ABSTRACT  Network Function Virtualization (NFV) can support customized on-demand network services with flexibility and cost-efficiency. Virtual Network Function (VNF) instances need to be scaled out, scaled in, and reallocated across the NFV infrastructure (NFVI) to avoid a violation of service agreements when the demand traffic changes. However, selecting the new placement of VNFs for migrating a service function chain (SFC) is an issue of efficient NFV control. We propose two novel integer linear programming (ILP) models and two approximation algorithms for SFC placement and migration to maximize the cost-efficiency of an NFV network regarding the changes of service demands and dynamic routing. The ILP models allow us to obtain the optimal solutions of SFC placement and migration with explicit dynamic paths. The approximation migration results provided by our proposed heuristic and reinforcement learning algorithms are close to the optimal solution. Evaluation results carried out with real datasets and synthetic network topologies provide a helpful suggestion of a migration strategy for an NFV service provider to optimize the operating cost of an NFV network in the long term.

INDEX TERMS  NFV, SFC, service migration, optimization, machine learning.

I. INTRODUCTION

The 6G network will support customized on-demand network services, which require a flexible network architecture. An enabling technology is Network Function Virtualization (NFV) that replaces a physical appliance of a particular network function (e.g., load balancing, network address translation (NAT), routing) with an instantiable software component referred to as a Virtual Network Function (VNF). A VNF instance operates on a virtual machine hosted in a commercial-off-the-shelf server and geographically dispersed in an NFV infrastructure (NFVI). In NFV, a network service comprises a series of connected VNFs to create a service function chain (SFC). When the volume of demand traffic changes, some nodes in NFVI are possibly overloaded, leading to a quality of services (QoS) violation. An NFV service provider (NSP) should reconfigure the SFC placement and resource allocation to avoid the QoS violation caused by an imbalanced load. Different migration strategies of SFC can cause different operating costs and performance of an NFV network. Hence, it is crucial to optimize SFC migration for enabling customized on-demand network services with flexibility and cost-efficiency in the 6G network.

The issue of efficient migration for the NFV flexibility and cost-efficiency has been specified in technical documents of The European Telecommunications Standards Institute (ETSI), and Internet Research Task Force (IRTF) [1], [2]. One of the main challenges of SFC migration is the need for jointly optimizing the new placement of VNFs and SFC routing in NFVI, where a VNF instance of an SFC is dispersed and chained in sequence. Previous researches on service migration use diverse techniques, including optimization methods and approximation algorithms, to enhance cost-efficiency in a particular application context with different assumptions [3]–[8]. However, all existing SFC migration approaches have not considered explicit SFC routing regarding network dynamics. Most previous work assumes a set of precomputed paths and does not produce an explicit SFC routing path on a physical NFV infrastructure, which leads to a suboptimal solution. Our work is designed to fill the gap by considering explicit dynamic routing paths when finding the optimal migration solution.

Our work aims to optimize SFC migration, considering both the new placement of VNF and explicit paths of SFC...
to maximize cost-efficiency under available system resources and a QoS requirement. The challenging questions are the following: What is the optimal migration of SFC when the volume of demand traffic changes? When considering the dynamic routing, what is an explicit path of SFC on NFVI in the optimal migration? What is the size of an NFV network in which we can find the optimal SFC migration solution with acceptable computation time? What efficient approximation algorithm can quickly react to the demand changes in an extensive NFV network? We address those issues as a vital component of a dynamic NFV network.

This paper has the following contributions:

- We propose two integer linear programming (ILP) based optimization models, called FMO and FPO, for the SFC migration and placement problems. The FMO model provides the optimal reallocation of new explicit SFC placement and routing for a service demand within a delay requirement of SFC when the change of demand traffic occurs. We derive the FPO model from FMO to find the optimal SFC placement for the initial deployment of SFC.

- We introduce a heuristic algorithm (i.e., FMH) and a Reinforcement Learning (RL) based algorithm (i.e., FMR) to solve the SFC migration problem in an extensive NFV system. The migration result produced by FMH and FMR is close to the optimal solution. Compared to FMH and FMO, FMR can provide an SFC migration result with a similar cost and significantly reduced time when we do not retrain the reinforcement learning model.

- We evaluate our proposed optimization models (FMO, FPO) and approximation algorithms (FMH, FMR) in two real datasets and six synthetic scenarios. The evaluation results suggest a selection strategy of an appropriate objective for an NSP to minimize the total operating cost of an NFV network regarding SFC migration.

The rest of the paper is arranged as follows. We review the related work in Section II. Section III presents the system and the optimization problem of SFC placement and migration with explicit SFC routing. Section IV describes the FMO model that provides the optimal SFC migration, including VNF placement and explicit dynamic routing. Section V presents two approximation algorithms based on heuristics (FMH) and reinforcement learning (FMR) for SFC migration.

