Joint Service Chain Deployment and Manager Placement in NFV

Parham Alvani[a], Bahador Bakhshi[a,*]

[a] *Amirkabir University of Technology, Tehran, Iran*

Abstract

a) Motivation

b) Introduce the problem

c) Why is there a research gap

d) What have we done

e) What are the results

In the old times, Network providers use physical network functions to create their service chains, but a single change in this manner is difficult and may cause many service disruption. SFC and NFV is the solution to this difficulty. By using SFC and NFV, providers can provision chains dynamically and then change them in the run-time. One of the main requirements this virtualized environment is management and monitoring for the chains. In this research, we consider the chain acceptance problem subject to management resources. In the first step, we formulate problem with ILP and then implement it in CPLEX framework. The optimal solution takes time to solve so we need a Polynomial-Time solution to the problem. In this research, we develop a

*Corresponding author.

Email addresses: `parham.alvani@aut.ac.ir` (Parham Alvani), `bbakhshi@aut.ac.ir` (Bahador Bakhshi)

heuristic algorithm and compare its result with the optimal solution. In the end, the heuristic solution produces near-optimal results in the polynomial time.
Keywords:

1. Introduction

a) A general introduction of the context

b) More specific context of problem and its importance

c) A very brief review of literature and introduce the research gap

d) What is the problem

e) What are the contributions

f) The structure of the paper

Telecommunications industry has traditionally been responsible for setting up and deploying services. Moreover, network operators deploy proprietary and custom-built hardware and equipment for each function in their infrastructure. Stability and high-quality results can earn the trust of service providers in regards to proprietary hardware.

However, we are facing increased users' demand for a variety of short-lived and high traffic services. Therefore, service providers should constantly upgrade themselves in terms of purchase, store, and deployment of new physical equipment. These implementations in turn increases the overall expenditure of service providers. In addition, with the increase in total number of equipment, the amount of available space for deploying new equipment shrinks. Finally, increased expenditure for retraining staff to gain the knowledge of working with new equipment should also be considered. The worst-case scenario is that, with the acceleration of innovation in services and technology, the hardware life cycle becomes shorter, which prevents innovation in network services.

In the traditional method of deploying network service, user traffic must pass through a number of network functions in a certain order to create a traffic processing route. At present, these functions are connected to each other in the form of hardware, and traffic is forwarded to them using routing tables. The main challenge of this method is that it is difficult to establish and change the order of functions. For example, over time, as network conditions change, we need to change the connectivity and location of functions to better serve users, which requires moving functions and changing routing tables. In the traditional way, it was a difficult and costly task that can lead to many errors. On the other hand, the rapid change of services desired by users requires a rapid change in the order of functions. In the current method, these changes are difficult to make. Therefore, network operators have found the need for programmable networks and dynamic service chaining.

In recent years, software-based networks and network virtualization have received much attention. Service providers have begun to move towards virtualized and software networked functions. Therefore, they will be able to provide innovative services to users. This process creates an option for service providers in regards to proprietary hardware and reduces the cost of setting up and maintaining the service.

As mentioned, with development of softwarized functions, service providers' reliance on proprietary hardware is reduced, and services can be scaled up / down quickly. Finally, virtualization of network functions and chaining of service functions are solutions that have been proposed for this purpose.

As mentioned, the main idea of virtualization of network functions is to separate the physical equipment of the network from its function. This means that a network function such as a firewall can be deployed on HVS servers as simple software. Therefore, a service can be deployed using virtual network functions, which can themselves be implemented as software and run on one or more physical standard hosts.

Virtual network functions can be relocated or prototyped in different locations without the need to purchase and install new equipment.

It is worth mentioning that SDN discusses forwarding tables and their centralized updates. This discussion can take place alongside NFV as well, as the NFV discussion is at the service layer and does not talk about how to update forwarding tables. Therefore, what is stated in this study is related to NFV networks, but it can also be used in networks that have SDN infrastructure. In the future, it will be possible to have networks whose services are based on SFC and NFV, and forwarding tables will be updated using SDN.

The problem of deploying virtual network functions is one of the major challenges in allocating resources to service function chaining and had great attention in past few years. The problem of embedding virtual network functions is divided into two sub-problems of mapping of virtual nodes and virtual links, which must be considered simultaneously.

There are many mapping limitations that need to be considered. The physical resources selected from the infrastructure network must meet the functional requirements of the virtual network. For example, the processing power of virtual functions must be less than or equal to the processing power of the physical node on which the mapping is performed. The need for a specific physical node in a function must also be considered.

In addition, there is a set of restrictions that apply to service function chaining. One is the existence of VNFM for the management of the life cycle of functions in these networks, which due to the importance of the delay rate of the connection between the virtual network function and VNFM must be located in a suitable place. Therefore, a new sub-problem is added to the main issue.

The issue of embedding service function chaining is of great importance. A great number of researches has been conducted on the issue. In addition, the issue of management and supervision of these chains is also raised. The present study discusses the issue for the first time, which increases the importance of the results. Nowadays, due to the importance of monitoring, huge amount of expenses is spent on monitoring data centers. In many cases, second thought monitoring is to bring about its own drawbacks and disadvantages in the future. This study intends to consider the need for supervision at the time of their mapping to prevent future losses.

The main idea of this research is to provide a comprehensive and complete solution that covers all aspects of the problem of embedding service function chaining. In fact, in addition to considering the main dimensions of the embedding problem, the acceptance control mechanism, the applicability of the solution to different conditions and the existence of limitations of the node and edge, other dimensions have been considered. Due to the existence of VNFM in the form of a special node and the importance of virtual network and VNFMs' connection delay, a placement and mapping step has been added to the main problem. The above are some restrictions on the connections between virtual network functions and VNFMs, and it is assumed that managing a certain number of virtual functions requires a license at a certain cost.

One of the main innovations of this research is the definition of a problem along with paying attention to the management needs, which allows the system administrator to implement and adjust all the required policies. The present study examines the issue of placement of service function chaining along with the limitations of management resources and formulates it in the form of an acceptance control problem in a linear manner.

Another innovations of the present research is the presentation of a Heuristic method based on [1] and [2] methods, which further enhances its implementation time and final output.

In NFV ecosystem, each chain must be monitored and managed by a VNFM. Similar to other virtual functions, VNFMs need computing and network resources and can manage a limited number of chains. Our work considers Service Chain Placement Problem and Manager Placement Problem jointly and wants to place chains and their corresponding VNFM at the same time. We believe this work hasn't been done in the literature before. By considering the joint problem, you may not accept a chain that you don't have any management resource for it, or you can get many chains that have little management resource requirement. These considerations create a better solution with more profit to datacenter from provisioning the set of chains, and the results in 6.1 approves it.

