Resource Optimization and Traffic-aware VNF placement in NFV-enabled Networks

Yi Yue, Bo Cheng, Xuan Liu, Meng Wang, Biyi Li
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
{yueyi, chengbo, liuxuan0527, mengwang, lbyi0402}@bupt.edu.cn

***Abstract*—Although network function virtualization (NFV) is a promising approach for providing elastic network functions, it faces several challenges. A critical but difficult issue for the service and network providers is deciding where to instantiate a list of virtual network functions (VNFs), namely VNF placement problem. In this paper, we investigate the VNF placement, for the purpose of resource and network traffic consumption minimization. Moreover, we consider the arrival rates of users' requests for different types of service function chains (SFCs). This allows the placement scheme to adapt to users' time-varying requests and improve the network resource utilization. Then we formulate the VNF placement problem as a jointly constrained optimization problem. Afterwards, we propose an approach called joint optimization resource and traffic consumption (J-ORTC) with enhanced biogeography-based (EBBO) optimization algorithm to resolve the VNF placement problem. Finally, the evaluation results show the effectiveness of J-ORTC approach and performance advancement over the benchmarks.**

***Index Terms*—Network Function Virtualization, Service Function Chain, Resource allocation, Virtual Network Function**

I. Introduction

Network function virtualization (NFV), an innovative network architecture paradigm, has emerged as a promising network architecture. Encouraged by the benefits of NFV, more and more enterprises and network operators exploit NFV technology to reduce the expenditure of network function management and infrastructure construction [1].

We use SFC requests (SFCRs) to describe users' service requests. Correspondingly, for the elements included in a specific SFCR, we treat them as VNF requests (VNFRs). When hundreds of SFCRs arrive in the cloud, the cloud service providers face a critical but difficult problem which refers to the VNF placement problem. Particularly, the problem is how to deploy a series of SFCs, which separately contains one or more VNFs, on physical nodes in the cloud. Generally, an effective scheme of VNF placement requires meeting one or more optimization objectives, such as maximizing the network throughput or minimizing the resource consumption [2]. A good placement scheme can not only increase the network resource utilization, but also provide more profits for service providers. Generally, VNF placement problem is tackled as an integer linear programming problem, with heuristic approaches to solve it, such as [3], [4] and [5].

Different from most of existing works, the VNF instances are shareable in our consideration, which implies a VNF instance can serve multiple tenants at the same time [5]. In this paper, we consider the VNFRs in SFCRs as the tenants. To improve the utilization of network resource in VNF placement, two optimization objectives are considered:

1) *Resource consumption*. The resource consumption consists of two parts. One part is the consumption for processing users' service requests. The other part is the basic resource overhead (BRO) of instantiating the basic functionality of a VNF, such as the consumption to run an operation system (OS) and related libraries of one VNF [6]. In this paper, the resource consumption is concretized as CPU consumption. Considering the shareable VNFs, by placing VNFRs that require the same type of VNF instance together, fewer VNF instances are needed and the BRO subsequently decreases, leading to less resource consumption.
2) *Network traffic consumption*. The traffic consumption occurs when a VNF communicates with another VNF via the virtual connection between them. Specifically, the network traffic consumption comes from two parts. One part is the total traffic capacity assigned to the virtual links, which mapped on an active physical link. The other part is the overhead of operating that active physical link based on the hop distance [7]. If all the VNFs in an SFC are placed on the same node, the traffic loads among these VNFs are limited in the node, so the traffic consumption can be avoided.

Afterwards, we take a less-considered factor into consideration, which is the arrival rate of users' SFCRs for using a specific SFC. Using this factor, we can derive a more precise resource consumption required by VNFs and use it to guide the VNF placement. Based on the more precise resource consumption, the scheme of VNF placement can better adapt to users' time-varying SFCRs and save network resource.

