it as a stochastic optimization programming problem. Then, we present our VIrtual Network functiOn proviSioning (VINOS) approach that can efficiently solve the stochastic optimization programming with a rolling horizon procedure. In particular, it first leverages Long Short Term Memory (LSTM) networks to predict future exogenous information and then optimally solves a deterministic problem over short horizon. We conduct extensive numerical experiments to evaluate the proposed approach. The experiment results suggest that our approach achieves total cost of 1.2 times offline optimum, and outperforms the benchmark algorithm by 8%, averagely.**

***Index Terms*—Network Function Virtualization, Virtualized Network Function Provisioning, stochastic optimization, long short term memory networks**

I. Introduction

NFV aims to build more dynamic and service-aware networks while reducing Capital Expense (CAPEX) & Operating Expense (OPEX) and improving service agility. It leverages standard virtualization technologies to decouple physical network devices from the functions that run on them. In this way, Telecommunication Service Providers (TSPs) can design, deploy, and manage network services in a new paradigm where services can be decomposed into sequences of VNFs, which are called Service Function Chainings (SFCs), and these VNFs can then be instantiated in software running on standard physical servers at different network locations without purchasing new hardware [1–3].

Urging to obtain the benefits of NFV, many initiatives start to investigate the possibility of deploying VNFs in the cloud [4, 5]. Within this marketplace, NFVPs can purchase public cloud resources to instantiate VNFs and construct SFCs, in order to serve the service requests of customers on demand.

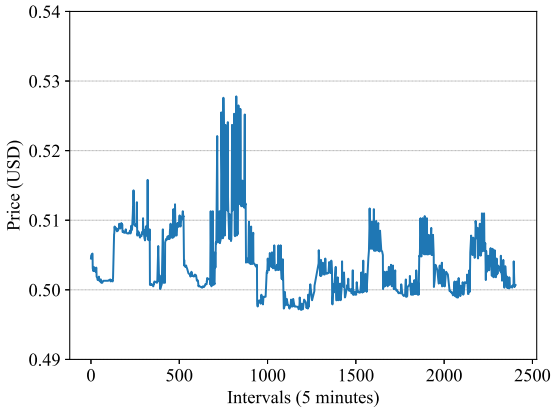

Fig. 1. Spot price of an instance of class `c3.8xlayge` VM [7].

In this paper, we investigate the problem of how NFVPs should purchase cloud resource for VNF provisioning for the sake of minimizing their expense. Nevertheless, there are several dynamics that make VNF provisioning very challenging. (i) The flow rates of service requests may change over time. This flow fluctuation is becoming more and more significant with the emergence of new network services. For example, in Internet of Things (IoT) applications, the flow is always time-varying with the asynchronous activation and silence, node failures and mobility of sensors and actuators [6]. (ii) The prices of cloud resources may vary over time, too. For example, Fig. 1 presents the spot price of `c3.8xlarge` type Virtual Machine (VM) instance at west of United States (US) of Amazon EC2, from which we can see that the price changes dramatically over time [7]. The fluctuations of service rates and cloud prices introduce uncertainties into decision-making process when NFVPs purchasing cloud resource for provisioning VNFs. These fluctuations make the decision-making process of NFVPs more difficult since decisions made current without knowing future information may turns out to be sub-optimal in the long run. However, by elaborately panning the decision-making process, we could also encounter expense as little as possible. Therefore, there is an urging demand for effective methods to deal with this dynamics in VNF provisioning system.

However, most existing proposals tend to deal with static

problems. These proposals are unable to deal with the dynamic problems, since these methods assume that the overall information is known when making decisions [8, 9]. There are some works try to deal with the online version of the problem where the service requests arrive to the system gradually. Nevertheless, there methods ignore the uncertainties introduced by the NFV system itself [10, 11]. For example, the price and flow rate fluctuations we consider in this paper. We will discuss more details about the drawbacks of previous works in Section II.