The paper structure as follow in 2 we reviewed the literature then describe the problem in 3 and formulate it in 4 as ILP. In 5 we proposed a polynomial time solution and evaluated its results in 6. We concluded the paper in 7.

2. Related Work

In this section, we review the works that have been done on service chain deployment and resource assignment in NFV, and mainly focus on the works that consider management requirements of chains.

2.1. Resource Allocation in NFV

The service chain deployment problem specially for virtual mobile core network functions has recently received substantial attention in the literature [3][4][5]. This problem is composed of two main steps: allocating computing and memory resources for virtual machines to run the chains' virtual network functions and assigning network resources to the chains' virtual links [4]. Various versions of this problem with different objective functions and considering wide range of constraints have been investigated in the literature [4]. Placement problem has many variations and we concentrate on VNF Placement with VNF Chain Placement. There are many different objectives like minimizing the deployment cost, increasing profit, minimizing the resource usage and etc. [4][5] Also there are many other variations to Chain Placement Problem such as online, dynamic, schedule and etc. [6]

From the objective function standpoint, deployment of service chains can be optimized for different goals. One of the most common category is the approaches that utilize SFC deployment as a tool to maximize the business profit of the infrastructure provider [7][8][9][10]. In these approaches, the objective is to maximize the revenue gained by accepting SFC requests while minimizing the cost of physical servers and links [11][12][10], the cost of VNF images and licenses [11]. Energy-awareness is another objective that has been considered in SFC deployment problem [8][13][14], wherein, it is aimed to minimize the power footprint of the infrastructure provider by turning off the unused servers and links via green networking techniques, e.g., traffic consolidation [15]. While QoS requirements are typically considered as the constraints of the SFC deployment problem, some work have optimized QoS metrics, e.g., delay and congestion, as the objective function of the problem [10][16][17]. Finally, it is worth to note that there are plenty of work in the literature that aimed to optimize multiple objectives simultaneously, e.g., load balancing and reconfiguration overheads [18], ...

The constraints of the SFC deployment problem are the second differentiating factor of the existing work on this problem. The required computing resources for virtual functions and bandwidth for virtual links of chains as well as the limited capacity of the infrastructure network resources are the most basic constraints in the SFC problem [9][7][10]. In addition to these basic constraints, some more realistic assumptions as well as QoS requirements of chains have also been considered. While VNFs are software images, their processing capacity is typically limited by software implementation issues and license beside the underlying hardware capacity; these limitations was considered in [19]. Authors also took into account different QoS requirements of service chains, for example end-to-end delay [16][20] and reliability [16][21][22]. Finally, it should be noted that management requirements of service chains are another important category of constraints, which have received a little attention yet [2][23][24]. We will elaborate it in the following subsection.

Several solution methods have been utilized to approach the SFC deployment problem. Mainly, the problem is formulated as a Mixed Integer Linear Programming (MILP) model; and since it is NP-Hard [7][25], various approaches have been proposed to tackle its complexity. Optimization theory techniques like decomposition methods and Lagrangian relaxation methods were proposed [26][14][27]. Another category is the solutions based on meta-heuristic algorithms, e.g., Tabu search, genetic algorithm, and ... [2]. The problem specific heuristic algorithms are also very common [28][1][7]. Available information about demands is another important issue in solution approaches. In off-line approaches, it is assumed that whole demands are known at the beginning, while in on-line solutions, demands arrive one-by-one [29].

Until now, we briefly reviewed the work on general SFC deployment problem; interested readers can refer to [4] for more details. In comparison to previous works, here, we consider profit maximization as the objective function, formulate the problem as MILP model, and use Tabu search to solve it. However, in addition to computation and bandwidth requirements, we consider the management requirement which have not investigated in the literature except very few studies that are discussed in the following.

2.2. Management Resources in NFV

As already mentioned, this work considers management requirements in SFC deployment. To the best of our knowledge, work [2], its next work [23], and reference [24] are the only ones that consider VNFM and other management resources in SFC deployment.

In [2], the authors studied the problem of VNFM placement in a distributed NFV infrastructure under the assumption that chains have already been deployed and consequently, the location of VNFs are known. The objective function of the VNFM placement problem is to minimize the operational costs which is composed of life cycle management cost, compute resources cost, migration cost, and reassignment cost. Delay constraint on management links is also considered. The authors used Tabu search algorithm to find a polynomial solution. They start from a feasible placement, and in each step, they improve VNFM placement by doing reassignment, relocation, bulk reassignment, and deactivation. While this is the first work that investigated the management resource assignment in NFV, it does not *jointly* deploy SFCs and VNFMs.

In [23], authors extended their previous work [2], and solve the VNFO placement problem. Authors consider the same system model as [2], but here they aim to place VNFO and VNFM jointly to minimize operational cost as defined in [2]. They propose a two step placement algorithm that first place VNFOs and then place VNFMs, which use Tabu search method. That paper also assumes that SFCs have already been deployed.

In [24], authors consider autonomy for VNFMs, so they select their managed VNFs dynamically. The authors use game theory to achieve a distributed solution to the VNFM Placement Problem as described in [2]. In that paper, again it is assumed that the location of VNFs are given and VNFM placement is the only problem.

To conclude, as discussed, the previous work on management resource in NFV, are based on the system model of [2] wherein it is assumed that service chains are deployed beforehand. But, here, we aim to *jointly* optimize SFC deployment and SFC placement. Moreover, we consider license cost for VNFM instances and in our model, each physical node has its specific list of nodes that can run its VNFM that provides a great flexibility for implementing management policies.

3. System Model and Problem Statement

In this section, after discussing the assumptions and system model, the problem of joint SFC deployment and VNFM placement is stated and clarified by an illustrative example.

3.1. Assumptions

In the JSD-MP problem, a set of SFCs need to be deployed in the physical infrastructure network. The main assumption is that in addition to computational and network resources, a VNFM should be assigned for each SFC to manage the VNFs of the SFC. All VNFs of a chain should be managed by a single VNFM, but a VNFM can manage multiple SFCs. The capacity of each VNFM is limited in terms of the number of VNF instances. Each VNFM needs a license for managing the specific number of VNF instances. Moreover, we assume that only a subset of physical servers can host VNFMs and similarly, each VNF type can only be placed on a subset of physical servers. It is assumed that traffic ingress and egress of each chain are conducted by a special type of NFVs, which need to be placed on ingress and egress nodes of the physical network respectively.