However, objective 1) and 2) are conflicting. Since minimizing the resource consumption can increase the number of active physical links that leads to high network traffic consumption. Meanwhile, optimizing the traffic consumption needs more VNF instances, then the resource consumption increases subsequently. Therefore, we devise an approach for joint optimization resource and traffic consumption (J-ORTC) of the VNF placement with enhanced biogeography-based

978-1-7281-2583-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICPADS47876.2019.00030

optimization [8] (EBBO) algorithm.

Our key contributions are summarized as follows:

1) We formulate the resource-efficient and traffic-aware VNF placement problem as a jointly constrained optimization problem, considering the time-varying arrival rates of users' SFCRs. Then we introduce J-ORTC approach which aims to minimize the total incurred resource and network traffic consumption in the system.
2) To derive the solution for J-ORTC problem, we propose an algorithm which is based on biogeography-based optimization (BBO) algorithm. Since the original BBO is expensive to directly solve J-ORTC, we carefully seek a penalty function that reduce the original BBO's state space of feasible solutions. The combination of the penalty function and BBO is EBBO algorithm, which efficiently reduces the convergence time.
3) To obtain more precise resource consumption of VNFs, we employ follow-the-regularized-leader (FTRL) [9], which is an online learning approach, to derive the arrival rates of users' SFCRs from the trace data in Facebook's datacenter [10]. We then compare J-ORTC approach with other approaches in several case studies. Our simulation results demonstrate that J-ORTC approach provides near-optimal solutions and effectively reduces the total incurred consumption in the system. Furthermore, J-ORTC achieves lower total consumption than the benchmarks.

The rest of this paper is organized as follows. Section II discusses about the related works. Section III describes the system model and problem statement. Our solution EBBO is introduced in Section IV. Afterwards, the performance evaluation of J-ORTC is represented in Section V. Finally, we make a conclusion in Section VI.

II. Related works

Recently, NFV and the related VNF placement have received plenty of attentions from academia and industry. In [3], they proposed a near optimal solution for actual placement of VNFs using the physical network and gave a theoretical proof. Kuo et al. [11] studied the VNF placement problem considering the relationship between server usage and link. In addition, the authors conducted a mathematical program analysis and presented a heuristic scheme. The authors in [7] employed the Markov approximation-based algorithm to optimize the VNFs placement, with the goal of minimizing the operational and traffic cost for NFV chaining. Rankothge et al. [12] proposed genetic algorithms to optimize the VNFs resource allocation in cloud datacenter. The authors in [5] considered the shareable VNFs and the authors proposed a two-stage heuristic solution to optimize the used nodes. However, most of the existing works on VNF placement problem pay little attention to the shareable VNFs, ignoring the fact that the shareable VNFs can improve the utilization of network resource.

Another important optimization objective of VNF placement is the network traffic consumption. Works in [13] and [7] considered solely the network traffic consumption and neglected the resource consumption. The authors in [14] only focused on traffic changing effects and dependencies between VNFs. In [15], the author analyzed the network traffic in detail, but they did not elaborate this factor in the objective function.

Although few works consider the arrival rate of users' SFCs requests in VNF placement, some researches on VNF provisioning have taken into account the time-varying users' SFC request. In [16], the authors designed online algorithms with the goal of minimizing the VNF provisioning cost, which considered both single SFC and multiple SFCs. Mijumbi et al. [17] designed a predictive algorithm based on the graph neural network to optimize resource allocation for each VNF component. Zhang et al. [18] designed a proactive online algorithm to reduce resource consumption, but the bandwidth consumption problem is ignored.