In this paper, we study the problem of optimizing the strategy for NFVPs to purchase cloud resource, such that their expense are minimized. We investigate the system model of VNF provisioning and formulate it as a stochastic optimization problem. However, we cannot solve this stochastic programming easily due to the explosion of the problem size over an infinite or even a long horizon. In order to overcome this challenge, we adopt *rolling horizon procedure* which is a widely used and efficient method in *operational research* community. This method first build a deterministic programming from origin stochastic programming leveraging predicted future exogenous information and then solve this deterministic problem in a short horizon. In particular, We obtain the predicted information using LSTM network which is well-known for predicting time series data, and solve the deterministic problem using CPLEX. Finally, comprehensive numerical simulations are conducted to evaluate the proposed approach. The experiment results show that the proposed algorithm achieves near-optimal performance compared to offline optimal and has superior performance over benchmark method.

The rest of this paper is organized as follows. In Section II, we discuss more details about existing works. We present the system model and formulate the problem in Section III. In Section IV, we present the solution adopting rolling horizon procedure. We evaluate the performances of proposed method in Section V. We conclude the paper in Section VI.

II. Related Works

Due to so many advantages of NFV, there are a vast amounts of proposals that study the problem of VNF provisioning.

Some literature investigates optimal VNF chain placement over objectives like throughput maximization, cost minimization, etc. Sallam and Ji [8] study the problem of maximizing total processed traffic under budget constraint and capacity constraint. By relaxing the requirement of fully processed flows and granting partially processed flows, the authors propose two performance guaranteed algorithms for the original problem. Huang et al. [9] consider how to maximize throughput of delay-sensitive service requests by leveraging horizontal scaling and vertical scaling, and they devise an efficient algorithm by reducing the problem to minimum-weight feedback arc set problem and the generalized assignment problem. Hawilo et al. [12] examine the problem of VNF placement in small and large scale Data Center (DC) networks, and the authors provide a novel Mixed Integer Linear Programming (MILP) optimization model and a novel heuristic solution. Similarly, Gu et al. [13] investigate the problem of VNF deployment and flow scheduling in geo-distributed DCs, considering minimizing deployment and communication costs. They formulate the problem into MILP and propose a relaxation-based algorithm. However, The above literature considers the static VNF placement problem, which neglects the dynamic characteristics of NFV system.

Many proposals study the online version of the VNF provisioning problem where service requests arrive to NFV system gradually. Chen et al. [10] propose a fully decentralized approach for online VNF placement problem by using Lyapunov optimization techniques. Gu et al. [14] investigate a fairness-aware flow scheduling problem for network utility maximization in NFV environment. Similarly, by using Lyapunov optimization framework, the authors propose a low complexity online distributed algorithm. Guo et al. [11] propose provable algorithm by combining techniques from multiplicative weight update and primal-dual update paradigms. Nevertheless, these works only consider the dynamics introduced by gradually arrived service requests and ignore the dynamics of NFV system itself. Cheng et al. [6] study the resource allocation of NFV from stochastic perspective. They consider the dynamics that are introduced by fluctuation of traffic rate and available amounts of wireless resources at access nodes. They formulate the problem as a stochastic optimization problem and propose a distributed algorithm with two-level decomposition by exploiting the hierarchical decision structures in the problem. However, this work considers a totally different system compared to ours and the proposed method cannot be adopted to our problem easily.

III. System Model and Formulation

In this section, we present the abstract system model with formal mathematical notations and then formulate the problem as a stochastic optimization problem. Table I summarizes the major notations used in this paper.

A. Clouds, VMs, and VNFs

The cloud provider, like Amazon, provides many kinds of VMs with different specifications that have different capacities of resources, e.g., Central Processing Unit (CPU), Random Access Memory (RAM), and storage. In this paper, without lost generality, we only consider one type of bottleneck resource, i.e., CPU. We assume the system is time slotted, and the NFVP offers N types of VNFs for customers in T time slots. In general, different types of VNFs have different resource requirements. For example, computation-consuming VNFs like Deep Packet Inspections (DPIs) require more CPU resource, hence, VMs with more CPU cores are supposed to deploy DPIs, so `c4.2xlarge` VMs with 8 vCPUs and 15G memory may be a proper choice for instantiating DPIs. To sum up, different types of VMs are needed to accomplish VNF provisioning. Therefore, for each type of VNFs, NFVPs should purchase VMs with different specifications. In addition,