VNF instances deployed on a physical server can only be managed by the VNFMs placed on a given subset of physical servers. To maintain the timing requirement of the management traffic, the distance (in term of hop-count) between each VNFM and its associated VNFs should be less than the specified threshold. Also, JSD-MP considers VNFs can't shared between chains.

3.2. System Model

The physical infrastructure topology is modeled as a directed graph $G^p = (V^p, E^p)$, where V^p is the set of NFVI-PoPs and E^p is a set of the inter-PoP links. The computational capacity of PoP $i \in V^p$ is specified by c_i^p number of CPU cores and m_i^p gigabytes of RAM. Bandwidth of link $(i, j) \in E^p$ is specified by $b_{(i.j)}^p$.

The set of requested SFCs is denoted by R. Each request $r \in R$ has revenue c_r and is represented by directed graph $G_r^s = (V_r^s, E_r^s)$, where V_r^s is the set of VNFs and E_r^s is the set of the virtual links of the chain. The type of VNF $v \in V_r^s$ is denoted by $t_{v,r} \in T$ where T is the set of the types of the functions. Each type t determines the required number of CPU cores c_t^s and the required amount of memory m_t^s to create an instance of that type. The required bandwidth of each virtual link $(u, v) \in E_r^s$ is denoted by $b_{(u,v),r}^s$.

Each VNFM requires c^m number of CPU cores and m^m gigabytes of RAM to handle κ number of VNFs at most. To lunching a VNFM, the service provider has to pay the license fee ϕ. There is a dedicated virtual management link with bandwidth b^m between every VNF instance and its associated VNFM. The restrictions of the placement of VNFMs and VNFs, discussed in the previous section, are formulated by a few given parameters. We use ρ to indicate the maximum hop counts for each virtual management link. Parameter $\eta_{(i,j)}$ indicates whether the VNFM mapped to the physical server $i \in V^p$ can manage the VNFs placed on the physical node $j \in V^p$ or not. The notations discussed here are summarized in Table 1.

3.3. Problem Statement

Under the assumptions of subsection 3.1, the JSD-MP problem is defined as following using the notations defined in subsection 3.2. The NFVI service provider owns the infrastructure network G^p. A set R of requests are given. The service provider's goal is to accept a subset of requests that maximize the profit. To accept a request r, the service provider should allocate the $c^s_{t_{v,r}}$ CPU cores, and $m^s_{t_{v,r}}$ amount of memory for all $v \in r$; moreover, $b^s_{(u,v),r}$ bandwidth should be allocated for all links $(u, v) \in E^s_r$, and a VNFM needs to be assigned for the chain. These allocations should satisfy the requirements mentioned in subsection 3.1. The profit is made by the revenue obtained from accepting the requests while the service provider should pay the license cost of ϕ for each VNFM.

The idea behind the JSD-MP problem is that in the placement of VNFM for a chain, the location of the VNFs of the chain should be taken into account. Therefore, by jointly conducting of SFC deployment and VNFM placement, the service provider can maximize the profit.

This problem is clarified by an illustrative example in the following subsection.

3.4. Illustrative Example

In this section, an illustrative example is presented to gain more sight into the JSD-MP problem. Figure 1 shows two request chains where the type of VNFs and the required bandwidth of the virtual links in Mbps are

Table 1: Parameters

V^p	the set of NFVI-PoPs
E^p	the set of the inter-PoP links
c_i^p	number of CPU cores of server i
m_i^p	amount of RAM in server i
$b_(^p i, j)$	bandwidth of link $(i, j) \in E^p$
V_r^s	the set of VNFs for the request r
E_r^s	the set of virtual links of the request r
m_t^s	required RAM of VNF instance with type $t \in T$ in GB
c_t^s	required CPU cores of VNF instance with type $t \in T$
m^m	required RAM of VNFM in GB
c^m	required CPU cores of VNFM
κ	maximum number of VNF instances that a VNFM can handle
$\tau(v, t)$	function that returns 1 if the VNF instance v has type t
$b_{(u,v),r}^s$	required bandwidth for virtual link $(u, v) \in E_r^s$
b^m	required bandwidth for each virtual management link
ρ	maximum hop count of management links
ϕ	VNFM license fee
$\eta_{(i,j)}$	binary parameter that is 1 if VNFs on physical server i can manage by VNFMs on physical server j

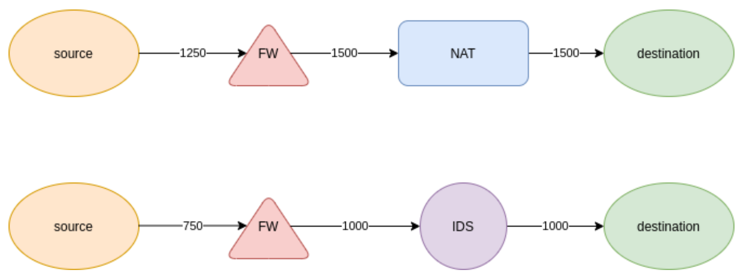


Figure 1: Chains for illustrative example

Table 2: VNFs type of illustrative example

Spec/VNF	vFW	vNAT	vIDS
CPU (vCore)	2	2	2
Memory (GB)	2	4	2

depicted. Each of these requests worth 100\$. The NFVI service provider owns the infrastructure that is shown in Figure 2. VNF's types are described in Table 2, and Table 3 shows the physical servers specifications. There a few constraints on placing the VNFs and capacity of the NFVI as following. Instances can only run on servers 1, 3, 5, and 7 since the remaining servers are fully utilized. The VNFs on servers 1 and 3 can only be managed by the VNFMs on servers 2 and 4. The VNFs on server 5 can only be managed by the VNFMs on servers 4 and 6. The VNFs on server 7 can only be managed by the VNFMs on servers 6 and 8. Each VNFM can handles at most 5 instances with 4 GB of RAM and 2 CPU cores and license that costs 10\$. Management links requires 10Mbps reservation and maximum hop counts for management is 10. All physical links has 40Gbps capacity.