TABLE I
Notations

Parameters	Descriptions
SFC related	
N	set of total physical nodes in network.
R	set of total SFCRs, $r \in R$ is an SFCR.
P_r	set of total VNFRs in SFCR r, $p \in P$ is a VNFR.
V	types number of VNFs, $v \in \{0, 1, ..., V-1\}$.
T	a time interval we analyze.
Φ	number of samples in a time interval T.
t_ϕ	the ϕth time point in the time interval T.
Optimization	
x^r_{pn}	whether VNFR p in SFCR r is hosted on physical node n.
y^n_v	whether a type-v VNF instance is placed on node n.
k^r_{pv}	whether VNFR p in SFCR r demands a type-v VNF.
$arr_r(t_\phi, t_{\phi+1})$	the arrival rate of users' requests for SFC r in sample time .
τ_v	resource consumption required by a type-v VNF to support its processing capacity
$prc_p(t_\phi, t_{\phi+1})$	resource consumption required by VNFR p to process users' requests.
bro_v	basic resource overhead required by a type-v VNF instance.
$trf^r_{pp'}$	traffic rate required by VNFRs p and p' in SFCR r.
$l_{nn'}$	hop distance between nodes n and n'.
Q^{res}_n	resource capacity of node n.
$Q^{link}_{nn'}$	link capacity of physical link (n, n').
α	cost of per unit resource consumption.
β	cost of per unit traffic capacity.
I_{total}	total resource consumption incurred in the system.
G_{total}	total traffic consumption incurred in the system.
F_{total}	objective function of the J-ORTC problem

III. System Model and Problem Statement

In this section, we formally present our system model and then present the mathematical formulations of the VNF placement problem.

A. System Model

Let R be the set of total SFCRs that arrive in time interval T. To make a distinction, we use $p \in P$ to denote the elements (which is a VNFR) in a specific SFCR and use V to indicate the specific VNF type that VNFR p demands. In each sample time $[t_\phi, t_{\phi+1}]$, we use $arr_r(t_\phi, t_{\phi+1})$ to denote the arrival rate of users' SFCRs. We assume that all arrival rates are fixed in $[t_\phi, t_{\phi+1}]$ but may vary in other sample times. Though the arrival rates are not guaranteed to be error-free, they are still a good reference for J-ORTC. We then assume that there are Φ sample times in T, and t_ϕ indicates the ϕth time point in T.

In most of existing works, the VNF instances are considered to be unshareable, which means each VNF instance can only serve one tenant. Different from the unshareable instances, we consider the VNF instances are shareable in this paper, which means a VNF instance can serve multiple tenants at the same time and provide lower service price, higher resource utilization, conveniences and flexibility for service providers. Specifically, we consider the VNFRs in SFCRs as the tenants in our paper. Hence, we only need to instantiate a specified VNF on one node. Notably, for one specified VNF, there exist multiple instances of it within the entire system.

B. Problem Formulation

The descriptions of the used notations are shown in Table 1.

Firstly, we need to guarantee that one VNFR p of an SFCR r is assigned to only one node n as captured by the following constraint:

$$\sum_{n\in N} x^r_{pn} = 1, r \in R, p \in P_r, \tag{1}$$

where x^r_{pn} indicates whether a VNFR p of an SFCR r is hosted ($x^r_{pn} = 1$) on node n or not ($x^r_{pn} = 0$).

We subsequently introduce the resource constraints for resource consumption. We firstly define the following binary:

$$y^v_n = \begin{cases} 1, & \sum_{r\in R}\sum_{p\in P_r} x^r_{pn}\cdot k^r_{pv} \geqslant 1, \\ 0 & \textit{otherwise,} \end{cases} \tag{2}$$

where $n \in N$, and k^r_{pv} indicates whether VNFR p in SFCR r demands ($k^r_{pv} = 1$) the type-v VNF instance or not ($k^r_{pv} = 0$). Therefore, Eq. (2) represents whether we need to deploy a type-v VNF instance on node n for the VNFRs that need it. Afterwards, based on [19], we formulate the processing consumption required by a VNFR p as follows:

$$prc_p(t_\phi, t_{\phi+1}) = \sum_{r\in R} k^r_{pv}\cdot arr_r(t_\phi, t_{\phi+1})\cdot \tau_v. \tag{3}$$