TABLE I
SUMMARY OF THE NOTATIONS USE IN THE FORMULATION

	Cloud and VNF
c^{nr}	Price of *reserved* VMs used to instantiate type-n VNF at t.
c^{no}	Price of *on-demand* VMs used to instantiate type-n VNF at t.
c_t^{ns}	Price of *spot* VMs used to instantiate type-n VNF at t.
τ^r	Duration of *reserved* VMs.
τ^o	Duration of *on-demand* VMs.
τ^s	Duration of *spot* VMs.
C_n	Flow processing capacity of type-n VNF.
	Service Requests
F	The set of service requests.
b_f	Initial flow rate of f, also denoted by b_f^0.
M_f	The set of VNFs request f should traverse.
η_m^n	Equals 1 if the type of VNF m is n.
d_t^n	Number of type-n VNF at t.
	Decision Variables
x_t^{nr}	Number of *reserved* VMs bought for deploying type-n VNF at t.
x_t^{no}	Number of *on-demand* VMs bought for deploying type-n VNF at t.
x_t^{ns}	Number of *spot* VMs bought for deploying type-n VNF at t.
	System
T	Timespan.

different type of VNFs also have different processing capacity in terms of bandwidth, and let C_n to represent the flow processing capacity of type-n VNF.

Each VM instance can be purchased as *reserved instance*, *on-demand instance*, and *spot instance*, this is the typical pricing method of Amazon EC2. In general, reserved instances are purchased for a longer duration with relatively lower prices, we denote the price and duration of a reserved instance used for deploy type-n VNF at time t as c^{nr} and τ^r, respectively. However, on-demand instances and spot instances can be bought for a shorter duration. The prices and durations of on-demand instances and spot instances that are used for deploying type-n VNF at time t are represented by c^{no}, c_t^{ns}, τ^o and τ^s, respectively. Particularly, the prices of spot instances vary dramatically, as presented in Fig. 1. However, the prices of reserved instances and on-demand instances tend to be relatively steady, comparing with the prices of spot instances. Therefore, in this paper, we handle the prices of spot instances as stochastic variables, where the spot prices may vary over time and future prices are not know when NFVPs make purchase decisions.

B. Service Requests

The set of service requests is denoted by F. The sequence of VNFs that the flow of request f should traverse is M_f, and for each VNF $m \in M_f$, let η_m^n indicate whether the type of VNF m is n. In addition, for each request $f \in F$, the initial flow rate is represented by $b_f(b_f^0)$. As mentioned in Section I, in this paper, we consider that the flow rates may fluctuate over time. Besides the flow rates change due to traffic fluctuation, the flow rates may also change after the flows are processed by some VNFs. For example, Intrusion Detection Systems (IDSs) may drop packets that violate security polices; IPSec/SSL VPN and media gateways can increase (decrease) packet size for encapsulation (decapsulation) [15]. We use δ_n to denote the traffic change ratio after the flow is processed by type-n VNF. Note that we also use δm to denote the change ratio for VNF m with little notation abuse, since we can get the value of δ_m easily by get the type VNF m. For example, the rate of flow of service request f after being processed by the first VNF is calculated $b_f^1 = b_f^0 \cdot \delta_0$.

C. Decision Variables and Formulation

In each time slot t, the NFVP observes the revealed exogenous information, i.e., the price fluctuation and flow rate fluctuation for every service request f. Then together with the inventory of VM instances bought before, the NFVP needs to decide how many VM instances should purchase for each specification, in order to serve current flow rate or reserve for future usage. In particular, the NFVP makes the following decisions in each time slot t: (i) the number of reserved instances to buy for each specification, denoted by x_t^{nr}; (ii) the number of on-demand instances to buy for each specification, denoted by x_t^{no}; (iii) the number of spot instances to buy for each specification, denoted by x_t^{ns}. Without lost generality, we assume the NFVP intents to minimize the overall cost for buying VM instances over T time slots.