The optimal solution of this instance of the JSD-MP, which is obtained by the MILP formulation presented in section 4, is as following:

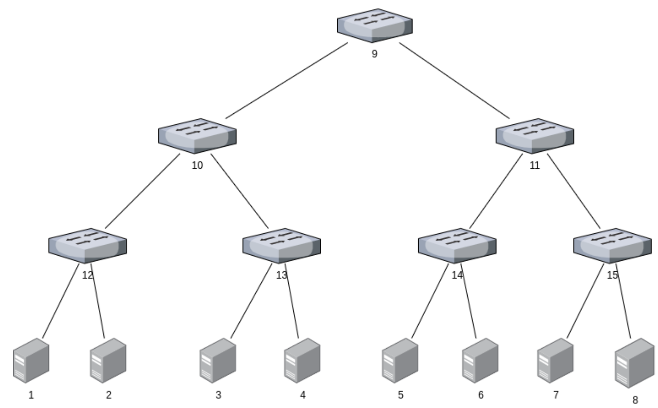


Figure 2: Topology of illustrative example

Table 3: Server specification of illustrative example

	Server 1,2,7,8	Servers 3,4,5,6
Installed vCPU	144	72
Installed Memory (GB)	1408	288
Link (Gbps)	40	40

Table 4: Chain-1 Instance Mapping of illustrative example

0: Source	1: FW	2: NAT	3: Destination	VNFM
9	1	3	9	4

Table 5: Chain-1 Instance Mapping of illustrative example

0: Source	1: FW	2: IDS	3: Destination	VNFM
9	3	3	9	4

Two chains are accepted with using two VNFM instances that cost 20 so the total revenue is 180. Instance mappings are shown in 4 and 5. Also link mappings are shown in 6.

4. Problem Formulation

In this section, the JSD-MP problem is formulated as a MILP model. To this, at first, the variables of the model are introduced. Then the constraints are discussed; and finally the objective is presented.

The variables of the model are listed in Table 8. x_r is a binary variable that corresponds to acceptance of demand $r \in R$. Integer variable y_{it} is the number of VNFs of type $t \in T$ on server i. Binary variable z_{vi}^{t} indicates whether an instance of VNF type t is running on server i or not. The integer variable $\bar{y}_i$ is the number of VNFMs placed on server i. The VNFM assignment of chain is represented by binary variable $\bar{z}_{ri}$. Finally, the $\pi_{ij}^{(}u, v)$ and $\pi_i^v j$ are binary variables that show the virtual link $(u, v$ is mapped on link (i, j) and link (i, j) is used for managing the VNF v respectivly.

Table 6: Chain-1 Link Mapping of illustrative example

Virtual Link	Physical Links
(0, 1)	(9, 10) (10, 12) (12, 1)
(1, 2)	(1, 12) (12, 10) (10, 13) (13, 3)
(2, 3)	(3, 13) (13, 10) (10, 9)
VNF 0 Mgmt.	(9, 10) (10, 13) (13, 4)
VNF 1 Mgmt.	(1, 12) (12, 10) (10, 13) (13, 4)
VNF 2 Mgmt.	(3, 13) (13, 4)
VNF 3 Mgmt.	(9, 10) (10, 13) (13, 4)

Table 7: Chain-2 Link Mapping of illustrative example

Virtual Link	Physical Links
(0, 1)	(9, 10) (10, 13) (13, 4)
(1, 2)	—
(2, 3)	(3, 13) (13, 10) (10, 9)
VNF 0 Mgmt.	(3, 13) (13, 10) (13, 4)
VNF 1 Mgmt.	(3, 13) (13, 4)
VNF 2 Mgmt.	(3, 13) (13, 4)
VNF 3 Mgmt.	(9, 10) (10, 13) (13, 4)

Table 8: Variables

x_r	binary variable assuming the value 1 if the hth SFC request is accepted; otherwise its value is zero
y_{it}	the number of VNF instances of type $t \in T$ that are set up in a server $i \in V^p$
z^t_{vi}	binary variable assuming the value 1 if the VNF node $v \in \cup_{r=1}^{R} V^s_r$ is served by the VNF instance of type $t \in T$ in the server $i \in V^p$
$\bar{y}_i$	the number of VNFMs (each VNFM has its capacity and license fee) that are set up in server $i \in V^p$
$\bar{z}_{ri}$	binary variable assuming the value 1 if rth SFC is assigned to VNFM on server $i \in V^p$
$\pi^{(u,v)}_{ij}$	binary variable assuming the value 1 if the virtual link (u, v) is routed on the physical network link (i, j)
$\bar{\pi}^v_{ij}$	binary variable assuming the value 1 if the management of VNF node v is routed on the physical network link (i, j)

Each node has limited installed memory. Each VNF instance based on its type, and each VNFM use a specific amount of it. Constraint 1 limits the used memory for each node by its installed memory.

$$\sum_{t \in T} y_{it} m^s_t + \bar{y}_i m^m \leq m^p_i \quad \forall i \in V^p \tag{1}$$

Each node has limited amount of processing power in terms of CPU cores.

Each VNF instance based on its type, and each VNFM use a specific amount of it. Constraint 2 limits the used processing power for each node by its installed CPUs.

$$\sum_{t \in T} y_{it} c_t^s + \bar{y}_i c^m \leq c_i^p \quad \forall i \in V^p \tag{2}$$

If a chain's node is served on a physical node, there must be a set up VNF instance on it. Constraint 3 controls this relationship.

$$\sum_{v \in \cup_{r=1}^{R} V_r^s} z_{vi}^t \leq y_{it} \quad \forall i \in V^p, \forall t \in T \tag{3}$$

Constraints 4 and 5 only accept a chain that all of its nodes are placed on physical servers and have an assigned VNFM.

$$x_r = \sum_{t \in T} \sum_{i \in V^p} z_{vi}^t \quad \forall v \in V_r^s, \forall r \in R \tag{4}$$

$$x_r = \sum_{i \in V^p} \bar{z}_{ri} \quad \forall r \in R \tag{5}$$

Constraint 6 limits managers capacity.

$$\sum_{r \in R} \bar{z}_{ri}.|V_r^s| \leq \kappa.\bar{y}_i \quad \forall i \in V^p \tag{6}$$

Constraint 7 assures that each chain's node must be serviced with a VNF from its type.

$$z_{vi}^t \leq \tau(v,t) \quad \forall i \in V^p, \forall t \in T, \forall v \in \cup_{r\in R} V_r^s \tag{7}$$

Each physical server have an array of servers that cannot manage it. Constraint 8 represents it mathematically.

$$\begin{aligned} & 1 - z_{vi}^t + \bar{z}_{rj} = 0 \\ & \forall i \in V^p, \forall j \in V^p \eta_{(i,j)} = 1 \\ & \forall r \in R, \forall v \in V_r^s, \forall t \in T \end{aligned} \tag{8}$$

Constrains 9 and 10 shows flow conservation in service and management links respectively.

$$\sum_{(i,j)\in E^p} \pi_{ij}^{(u,v)} - \sum_{(j,i)\in E^p} \pi_{ji}^{(u,v)} = \sum_{t\in T} z_{ui}^t - \sum_{t\in T} z_{vi}^t \\ \forall i \in V^p, (u,v) \in E_r^s, r \in R \tag{9}$$

$$\sum_{(i,j)\in E^p} \bar{\pi}_{ij}^v - \sum_{(j,i)\in E^p} \bar{\pi}_{ji}^v = \sum_{t\in T} z_{vi}^t - \bar{z}_{ri} \\ \forall i \in V^p, v \in V_r^s, r \in R \tag{10}$$