where τ_v is the resource consumption required by a type-v VNF instance to support its processing capacity. So Eq. (3) indicates the resource consumption of a VNF p for processing users' service requests in sample time $[t_\phi, t_{\phi+1}]$. Finally, we need to guarantee that the total resource consumption assigned to node n should be less than the resource capacity Q^{res}_n of that node in each time interval T. Hence, the constraint for resource consumption is

$$\sum_{r\in R}\sum_{p\in P_r} prc_p(t_\phi, t_{\phi+1})\cdot x^r_{pn} + \sum_{v\in V} bro_v\cdot y^v_n \leq Q^{res}_n, \forall n \in N, \tag{4}$$

where bro_v indicates the BRO to instantiate a type-v VNF instance.

Similarly, the traffic rate constraint in time interval T can be formulated as follows:

$$\sum_{r\in R}\sum_{\substack{p,p'\in P_r\\ p\neq p'}} trf^r_{pp'}\cdot x^r_{pn}\cdot x^r_{p'n'} \leq Q^{link}_{nn'}, n, n' \in N, n \neq n', \tag{5}$$

where $trf^r_{pp'}$ represents the required traffic rate of logical link (p, p'), and $Q^{link}_{nn'}$ represents the traffic rate capacity physical of link (n, n').

Based on Eq. (4) and (5), we have the following total resource consumption and the total traffic consumption.

$$I_{total} = \alpha\cdot\sum_{n\in N}\Bigg[\sum_{r\in R}\sum_{p\in P_r}\sum_{\phi\in\Phi} prc_p(t_\phi, t_{\phi+1})\cdot x^r_{pn} + \sum_{v\in V} bro_v\cdot y^v_n\Bigg], \tag{6}$$

$$G_{total} = \beta\cdot\sum_{\substack{n,n'\in N\\ n\neq n'}} l_{nn'}\sum_{r\in R}\sum_{\substack{p,p'\in P_r\\ p\neq p'}} trf^r_{pp'}\cdot x^r_{pn}\cdot x^r_{p'n'}. \tag{7}$$

In Eq. (6), we use α to indicate the cost of per unit resource consumption. Similar to α, let β denotes the cost of per unit required traffic capacity. As regards the total traffic consumption, we use the hop distance $l_{nn'}$ between nodes (n, n') to measure the required network traffic capacity, which is proved in [7]..

The combination of Eq. (6) and Eq. (7) is the joint optimization resource and traffic consumption (J-ORTC) problem. The objective function is as follows:

$$F_{total} = I_{total} + G_{total}. \tag{8}$$

Our target is to minimize the total cost incurred in the system,

$$\min F_{total}, \textit{s.t. Eq. 1 to Eq. 8.} \tag{9}$$

However, since the problem is NP-hard, the optimal result cannot be found in foreseeable time. So we propose the EBBO heuristic solution to solve J-ORTC in polynomial time.

IV. Enhanced Biogeography-based Optimization Algorithm For J-ORTC

A. BBO Algorithm

For BBO algorithm, various habitats correspond to various candidate solutions of the optimized problem. The Habitat Suitability Index (HSI) is a factor, which reflectis the quality of candidate solutions. The higher the HSI is, the better the solution will be. A candidate solution is represented by a Suitability Index Variables (SIVs) vector. There are two important operations for SIVs vectors: migration and mutation. The migration operation indicates the probabilistic sharing of information among habitats, and the mutation operation indicates the habitat changes caused by emergencies. With the help of these two operations, diverse suitability are obtained by diverse habitats. Eventually, the system can not only achieve a dynamic balance, but also obtain the final solution of the optimized problem.