In Section VI, we describe the evaluation of our proposed model and algorithms for the SFC placement and migration problems. We summarize the paper and present several future directions in Section VII.

II. RELATED WORK

Many service migration solutions have been proposed for cloud computing to improve the operating cost (e.g., [9]–[11]). Existing service migration solutions in a cloud system cannot be applied to an NFV network due to several fundamental differences in the NFV architectural framework, such as service function chaining and network service requirements. In [12], Gonzalez et al. carried out an experiment on NFV orchestration to identify NFV challenges in VNF migration. Some research efforts have considered the migration problem in NFV to improve the system performance and operating cost [3]–[8]. We can roughly put solution approaches into three groups: optimal, heuristic, and learning-based models.

Several studies proposed a heuristic algorithm for the migration problem in different scenarios and various objectives [3]–[8]. For example, Wang et al. proposed a heuristic algorithm that decouples the state transfer and packets migrations for NFV [3]. Huang et al. proposed a heuristic algorithm for maximizing the network throughput in the VNF migration problem [4]. In [5]–[8], [13], [14], the authors moved a step forward as they obtained both optimal and approximation results for the migration problem. In [5], Eramo et al. solved the VNF migration problem considering QoS degradation in an NFV network with a given set of physical network paths. In [6], Sun et al. built a flow migration controller to move traffic flows across network function instances. They assumed that a traffic flow could be migrated between any pair of nodes. In [7], Qu et al. proposed a mixed-integer quadratically constrained programming model to solve the flow migration problem with delay constraints in 5G core networks. However, they did not consider a physical path for a logical link in NFVI. In [8], Hejja et al. developed further Eramo et al.’s paper and presented a resource allocation algorithm with migration and delay constraints for minimizing the power consumption in an NFV network. They assumed that the NFV orchestrator already has the list of all paths between any pair of nodes in the physical network.

Recently, Yi et al. designed a VNF migration mechanism for a QoS guarantee [13]. Eramo et al. proposed a reconfiguration cost-aware policy in a scenario of multiple NFV providers [14]. They considered several essential realistic parameters, such as the modulation system in an elastic optical network that interconnects the NFV providers. However, these and other results in the literature for the migration problem in NFVs have not considered explicit dynamic routing paths when finding a migration policy. As a routing path in a dynamic network changes over time, developing a migration solution that can overcome such restrictions in previous research is strongly desired.

A few studies have been used machine learning as an approximation approach to the migration problem with various assumptions. [15]–[17]. In [15], Pei et al. presented a VNF placement algorithm based on a deep Q-Network. In [16], Sun et al. proposed a flow migration algorithm based on a Graph Neural Network to minimize the migration cost. In [17], Chen et al. used a deep reinforcement learning approach to propose an SFC migration algorithm for improving resource utilization. The authors assumed a mapping between an SFC link and a physical link. None of the prior studies analyze the effectiveness of an approximation model based on machine learning compared to the optimal result.
Our work is different from existing studies as we address the SFC migration problem with explicit paths on NFVI regarding dynamic routing for delay-guarantee SFC. We provide an optimization model, a heuristic algorithm, and a reinforcement learning-based algorithm for finding an SFC migration solution with explicit SFC routing in the physical infrastructure of NFV. We demonstrate the effectiveness of our proposed models and algorithms by extensive evaluations using both real datasets and synthetic topologies.

III. SYSTEM DESCRIPTION

The ETSI NFV framework is composed of the NFV infrastructure (NFVI), the VNFs, and the management and orchestration of NFVs (MANO) [1]. The NFVI resources used by VNF instances can be shared by more than one SFC, which is managed by MANO [18]. For example, the MANO element can update the state and data of a VNF instance that another SFC can reuse. The MANO also processes resource scaling to optimize resource utilization. In the SFC migration problem, we assume that the VNF sharing between SFCs is processed by MANO and do not explicitly incorporate it into the problem formulation. An SFC migration solution can be integrated into MANO to manage SFC migration automatically for service demands.

We model an NFVI as a directed graph $G$ composed of the node set $V$ and the link set $E$. To approach the SFC placement and migration problems, we define the following:

- $r_n^v$ is the computing capacity of node $v$, which is the number of CPU cores.
- $r_n^e$ is the bandwidth capacity of link $e$.
- $F$ is the set of VNF types.
- $\eta_f$ is the number of CPU cores needed to process one traffic unit of VNF type $f$.
- $\kappa_v$ is the cost for providing one core at node $v$.
- $\beta_v$ is the routing delay that is the time period required by node $v$ to direct one traffic unit.
- $\mu_{vf}$ is the processing delay that is the time period needed by node $v$ to provide VNF $f$.
- $\Omega = \{S_i\}$ is the set of SFCs provided by the NFV system.
- $S_i = (f_{i1}, \ldots, f_{ij}, \ldots, f_{in})$ is an SFC. $f_{ij}$ is the $j$th VNF of SFC $S_i$.
- $\Gamma = \{d\}$ is the set of service demands. The parameters of demand $d$ are as follows: $s_d$ is the arrival node, $t_d$ is the departure node, $S_d \in \Omega$ is the SFC, $\alpha_d$ is the SFC delay, and $b_d$ is the bandwidth volume.
- $i_e$ and $j_e$ is the starting node and ending node of link $e$.
- $w = (w_e)$ is the weight vector of NFVI where $w_e$ is the weight of link $e$.