Constraint 11 limits the consumed bandwidth in each link by its capacity.

$$\sum_{v\in \cup_{r\in R} V_r^s} \bar{\pi}_{ij}^v . b^m + \sum_{r\in R} \sum_{(u,v)\in E_r^s} \pi_{ij}^{(u,v)} . b_{(u,v),r}^s \leq b_{(i,j)}^p \\ \forall (i,j) \in E^p \tag{11}$$

Constraint 12 controls the management radius.

$$\sum_{(i,j)\in E^p} \bar{\pi}_{ij}^{v} \leq \rho \quad \forall v \in \cup_{r\in R} V_r^s \tag{12}$$

The objective is to maximize profit from placing chains:

$$max \sum_{r\in R} c_r x_r - \sum_{i\in V^p} \phi.\bar{y}_i \tag{13}$$

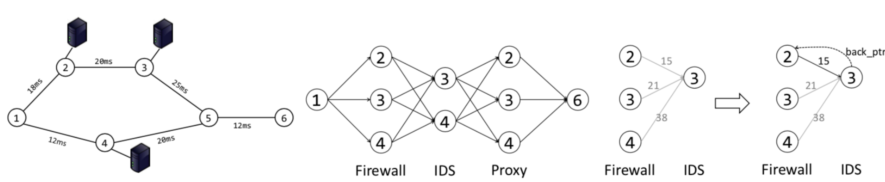
I don't know it should be here or it is correct or not The problem formulates as ILP and it can be reduced to bin packing so it is NP-Hard.

5. Proposed Solution

We have formulated the JSD-MP problem in ILP and its optimal solution is NP-Hard so it takes a long time to solve and data centers need a faster way for their placement. In this section we represent a way to have a near-optimal solution in polynomial time for deciding quickly on SFC request with management resources and it can be used at large datacenters with many nodes and requests.

5.1. Maximizing the Accepted SFC Requests with Management Constraints (MASMAN)

As we mentioned before JSD-MP problem consists of placing chains that each chain placmenet includes two parts which are VNF placement, that means placing chain's VNFs, and VFNM placement, which means placing the chain VNFM, that we formulate them together. For each chain we divide the placement into two different phase that are solved using different algorithms but for considering the managers we are going the reserve at least one VNFM resource on each physical node before the placement phase. The first phase is the VNF placement phase that we are solve it with Bari [1] algorithm. This algorithm is an efficient dynamic programming algorithm based on Veterbi algorithm for SFC placement that can be used here but it doesn't consider the management resources so we need to tweak it. [1] algorithm for each chain solves a Dynamic Programming problem that finds the minimum cost path for the chain in a multi-stage graph. It decides each node location with its previous one to minimize the cost variable because it believs a minimum cost path consists of minimum cost sub-paths. Each stage in the graph is a one network function that needs a placement and graph nodes are the candidates. Each link is illustrated with edges between stages in this graph. At the end, we used Tabu Search for improving the VNFM placement. By improving we mean merging VNFMs to reduce the license cost. In tabu search we randomly select two physical nodes to merge them and also update the tabu table to not check them again at the end if this merge is fasible we wil update the min cost and continue the iteration.



(a) Example Network Topolog (b) Multi-Stage Graph (c) A Single Stage

Figure 3: [1] algorithm transmissions to create multi-stage graph

Another way for placing the chains is to use [1] algorithm but add new

Algorithm 1 MASMAN Algorithm - 1

```
Require: chains, topoloy
function MASMAN__PLACEMENT(chains, topology)
    for all chain ∈ chains do
        for i ← 1, len(chain) do        ▷ length of a chain is its number of VNFs
            if i == 0 then ▷ special treatment for the first chain because it doesn't have any predecessor
                for all n ∈ topology.nodes do
                    if hasEnoughResource(n) then
                        cost[(0, n)] ← cost(n)
            else
                for all n ∈ topology.nodes do
                    for all k ∈ topology.nodes do
                        if hasEnoughResourceOnState(i − 1, k, n) then ▷ decide on cost based on previous node
                            cost ← costOnState(i − 1, k, n)
                            if cost ≤ min then
                                min = cost
                    cost[(i, n)] ← min
```

Algorithm 2 MASMAN Algorithm - 2

```
function MASMAN__MANAGER__PLACEMENT(chains, topology)
    for all chain ∈ chains do
        for all n ∈ topology.nodes do
            if canBeManager(chain, n) then        ▷ node n must have connections to all chain's nodes besides required resources
                chain.manager = n
                reserveManagementResources(n)
    for i ← 0, maxIterations do
        n1 = randomNode(topology)
        n2 = randomNode(topology)
        updateTabuTable(n1, n2)
        if isMergePossible(n1, n2) then
            for all chain ∈ chains do
                if chain.manager == n1 then
                    chain.manager = n2
                    freeManagementResources(n1)
                    reserveManagementResources(n2)
                    updateMinCost()
```

stage at the end for placing VNFM we will use as another algorithm in the following section to compare with 1 algorithm.

6. Evaluation and Numerical Results

a) Simulation settings: Topologies and all other parameters used in simulation it is better to use a table

The algorithms which are simulated

Parameters used for evaluation

b) A subsection per parameter

Emphasize on achievements

6.1. Joint vs Disjoint

We have discuss about joint and disjoint solutions, in the following section we want to compare these solutions in the optimal form with eachother. For this comparison we use two different topology to have better insights and fair results. We have changed number chains in each step and these chains are generated randomly with configuration mentioned in **??**.

Table 9: Chain Configuration

Property	Value
Length	$[4, 7)$
Per Instance Cost	100$
Bandwidth	250bps

6.1.1. FatTree

Here we will use FatTree topology with k equals to 6. Each result is baed on 15 runs in the CPLEX with the limited time and optimality gap. As you can see results confirm our hypothesis that joint solutions makes better revenues.