1) Migration Operation: We denote the migration in a single habitat by a model. In the model, λ_s and μ_s respectively indicate the functions of immigration rate and the emigration rate. After plenty of experiments, we find that the Linear Migration Model can acquire the best effect. Hence, we use it as the final model for EBBO. For each habitat, the SIV migration operations will bring new habitats. For the habitats, whose HSI is relatively high, they can generate more SIVs. Therefore, more SIVs vectors with higher HSI habitats can be searched. The new habitats will then be ready for the next mutation operation. Population s, migration rate λ_s and emigration rate μ_s are three factors influence the mutation probability.

We use P_s to denote the probability which the habitat can exactly accommodate s populations. The function model is shown in Eq. (10), which indicates the P_s from t to $t + \Delta t$.

$$\begin{aligned} P_s(t+\Delta t) &= (1-\lambda_s \Delta t - \mu_s)P_s(t) \\ &+\lambda_{s-1}\Delta t P_{s-1} + \mu_{s+1}\Delta t P_{s+1} \end{aligned} \tag{10}$$

when $\Delta t \to 0$, we use Eq. (11) to derive the equilibrium value of P_s.

$$P_s = \begin{cases} \dfrac{1}{1+\sum\limits_{k=1}^{s^{max}} \frac{\lambda_0\lambda_1\cdots\lambda_{k-1}}{\mu_1\mu_2\cdots\mu_k}} & s=0 \\ \dfrac{\lambda_0\lambda_1\cdots\lambda_{s-1}}{\mu_1\mu_2\cdots\mu_s\Big(1+\sum\limits_{k=1}^{s^{max}} \frac{\lambda_0\lambda_1\cdots\lambda_{k-1}}{\mu_1\mu_2\cdots\mu_k}\Big)} & 1 \le s \le s^{max} \end{cases} \tag{11}$$

2) Mutation Operation: The BBO algorithm simulate natural disasters and diseases of a habitat via a mutation operation. The mutation probability m_s can be formulated as Eq. (12), where P_{max} indicates the max value of P_s. And m_{max} is a specific value, which indicates the upper limit of the mutation probability.

$$m_s = m_{max}(1-\frac{P_s}{P_{max}}) \tag{12}$$

B. Combination of J-ORTC Problem and EBBO Algorithm

First, we should establish the correspondences between the J-ORTC problem and BBO algorithm. Therefore, we convert constraints Eq. (4) and (5) into the constraint functions as shown in Eq. (13). Since the original BBO has a large state space of feasible solutions, it is inefficient to solve J-ORTC problem. Therefore, we carefully seek a penalty function and combine it with the original BBO algorithm. When a penalty function $\sum_k [max(h_i(x,k),0)]^2$ is added, whether a solution meets all constraints can be effectively distinguished. And this

---

Algorithm 1 J-ORTC approach with EBBO algorithm

Input: nodeList, SFCList, VNFList, communication cost matrix C
Output: the VNF placement scheme X

Initialize nodeList, SFCList, VNFList
Initialize parameters E, I, m_{max},the max amount of populations s^{max}, the times of generations G, the amount of elites E_n, dimension of SIVs vector D
Compute migration rates λ_s and μ_s,mutation rate m_s
Compute $prc_p(t_\phi, t_{\phi+1})$ and $trf^r_{pp'}$
Use F^*_{total} to randomly initialize and classify a set of habitats H in ascend
while g = 1 to G **do**
  Pick out E_n as best habitats
  while i = 1 to s^{max} **do**
    if $rand(0,1) < \lambda_i$ **then**
      while j = 1 to G **do**
        if $rand(0,1) < \mu_j$ **then**
          Randomly pick out a SIV from the habitat H_j
          Use this SIV in H_j to replace the corresponding SIV in H_i
        end if
      end while
    end if
  end while
  while i = 1 to s^{max} **do**
    while $j = 1$ to D **do**
      if $rand(0,1) < m_j$ **then**
        Use a random SIV to replace the j-th SIV of H_i
      end if
    end while
  end while
  Classify all habitats according to HSI
  Use the eltis to replace the e_n habitats
  Classiyf all habitats according to HSI again
end while
return the VNF placement scheme X from H