To sum up, the cost minimization problem can be formulated as follows stochastic programming:

$$\min \mathbb{E} \sum_{t \in T} \sum_{n \in N} (c^{no}\tau^o x_t^{no} + c^{nr}\tau^r x_t^{nr} + c_t^{ns}\tau^s x_t^{ns}) \tag{1}$$

subject to

$$x_t^{no} + x_t^{nr} + x_t^{ns} + r_{t-1}^n \geq d_t^n, \forall t \in T, \forall n \in N \tag{2}$$

$$\sum_{f \in F_t} \sum_{m \in M_f} b_{ft}^m \eta_m^n \leq d_t^n C_n, \forall t \in T, \forall n \in N \tag{3}$$

$$b_{ft}^m = b_{ft}^{m-1} \delta_{m-1}, \forall f \in F, \forall m \in M_f \setminus \{0\} \tag{4}$$

$$r_{t-1}^n = \sum_{\tau=1}^{\tau=\tau^o} x_{t-\tau}^{no} + \sum_{\tau=1}^{\tau=\tau^r} x_{t-\tau}^{nr} + \sum_{\tau=1}^{\tau=\tau^s} x_{t-\tau}^{ns}, \forall t \in T, \forall n \in N \tag{5}$$

$$x_t^{no}, x_t^{nr}, x_t^{ns} \in \mathbb{Z}_{\geq 0}, \forall t \in T, \forall n \in N \tag{6}$$

Formula (1) is the total cost of buying VM instances, which is the summation of money for purchasing VMs of each specification using three pricing methods. Constraint (2) ensures that the required number of VMs for each type of VNFs are satisfied. Constraint (3) guarantees that the VNF capacities are sufficient for processing all the flows of all service requests. Constraint (4) denotes the flow rate changes after the flows traverse VNFs. Since the initial flow rates are know, therefore, we remove the 0-th VNF from M_f denoting by $M_f \setminus \{0\}$. Constraint (5) calculates the inventory of VM instances. (6) ensures that the decision variables are nonnegative integers.

IV. THE VNF PROVISIONING ALGORITHM

One can easily obtain the history data of Amazon spot prices from Amazon EC2 console [7]. Although the spot prices fluctuate dramatically over time, it is possible to predict the prices with tolerable inaccuracy, especially when machine learning technologies have made great progress in recent years.

Algorithm 1: VINOS: VIrtual Network functiOn proviSioning

input : History data of prices, flow rates and purchase decisions; exogenous information: c_t^{ns} and b_{ft}

output: Purchase decisions at time t: x_t^{nr}, x_t^{no} and x_t^{ns}

1 Predict $c_{t'}^{ns}$ and $b_{ft'}$, $\forall t^{'} \in t+1, \cdots, t+H$, $\forall f \in F$, $\forall n \in N$ using LSTM network

2 Construct deterministic problem $\mathcal{P}$ based on predicted data over t to $t+H$

3 Solve problem $\mathcal{P}$ using CPLEX solver; Make decisions according to the optimal solution of problem $\mathcal{P}$ at time t

4 Collect price, flow rates, and decisions at time t for future usage

In particular, LSTM networks are confirmed to be very good at predicting time series data. Similarly, we can also investigate the possibility of predict flow rates of service requests based on collected history data. Therefore, it is workable that the NFVP makes decisions based on not only current state and observed exogenous information, but also prediction information obtained by leveraging advanced time series data prediction techniques.

Along this line of thinking, in order to solve the previous stochastic optimization problem, we can resort to *rolling horizon procedure (model predictive control)* which is a well-known approach in stochastic optimization society, especially in operational research community. This method is a natural approximation strategy that solves the original stochastic problem by repeatedly solve a deterministic problem over a shorter horizon. A popular method in practice for building the deterministic model is to forecast future exogenous information over a H-period horizon. In particular, at time t, we can solve the problem optimally over horizon from t to $t+H$, and we implement the decisions x_t^{no}, x_t^{nr} and x_t^{ns} for slot t. Then we repeat the process by solving the problem over horizon $t+1$ to $t+H+1$, and so on [16].

To sum up previous analysis, the proposed solution adopts rolling horizon procedure which leverages LSTM networks to predict future exogenous information and CPLEX to solve the deterministic problems. The detail procedures are presented in Algorithm 1. In particular, We name this algorithm as VIrtual Network functiOn proviSioning (VINOS).

V. Trace-Driven Evaluation

In this section, we evaluate the performance of the propose method. We first measure the prediction accuracy of the LSTM networks, and then we evaluate the performance of VINOS under different settings of the timespan and the number of service requests.