Table 10: Revenues from Joint and Disjoint Solutions on FatTree Topology with k equals to 6

	joint	disjoint	nchains
0	4453.333333	1913.333333	10
1	6586.666667	1000.000000	15
2	8886.666667	1593.333333	20
3	11133.333333	3266.666667	25
4	13326.666667	7906.666667	30
5	15466.666667	13073.333333	35
6	17973.333333	14686.666667	40
7	20146.666667	17553.333333	45
8	22440.000000	15633.333333	50
9	24446.666667	14613.333333	55
10	26940.000000	14713.333333	60
11	29313.333333	18886.666667	65
12	31486.666667	16313.333333	70
13	33713.333333	17340.000000	75
14	36253.333333	19553.333333	80
15	37846.666667	18540.000000	85
16	40500.000000	23360.000000	90

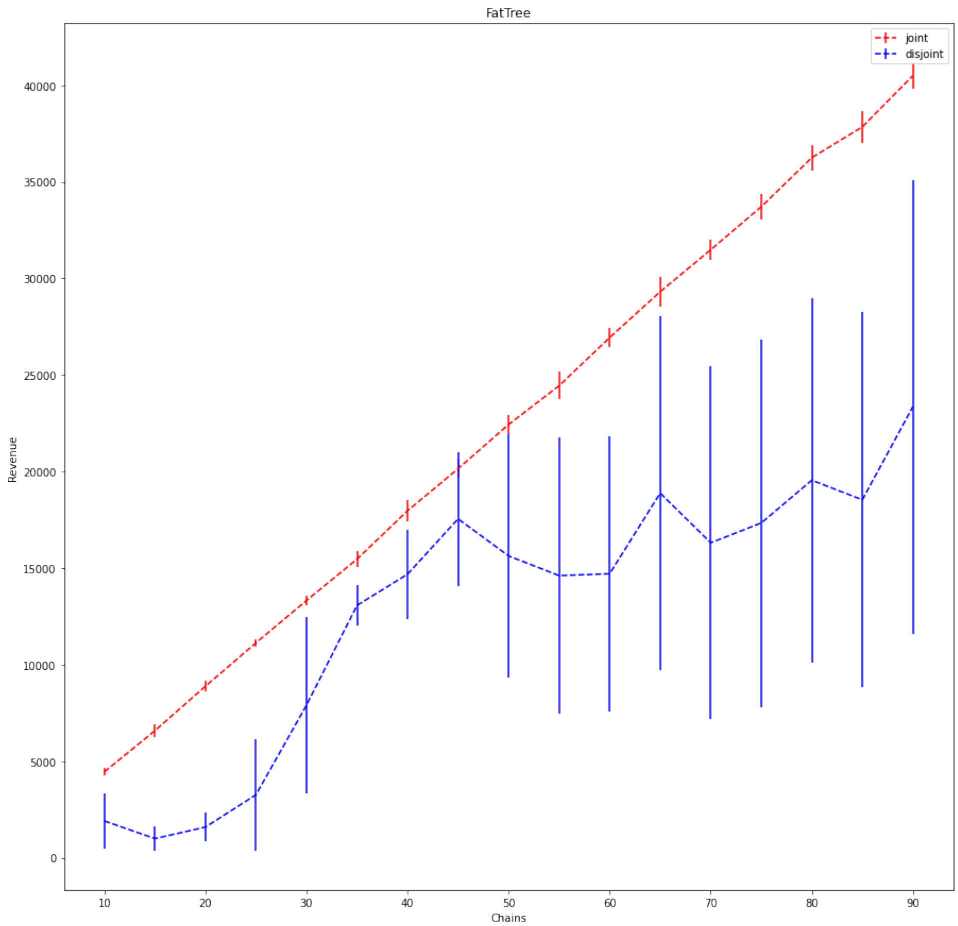


Figure 4: Revenues from Joint and Disjoint Solutions on FatTree Topology with k equals to 6

6.2. USNet

Here we will use USNet topology with 3 to 4 nodes attached to each of its points. As you can see the results show joint and disjoint solutions on this setup work equally because there is no specific management requirement and topology can handle all chains.

Table 11: Revenues from Joint and Disjoint Solutions on USNet Topology

	joint	disjoint	nchains
0	23000.0	23010.0	50
1	34180.0	34260.0	75
2	45430.0	45560.0	100
3	57180.0	57380.0	125
4	68750.0	69360.0	150

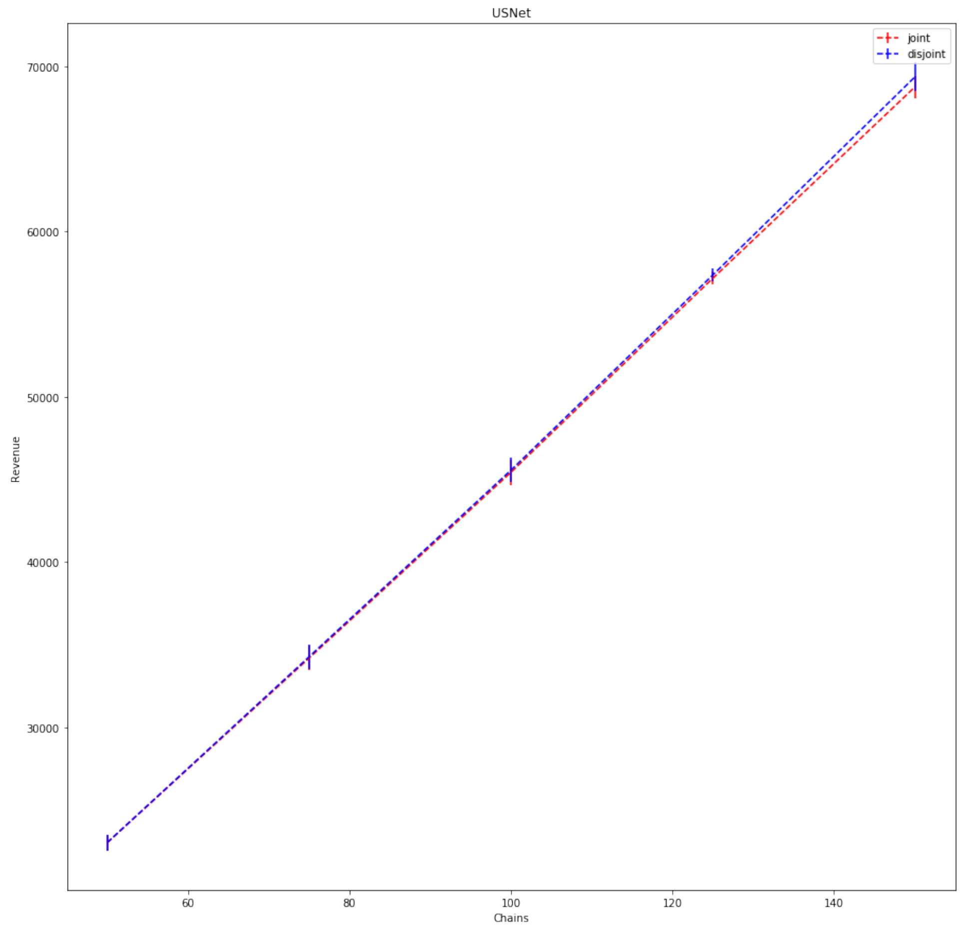


Figure 5: Revenues from Joint and Disjoint Solutions on USNet Topology

But we continue by adding an specific management constriant which is the distance between manager and chain's nodes must be less than equal to 4 and then the difference increases.