An NFV network initially selects a proper routing path and the placement of VNFs on the path for an SFC of a service demand, which is mentioned as the SFC placement (FP) problem. When traffic changes occur, the MANO modifies the location of some VNF instances and SFC paths for satisfying the service requirements and optimizing resource utilization. We call it the SFC migration (FM) problem. Note that the FM problem is different from the FP problem as we need to consider the current SFC placement when finding the new optimal SFC placement and routing solution. The FM problem is also required to be solved more frequently than the FP problem during the system operation. Optimizing SFC placement and migration could significantly impact the cost efficiency and performance of an NFV network. These problems are stated as follows:

Problem 1 (SFC Placement (FP)): Given an NFV network $G$, the service demand set $\Gamma$, find an SFC placement strategy and its paths, considering dynamic routing to
minimize the deployment cost under constraints on service function chaining, SFC delay, and the available system resource.

**Problem 2 (SFC Migration (FM))**: Given an NFV network $G$, the service demand set $\Gamma$, a current SFC placement matrix $\pi$, find an SFC migration strategy and its paths, considering dynamic routing to minimize the reallocation cost when the bandwidth demand changes under constraints on service functions chaining, SFC delay, and the available system resource.

Fig. 1 illustrates a simple example of our proposed solutions that consider dynamic routing paths when finding the optimal migration solution. In the NFVI shown in the figure, nodes $v_1$, $v_3$ and $v_6$ are not equipped with resources for VNFs. The computing capacity of nodes $v_2$ and $v_4$ for VNFs is 30 CPU cores. Node $v_3$ has 15 cores. The link bandwidth capacity is 80 Gbps. The cost for providing one core at nodes $v_2$, $v_4$ and $v_5$ is 3, 1 and 2 $/h$, respectively. Service demands 1 and 2 with arrival node $v_1$ and departure $v_6$ request an SFC composed of the firewall (FW) and NAT functions. The number of cores required by the FW and NAT functions are two and one core per one Gbps. The bandwidth volume requested by service demands 1 and 2 is 5 and 10 Gbps in the first time slot. The bandwidth volume requested by service demand 1 increases to 10 Gbps in the second time slot.

Fig. 1a shows the optimal SFC placement and routing for the two service demands in the first time slot. Fig. 1b shows the migration solution for optimizing the reallocation cost in the second time slot when considering two pre-computed paths ($v_1, v_2, v_3, v_6$) and ($v_1, v_4, v_5, v_6$). The FW function of service demand 2 is migrated from $v_5$ to $v_4$. The FW and NAT functions of service demands 1 are moved from $v_4$ and $v_5$ to $v_2$. The reallocation cost of the migration solution is 120 $. This simple example ignores the cost of moving functions between nodes when computing the reallocation cost. Fig. 1c shows the migration solution for optimizing the reallocation cost in the second time slot when considering dynamic routing paths. The NAT function of service demands 1 and 2 is moved from $v_5$ to $v_2$ and $v_4$. The reallocation cost in the case of dynamic routing paths is 110 $, which is better than that of pre-computed paths. It means that without considering dynamic routing paths, the migration solution is suboptimal, resulting in a high operating cost of an NFV system. Our proposed model is designed to overcome such a limitation.

**IV. OPTIMIZATION MODEL FOR SFC MIGRATION**

We first propose an ILP model to obtain the optimal migration result for the FM problem. We call it FMO. We then derived an ILP model from FMO to solve the FP problem, which we call FPO. The main variables of FMO are as follows:

- $\mathbf{m} = (m_{v'v'di})$ is the migration decision matrix. If the NFV system moves the $i$th VNF of $S_d$ from $v$ to $v'$, $m_{v'v'di} = 1$, otherwise, $m_{v'v'di} = 0$.