---

is the enhanced biogeography-based optimization (EBBO) algorithm. This penalty function eliminates solutions that violate constraints during the iteration, thus, to improve the efficiency of the algorithm. Afterwards, F^*_{total} is formulated with penalty multipliers $\eta_i \gg 1$, as shown in Eq. (14). Consequently, the optimization objective is to minimize $F^*(X)$.

$$\begin{cases} h_1(x) = \Big[\sum_{r\in R}\sum_{p\in P_r} prc_p(t_\phi, t_{\phi+1})\cdot x^r_{pn} \\ +\sum_{v\in V} bro_v \cdot y^v_n\Big] - \; Q^{res}_n \\ \\ h_2(x) = \; \sum_{r\in R}\sum_{\substack{p,p'\in P_r \\ p\neq p'}} trf^r_{pp'}\cdot x^r_{pn}\cdot x^r_{pn'} - \; Q^{link}_{nn'} \end{cases} \tag{13}$$

TABLE II
RESOURCE REQUIREMENT FOR DIFFERENT VNF TYPES

Parameter \ VNF types	FW	NAT	WOC	IDPS	TM	VOC
resource for processing (Mbps)	12.5	10.3	7.5	9.38	8.5	18.1
resource for instantiating (Mbps)	400	300	100	600	268	580

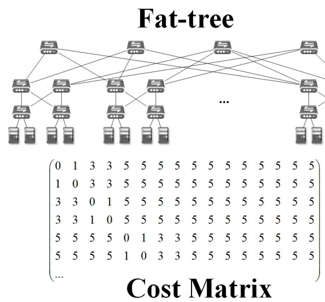

Fig. 1. The network topology and cost matrix.

$$F^{*}_{total} = F_{total} + \sum_{i} \eta_i \cdot \sum_{k} [max(h_i(X), 0)]^2 \quad (14)$$

Populations s can correspond to the number of used nodes. Therefore, λ_s and μ_s can be obtained with a specific migration model. SIVs vector is the solution to the VNF placement problem. Mutation operation can randomly modify SIVs based on P_s and m_{max}. The value of m_{max} is customized and tunable.

C. Implementation Algorithm of J-ORTC Approach

We introduce the implementation EBBO algorithm of J-ORTC approach, as shown in Algorithm 1. E_n is the amount of the elite solutions, and we make $E_n = 2$. Steps 1-5 are initialization, then the optimization loop begins. The migration operation is steps 8-17, we use λ_s and μ_s to decide the amount of information to share among habitats.The mutation operation is steps 18-24, we use μ_s to decide the SIVs that are required to be updated. In the end, habitats should be classified by HSI. At the end of iterative process, we can obtain the final scheme of the VNF placement.

V. EXPERIMENTAL EVALUATIONS

In this section, to validate the efficacy of J-ORTC approach, we evaluate it in detail and compare it with other state-of-the-art approaches. We conduct experiments in Java Alevin [20], which is a simulation environment widely used in VNF placement problem.

A. Simulation Setup

First, we consider three types of SFCs whose configurations are set according to three representative services. And the specific VNFs in these SFCs are demonstrated in Table 2. The required resource consumption of these VNFs conforms to commercial appliances in [19], as shown in Table 3. Then we set the time interval T is a day, and each sample time $[t_\phi, t_{\phi+1}]$ is an hour. Based on the topology in [7], we use the Fat-tree cost matrix C to describe the network topology of the system, as shown in Fig. 1.

TABLE III
SFCS AND NETWORK TRAFFIC REQUIREMENTS

SFC types	**Used VNFs**	**required traffic rate ($trf^{r}_{pp'}$)**
Web Service	NAT-FW-WOC-IDPS	100kbps
VoIP	NAT-FW-TM-FW-NAT	64kbps
Online Game	NAT-FW-VOC-WOC-IDPS	50kbps

Afterwards, we use the trace data from Facebook [10] within 5 time intervals to evaluate the performance of J-ORTC approach. Moreover, we use the online learning method FTRL [9] to derive users' SFCs requests from the trace data. In terms of basic configuration, α and β are respectively set to 0.05 and 0.1, each node has 5 Gbps resource capacity to hold VNF instances, and the link capacity is set to 10 Gbps.