Table 12: Revenues from Joint and Disjoint Solutions on USNet Topology with Managemnt Constraint

	joint	disjoint	nchains
0	4473.333333	1420.000000	10
1	6926.666667	2733.333333	15
2	8940.000000	3533.333333	20
3	11133.333333	4833.333333	25
4	13313.333333	9020.000000	30
5	15486.666667	14693.333333	35
6	17913.333333	15433.333333	40
7	19813.333333	17586.666667	45
8	22213.333333	21020.000000	50
9	24113.333333	21866.666667	55
10	26640.000000	24546.666667	60
11	28833.333333	25760.000000	65
12	31360.000000	29420.000000	70
13	33140.000000	29660.000000	75
14	35686.666667	33320.000000	80
15	37600.000000	34986.666667	85
16	39840.000000	37473.333333	90

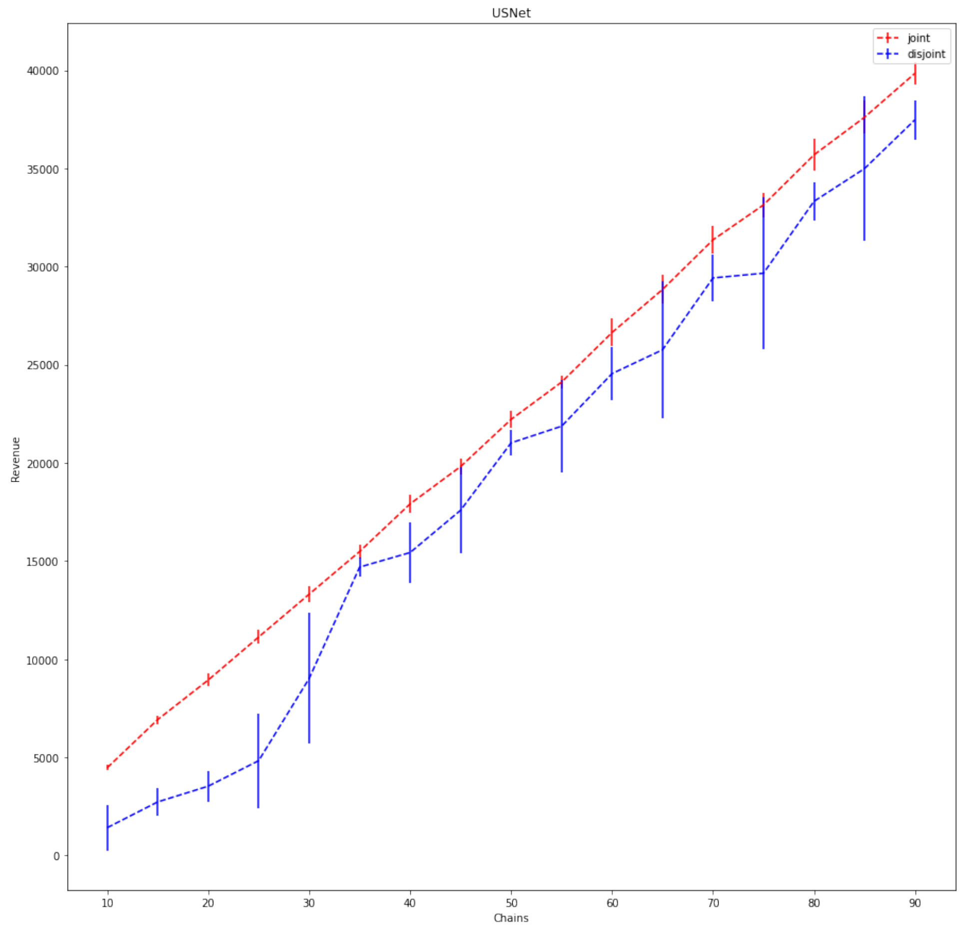


Figure 6: Revenues from Joint and Disjoint Solutions on USNet Topology with Management Constraint

7. Conclusion and Future Work

a) Review of what we have done

b) What is the future step

References

[1] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, On orchestrating virtual network functions, in: 2015 11th International Conference on Network and Service Management (CNSM), IEEE, 2015. `doi:10.1109/cnsm.2015.7367338`.
URL `https://doi.org/10.1109/cnsm.2015.7367338`

[2] M. Abu-Lebdeh, D. Naboulsi, R. Glitho, C. W. Tchouati, On the placement of VNF managers in large-scale and distributed NFV systems, IEEE Transactions on Network and Service Management 14 (4) (2017) 875–889. `doi:10.1109/tnsm.2017.2730199`.
URL `https://doi.org/10.1109/tnsm.2017.2730199`

[3] J. G. Herrera, J. F. Botero, Resource allocation in NFV: A comprehensive survey, IEEE Transactions on Network and Service Management 13 (3) (2016) 518–532. `doi:10.1109/tnsm.2016.2598420`.
URL `https://doi.org/10.1109/tnsm.2016.2598420`

[4] A. Laghrissi, T. Taleb, A survey on the placement of virtual resources and virtual network functions, IEEE Communications Surveys & Tutorials 21 (2) (2019) 1409–1434. `doi:10.1109/comst.2018.2884835`.
URL `https://doi.org/10.1109/comst.2018.2884835`

[5] D. Bhamare, R. Jain, M. Samaka, A. Erbad, A survey on service function chaining, Journal of Network and Computer Applications 75 (2016) 138–155. `doi:10.1016/j.jnca.2016.09.001`.
URL `https://doi.org/10.1016/j.jnca.2016.09.001`

[6] Y. Xie, Z. Liu, S. Wang, Y. Wang, Service function chaining resource allocation: A survey (2016). `arXiv:arXiv:1608.00095`.