A. Simulation Settings

We conduct a trace-driven evaluation using real data. We construct service requests based on Abilene dataset [17], which is collected from and educational backbone network in North America. In particular, we set service flow rates using flow rates from this dataset. In addition, there are total 5 types of VNFs in our simulation, and we randomly generate a sequence of 3 VNFs for each service request. The price data is collected from Amazon. We compare VINOS with offline optimum calculated by CPLEX and a heuristic algorithm, and the two methods are indicated as `CPLEX` and `Heuristic` in the legend, respectively. In particular, the heuristic algorithm makes greedy purchase decisions at each interval t such that the remaining VMs plus purchased VMs at t exactly equals the demand at t, and the algorithm chooses the cheapest pricing scheme at each slot.

TABLE II
Main parameters of the experiments.

Length of training data	11068 for prices & 1679 for flow rates
Length of test data	382
Number of requests	$\{50, 60, 70, 80, 90, 100\}$
Timespan	$\{10, 15, 20, 25, 30, 35, 40, 45\}$
Type of VNFs	5
Number of VNFs each request	3

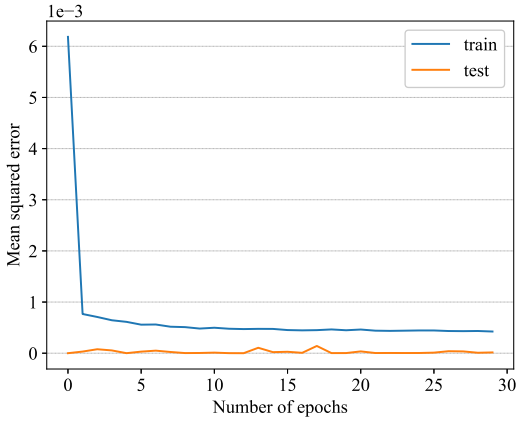

Fig. 2. Loss of train and test datasets when training LSTM network for a price trace.

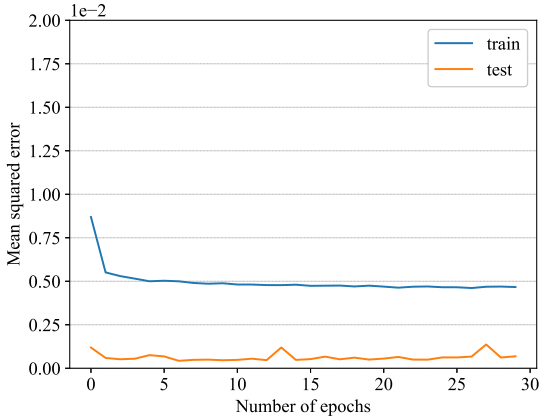

Fig. 3. Loss of train and test datasets when training LSTM network for a flow rate trace.

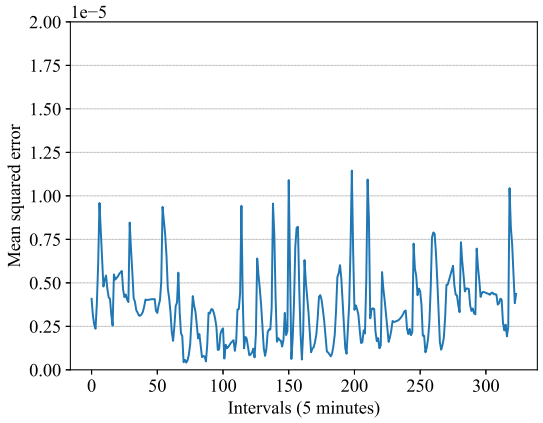

Fig. 4. Prediction error of a spot price trace.

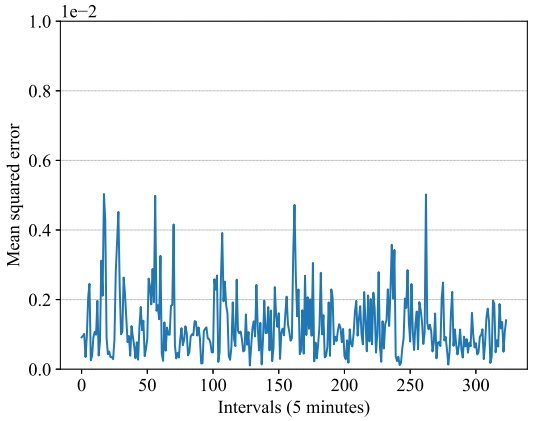

Fig. 5. Prediction error of a flow rate trace.