We use the following approaches as benchmarks to evaluate the efficacy of J-ORTC approach.

- *Optimal baseline.* We use the JuMP Solver [21] to derive the optimal solution. Then, the gap between the scheme of J-ORTC approach and the optimal results is revealed. Since the optimal VNF placement problem is NP-hard, the runtime of JuMP solver grows exponentially in large scale problems.
- *First-fit placement.* First-fit (FF) [22] algorithm selects the first unsaturated VNF instance with sufficient residual traffic capacity. If not, it instantiates a VNF instance in the first unsaturated node with enough available traffic capacity.
- SAMA: We also compare J-ORTC with SAMA [7], which is an algorithm based on Markov approximation approach.

B. Simulation Results

1) Comparison with two baselines. In this section, we first demonstrate the performance of different approaches in optimizing resource consumption. Fig. 2 shows that J-ORTC approach performs much better than FF and SAMA, when only resource consumption is considered. In terms of network traffic consumption, EBBO has no obvious advantage over these two benchmarks, as shown in Fig. 3. It is because J-ORTC is inclined to place VNFs of one SFC in multiple nodes, and then there is more network traffic consumption between the nodes. As for the total incurred consumption, we can see from Fig. 4 that J-ORTC performs the best in all considered time intervals. Since FF and SAMA do not consider the arrival

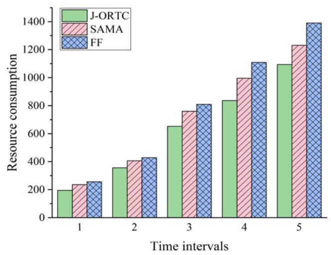

Fig. 2. Performance of different approaches in resource consumption.

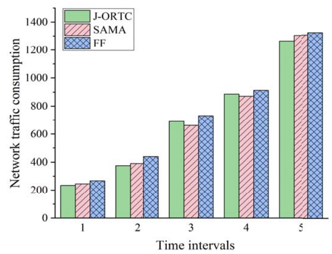

Fig. 3. Performance of different approaches in traffic consumption.

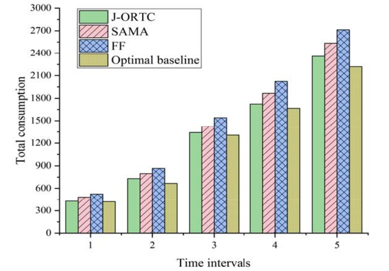

Fig. 4. Performance of different approaches in total consumption..

rates of users' SFCRs, which makes the total consumption increase over a long term.

2) Comparison with optimal results. Due to the inherent complexity of the VNF placement optimal problem, the time complexity of the optimal approach turns out to be exponential when the number of SFCRs is large. Therefore, with the help of JuMP Solver [21], we derive the optimal results with a small number of SFCRs. Fig. 4 illustrates the performance gap between J-ORTC and the optimal baseline. We can see that J-ORTC can obtain near effect compared with the optimal baseline derived by the JuMP Solver.

VI. Conclusion

In this paper, we study the VNF placement for cloud service providers, considering resource consumption and network traffic consumption. Moreover, we take the arrival rates of users' SFCs requests into consideration, that can derive a more precise resource consumption to improve network resource utilization. Then we formulate the problem as a constrained joint optimization problem and establish a suitable model, aiming to minimize the total consumption incurred in the system. Afterwards, we propose an approach called J-ORTC with EBBO algorithm to solve the problem. In order to evaluate the efficiency of J-ORTC approach, we compare it with several current approaches. Our simulation results show that the scheme derived by J-ORTC is close to the optimal result and outperforms the benchmarks.