[7] V. Eramo, A. Tosti, E. Miucci, Server resource dimensioning and routing of service function chain in NFV network architectures, Journal of Electrical and Computer Engineering 2016 (2016) 1–12. `doi:10.1155/2016/7139852`.
URL `https://doi.org/10.1155/2016/7139852`

[8] V. Eramo, E. Miucci, M. Ammar, F. G. Lavacca, 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) (2017) 2008–2025. `doi:10.1109/tnet.2017.2668470`.
URL `https://doi.org/10.1109/tnet.2017.2668470`

[9] M. A. T. Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, B. H. Khalaj, vSPACE: VNF simultaneous placement, admission control and embedding, IEEE Journal on Selected Areas in Communications 36 (3) (2018) 542–557. `doi:10.1109/jsac.2018.2815318`.
URL `https://doi.org/10.1109/jsac.2018.2815318`

[10] M. Huang, W. Liang, Y. Ma, S. Guo, Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement, IEEE Transactions on Cloud Computing (2019) 1–1`doi:10.1109/tcc.2019.2915835`.
URL `https://doi.org/10.1109/tcc.2019.2915835`

[11] M. Bouet, J. Leguay, T. Combe, V. Conan, Cost-based placement of vDPI functions in NFV infrastructures, International Journal of Network Management 25 (6) (2015) 490–506. doi:10.1002/nem.1920.
URL https://doi.org/10.1002/nem.1920

[12] C. Pham, N. H. Tran, S. Ren, W. Saad, C. S. Hong, Traffic-aware and energy-efficient vNF placement for service chaining: Joint sampling and matching approach, IEEE Transactions on Services Computing 13 (1) (2020) 172–185. doi:10.1109/tsc.2017.2671867.
URL https://doi.org/10.1109/tsc.2017.2671867

[13] V. Eramo, M. Ammar, F. G. Lavacca, Migration energy aware reconfigurations of virtual network function instances in NFV architectures, IEEE Access 5 (2017) 4927–4938. doi:10.1109/access.2017.2685437.
URL https://doi.org/10.1109/access.2017.2685437

[14] B. Farkiani, B. Bakhshi, S. A. MirHassani, A fast near-optimal approach for energy-aware SFC deployment, IEEE Transactions on Network and Service Management 16 (4) (2019) 1360–1373. doi:10.1109/tnsm.2019.2944023.
URL https://doi.org/10.1109/tnsm.2019.2944023

[15] K. Zheng, X. Wang, J. Liu, Distributed traffic flow consolidation for power efficiency of large-scale data center network, IEEE Transactions on Cloud Computing (2020) 1–1doi:10.1109/tcc.2020.2970403.
URL https://doi.org/10.1109/tcc.2020.2970403

[16] R. Yu, G. Xue, X. Zhang, QoS-aware and reliable traffic steering for

service function chaining in mobile networks, IEEE Journal on Selected Areas in Communications 35 (11) (2017) 2522–2531. doi:10.1109/jsac.2017.2760158.
URL https://doi.org/10.1109/jsac.2017.2760158

[17] Y.-J. Chen, L.-C. Wang, F.-Y. Lin, B.-S. P. Lin, Deterministic quality of service guarantee for dynamic service chaining in software defined networking, IEEE Transactions on Network and Service Management 14 (4) (2017) 991–1002. doi:10.1109/tnsm.2017.2758328.
URL https://doi.org/10.1109/tnsm.2017.2758328

[18] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, J. Rao, Dynamic flow migration for embedded services in SDN/NFV-enabled 5g core networks, IEEE Transactions on Communications 68 (4) (2020) 2394–2408. doi:10.1109/tcomm.2020.2968907.
URL https://doi.org/10.1109/tcomm.2020.2968907

[19] Z. Luo, C. Wu, An online algorithm for VNF service chain scaling in datacenters, IEEE/ACM Transactions on Networking 28 (3) (2020) 1061–1073. doi:10.1109/tnet.2020.2979263.
URL https://doi.org/10.1109/tnet.2020.2979263

[20] F. Yaghoubpour, B. Bakhshi, Optimal delay-aware service function chaining in NFV, in: 2019 27th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2019. doi:10.1109/iraniancee.2019.8786444.
URL https://doi.org/10.1109/iraniancee.2019.8786444

[21] P. Torkamandi, S. Khorsandi, B. Bakhshi, Availability aware SFC embedding in NFV: A clustering approach, in: 2019 27th Iranian Conference on Electrical Engineering (ICEE), IEEE, 2019. doi:10.1109/iraniancee.2019.8786672.
URL https://doi.org/10.1109/iraniancee.2019.8786672

[22] H. R. Khezri, P. A. Moghadam, M. K. Farshbafan, V. Shah-Mansouri, H. Kebriaei, D. Niyato, Deep reinforcement learning for dynamic reliability aware NFV-based service provisioning, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019. doi:10.1109/globecom38437.2019.9013214.
URL https://doi.org/10.1109/globecom38437.2019.9013214

[23] M. Abu-Lebdeh, D. Naboulsi, R. Glitho, C. W. Tchouati, NFV orchestrator placement for geo-distributed systems, in: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), IEEE, 2017. doi:10.1109/nca.2017.8171391.
URL https://doi.org/10.1109/nca.2017.8171391

[24] M.-J. Chiang, L.-H. Yen, Distributed approach to adaptive VNF manager placement problem, in: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2019. doi:10.23919/apnoms.2019.8893005.
URL https://doi.org/10.23919/apnoms.2019.8893005

[25] W. Ma, J. Beltran, Z. Pan, D. Pan, N. Pissinou, SDN-based traffic aware placement of NFV middleboxes, IEEE Transactions on Network and Service Management 14 (3) (2017) 528–542. doi:10.1109/tnsm.

2017.2729506.
URL https://doi.org/10.1109/tnsm.2017.2729506

[26] J. Liu, W. Lu, F. Zhou, P. Lu, Z. Zhu, On dynamic service function chain deployment and readjustment, IEEE Transactions on Network and Service Management 14 (3) (2017) 543–553. doi:10.1109/tnsm.2017.2711610.
URL https://doi.org/10.1109/tnsm.2017.2711610

[27] A. Gupta, B. Jaumard, M. Tornatore, B. Mukherjee, A scalable approach for service chain (sc) mapping with multiple sc instances in a wide-area network (2017). arXiv:arXiv:1709.04772, doi:10.1109/JSAC.2018.2815298.

[28] M. Ghaznavi, N. Shahriar, S. Kamali, R. Ahmed, R. Boutaba, Distributed service function chaining, IEEE Journal on Selected Areas in Communications 35 (11) (2017) 2479–2489. doi:10.1109/jsac.2017.2760178.
URL https://doi.org/10.1109/jsac.2017.2760178

[29] O. Alhussein, W. Zhuang, Robust online composition, routing and NF placement for NFV-enabled services, IEEE Journal on Selected Areas in Communications 38 (6) (2020) 1089–1101. doi:10.1109/jsac.2020.2986612.
URL https://doi.org/10.1109/jsac.2020.2986612