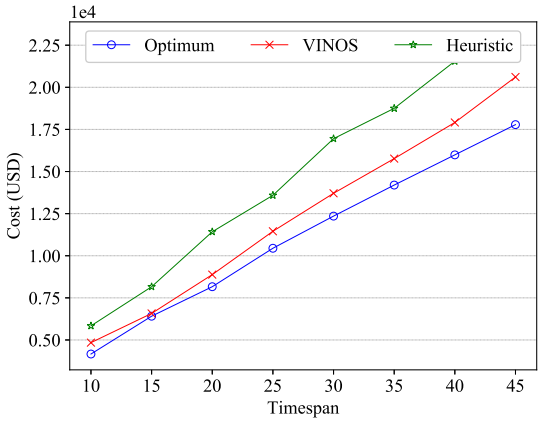

Fig. 6. Cost over different timespan.

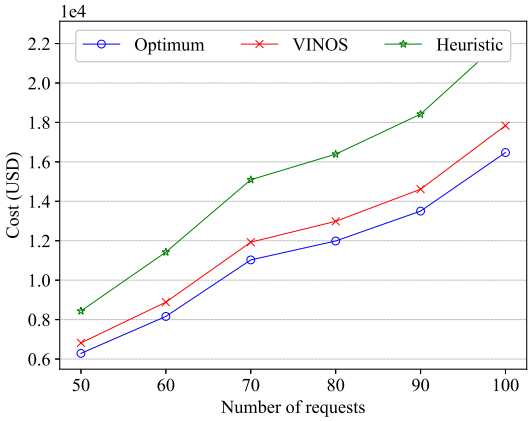

Fig. 7. Cost over different numbers of requests.

B. Accuracy of Prediction

The neural network is composed of three layers. The first layer is LSTM network with 200 units and the activation function is Rectified Lined Unit (RELU), the second layer is fully connected layer composed of 100 nodes and the activation function is also RELU, the last layer (output layer) is fully connected layer with 5 nodes depending on the output length. We use Mean Squared Error (MSE) as the loss function and adopt Adaptive Moment Estimation (Adam) as the optimizer in our training process.

We picked the last 328 pieces of the dataset as test set, and the rest data were used as training set. In our model, we used the latest 5 time slots historical datas to predict the next five data, and trained the model for total 30 epochs. Finally, we applied the model to test dataset. In the following experiment, we take the test set as our dataset in real physical system we performed.

We now analyze the accuracy of our predictions for prices and flow rates. Fig. 2 and 3 show the MSE of training and testing dataset in training process for two specific price and flow traces over 30 epochs, respectively. We can see that with the epochs increasing, the MSE (loss value) decreases and tends to be stable.

Fig. 4 and 5 show MSE for real and predicted values of two specific price and flow traces previously mentioned, respectively. The MSE is different from the above one, and it is calculated by using the prediction values and ground truth. For each moment, we predict data for future five days and calculate MSE. The MSE is relatively small. In particular, in Fig. 4 MSE values range from 0 to 0.000015, in Fig. 5 MSE values are between 0 and 0.006, which illustrates the predictions have relatively high accuracy. To sum up, our predictions of the prices of virtual machines and the flow rates are relatively accurate.

C. The Impact of Different Timespan

In this experiment, we consider the performances of the three methods over different timespan. We set number of requests to 60 and timespan from 10 to 45, then we observe the total cost as the timespan increases. As Fig. 6 shows, we can draw a conclusion that VINOS achieves near-optimal solution, compared with optimum solution which is obtained by CPLEX. In addition, VINOS shows better performance than heuristic algorithm. On the one hand, VINOS achieves

objective of 1.2 $\times$ OPT, where OPT is the offline optimal objective value obtained by CPLEX and performs 8.6% better than the heuristic algorithm on average. On the other hand, as timespan increases, the cost generated by VINOS is 20% to 25% greater than what CPLEX obtained, and is 4.2% to 14.7% smaller than what heuristic algorithm obtained.

D. The Impact of Different Numbers of Service Requests

In the following experiment, we investigate the performances over different numbers of service requests. We set timespan to 20 and the number of requests from 50 to 100, then we observe the total cost as the number of requests increase. As is shown in Fig. 7, it is also easily to see that VINOS achieves good performance compared with CPLEX and outperforms the heuristic algorithm. In particular, we can see that VINOS achieves objective of 1.2 $\times$ OPT, moreover, VINOS performs 8.0% better than the heuristic algorithm on average. As the number of requests increases, VINOS accomplishes steady performance, ranging from $25.7\%\times$OPT to $27.2\%\times$OPT, and achieves smaller cost than the heuristic algorithm, ranging from 5.4% to 11.3%.