References

[1] Han Bo, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015.

[2] Zilong Ye, Xiaojun Cao, Jianping Wang, Hongfang Yu, and Chunming Qiao. Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization. *IEEE Network*, 30(3):81–87, 2016.

[3] Rami Cohen, Liane Lewin-Eytan, and et al. Near optimal placement of virtual network functions. In *IEEE INFOCOM*, pages 1346–1354, 2015.

[4] Po Wen Chi, Yu Cheng Huang, and et al. Efficient nfv deployment in data center networks. In *IEEE International Conference on Communications*, 2015.

[5] Defang Li, Peilin Hong, Kaiping Xue, and Jianing Pei. Virtual network function placement considering resource optimization and sfc requests in cloud datacenter. *IEEE Transactions on Parallel Distributed Systems*, pages 1–1, 2018.

[6] Chang Jie Guo, Wei Sun, and et al. Study of software as a service support platform for small and medium businesses. In *New Frontiers in Information and Software as Services.*, pages 1–30, 2011.

[7] Chuan Pham, Nguyen H. Tran, and et al. Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach. *IEEE Transactions on Services Computing*, pages 1–1, 2017.

[8] D. Simon. Biogeography-based optimization. *IEEE Transaction on Evolutionary Computation*, 12(6):702– 713, 2008.

[9] H. Brendan Mcmahan. A unified view of regularized dual averaging and mirror descent with implicit updates. *Computer Science*, abs/1009.3240:2011, 2011.

[10] Facebook. Open sourcing pue, wue dashboards. Available:https://code.facebook.com/posts/272417392924843/ open-sourcing-pue-wue-dashboards.

[11] Tung Wei Kuo, Bang Heng Liou, and et al. Deploying chains of virtual network functions: On the relation between link and server usage. In *IEEE Infocom -the IEEE International Conference on Computer Communications*, 2016.

[12] Windhya Rankothge, Frank Le, and et al. Optimizing resources allocation for virtualized network functions in a cloud center using genetic algorithms. *IEEE Transactions on Network Service Management*, 14(2):343–356, 2017.

[13] M. Bagaa and et al. Service-aware network function placement for efficient traffic handling in carrier cloud. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2402–2407, 2014.

[14] Wenrui Ma, Oscar Sandoval, and et al. Traffic aware placement of interdependent nfv middleboxes. In *Proc. of IEEE INFOCOM*, 2017.

[15] Attila Takacs, Howard Green, Meral Shirazipour, Xia Ming, and Zhang Ying. Network function placement for nfv chaining in packet/optical datacenters. *Journal of Lightwave Technology*, 33(8):1565–1570, 2015.

[16] Xiaoke Wang, Chuan Wu, and et al. Online vnf scaling in datacenters. In *Proc. of IEEE ClOUD*, 2016.

[17] Rashid Mijumbi, Sidhant Hasija, Steven Davy, Alan Davy, and Raouf Boutaba. A connectionist approach to dynamic resource management for virtualised network functions. In *International Conference on Network Service Management*, 2017.

[18] Xi. Zhang, C. Wu, Z. Li, and F. C. Lau. Proactive vnf provisioning with multi-timescale cloud resources: Fusing online learning and online optimization. In *Proc. of IEEE INFOCOM*, 2017.

[19] Vincenzo Eramo, Emanuele Miucci, and et al. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking*, 25(4):2008–2025, 2017.

[20] J. Gil Herrera and et al. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532, 2016.

[21] Jump. Available:http://jump.readthedocs.org/en/latest/index.html.

[22] Jialei Liu, Shangguang Wang, Ao Zhou, Fangchun Yang, and Rajkumar Buyya. Availability-aware virtual cluster allocation in bandwidth-constrained datacenters. *IEEE Transactions on Services Computing*, PP(99):1–1, 2018.