According to previous analysis, we can conclude that VINOS accomplishes superior performances, this is because VINOS takes account future exogenous information when it makes decisions, while, the heuristic algorithm can only make greedy decisions based on history and current information.

VI. Conclusions

In this paper, we have investigated the VNF provisioning problem in stochastic cloud environment. We formulate this problem as a stochastic optimization problem, and we propose VINOS that can deal with the uncertainties introduced by the fluctuation of cloud price and flow rate. VINOS follows the paradigm of rolling horizontal procedure: first constructing a deterministic problem over a short horizon with the predictions of LSTM networks and then optimally solving this problem. Our extensive trace-driven experimental results show that VINOS achieves the near-optimal performance and outperforms the benchmark solution.

For future work, more considerations will be explored in terms of predicting exogenous information and leveraging this information to design high performance algorithms. For example, design algorithms based on distribution of exogenous information. Moreover, it is also interesting to design more efficient algorithm to solve the deterministic problem. In addition, we would like investigate other dynamic characteristics ignored by existing works, such as substrate topology changes.

Acknowledgment

This work is partially supported by NSFC Fund (61671130, 61301153, 612711656, 61671124), 973 Program (2013CB329103), Program for Changjiang Scholars and Innovative Research Team (PCSIRT) in University, the 111 Project (B14039), and Project on Public Safety Risk Prevention and Control and Emergency Technical Equipment (2018YFC0831002).

References

[1] Write Paper, "Network Functions Virtulisation: An Introduction, Benifits, Enablers, Challenges & Call for Action. Issue 1." in *2012, SDN and OpenFlow World Congress*, Oct. 2012.

[2] ETSI NFV ISG, "Network Functions Virtulisation (NFV); Management and Orchestration," *ETSI GS NFV-MAN 001 V1.1.1*, Dec. 2014.

[3] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges," *Communications Surveys & Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[4] H. Tang, D. Zhou, and D. Chen, "Dynamic Network Function Instance Scaling Based on Traffic Forecasting and VNF Placement in Operator Data Centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 3, pp. 530–543, Mar. 2019.

[5] ETSI NFV ISG, "Network Functions Virtualisation (NFV) Release 3; Virtualised Network Function; Specification of the Classification of Cloud Native VNF implementations," Tech. Rep., Oct. 2018.

[6] X. Cheng, Y. Wu, G. Min, and A. Y. Zomaya, "Network Function Virtualization in Dynamic Networks: A Stochastic Perspective," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2018.

[7] "Amazon EC2 spot instance pricing history." [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html

[8] G. Sallam and B. Ji, "Joint Placement and Allocation of Virtual Network Functions with Budget and Capacity Constraints," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 523–531.

[9] M. Huang, W. Liang, Y. Ma, and S. Guo, "Maximizing Throughput of Delay-Sensitive NFV-Enabled Request Admissions via Virtualized Network Function Placement," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[10] X. Chen, W. Ni, I. B. Collings, X. Wang, and S. Xu, "Automated Function Placement and Online Optimization of Network Functions Virtualization," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1225–1237, Feb. 2019.

[11] L. Guo, J. Pang, and A. Walid, "Joint Placement and Routing of Network Function Chains in Data Centers," in *INFOCOM 2018-IEEE International Conference on Computer Communications*, 2018, pp. 1–9.

[12] H. Hawilo, M. Jammal, and A. Shami, "Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 643–655, Mar. 2019.

[13] L. Gu, X. Chen, H. Jin, and F. Lu, "VNF Deployment and Flow Scheduling in Geo-Distributed Data Centers," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[14] L. Gu, D. Zeng, S. Tao, S. Guo, H. Jin, A. Y. Zomaya, and W. Zhuang, "Fairness-Aware Dynamic Rate Control and Flow Scheduling for Network Utility Maximization in Network Service Chain," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1059–1071, May 2019.

[15] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, "Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[16] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, 2007, vol. 703.

[17] "Abilene dataset." [Online]. Available: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/