| TABLE 1. Summary of main notations. |
|-------------------------------------|
| **Input parameters**               |
| $G = (V, E)$: The NFVI graph where $V$ is the set of physical nodes and $E$ is the set of physical links. |
| $r^n_v$: The computing resource of node $v$. |
| $r^l_e$: The bandwidth capacity of link $e$. |
| $s_i$: The starting node of link $e$. |
| $t_e$: The ending node of link $e$. |
| $f$: The set of VNF types. |
| $\eta_f$: The number of cores required to process one traffic unit of VNF type $f$. |
| $\kappa_v$: The cost for providing one core at node $v$. |
| $\Omega$: The set of SFCs provided by the NFV system: $\Omega = \{S_i\}, S_i = \{f_{i1}, \ldots, f_{i\nu}, \ldots, f_{i\nu_i}\}$: $f_{ij}$ is the $j$th VNF of $S_i$. |
| $\Gamma = \{d\}$: The set of service demands. |
| $s_d$: The arrival node of demand $d$. |
| $t_d$: The departure node of demand $d$. |
| $b_d$: The bandwidth volume of demand $d$. |
| $\sigma_d$: The SFC delay of demand $d$. |
| $S_d$: The SFC of demand $d$. |
| $\beta_v$: The routing delay required by node $v$ to route one traffic unit. |
| $\mu_v$: The processing delay required by node $v$ to provide VNF $f$. |
| $\gamma_{v'v'di}$: The migration cost when the location of $f_{di}$ is changed from $v$ to $v'$. |
| $\rho_{v'v}$: The minimum-weight path from $v$ to $v'$. |
| $\varphi_f$: The size of the state and data of a VNF type $f$. |
| $\pi = (\pi_{v'di})$: The current SFC placement matrix: If node $v$ provides $f_{di}$, $\pi_{v'di} = 1$, otherwise $\pi_{v'di} = 0$. |
| $w = (w_e)$: The NFVI link weight system. $w_e$ is the weight of link $e$. |

| **Output variables**               |
|-------------------------------------|
| $\mathbf{m} = (m_{v'v'di})$: The migration decision matrix: If the NFV system migrates the $i$th VNF of SFC $S_d$ from $v$ to $v'$, $m_{v'v'di} = 1$, otherwise, $m_{v'v'di} = 0$. |
| $x = (x_{ed})$: The explicit routing matrix of service demands on NFVI: If demand $d$'s path contains link $e$, $x_{ed} = 1$, otherwise, $x_{ed} = 0$. |
| $z = (z_{v'di})$: The SFC placement matrix after migration: If node $v$ provides $f_{di}$, $z_{v'di} = 1$, otherwise, $z_{v'di} = 0$. |

- $\mathbf{x} = (x_{ed})$ is the explicit routing decision matrix of service demands on NFVI. If demand $d$ uses link $e$, $x_{ed} = 1$, otherwise, $x_{ed} = 0$. 
- $\mathbf{z} = (z_{v'di})$ is the SFC placement matrix after migration. If node $v$ provides $f_{di}$, $z_{v'di} = 1$, otherwise, $z_{v'di} = 0$.

We summarize the main notations of the NFVI system and the optimization models in Table 1.

**A. FEASIBLE ROUTING**

The flow conservation condition guarantees that there is one incoming link and one outgoing link used by a service demand at a middle node of the demand path. There is also one outgoing link and one incoming link realizing a service demand at its arrival node and its departure node, respectively. We define $\theta$ to be a large number. $r_{max}$ is the length of the path
between \(u\) and \(v\). The condition is given by:
\[
\sum_{(e,i=v)} x_{ed} - \sum_{(e,i=v)} x_{ed} = 0, \quad \forall d, \forall v, v \neq s_d, v \neq t_d, \tag{1}
\]
\[
\sum_{(e,i=s_d)} x_{ed} = 1, \quad \forall d, \tag{2}
\]
\[
\sum_{(e,i=t_d)} x_{ed} = 1, \quad \forall d. \tag{3}
\]
A feasible routing assures that a path realizing a service demand contains no cycles. The condition is given by:
\[
(x_{ed} - 1) \leq r_{ed} - w_e - r_{ed} \leq (1 - x_{ed}) \theta, \quad \forall d, \forall e. \tag{4}
\]

**B. SERVICE FUNCTION CHAIN**

Three conditions of service function chaining provided by the NFV are the completion, sequence, and delay constraints. First, the completion constraint guarantees that the NFV network allocates all VNFs belonging to SFC \(S_d\). The condition is given by:
\[
\sum_{v} z_{vdi} = 1, \quad \forall d, \forall i, \tag{5}
\]
\[
z_{vdi} \leq \sum_{(e,i=v \text{ or } j,v)} x_{ed}, \quad \forall v, \forall d, \forall i. \tag{6}
\]
Constraint (5) ensures that the NFV system satisfies all SFCs requested by all demands. Constraint (6) guarantees that the NFV system only consider a node on demand \(d\)'s path to provide VNFs of SFC \(S_d\).

Second, the sequence constraint assures that all VNFs of \(S_d\) are joined in an ordered list on NFVI. We use two additional binary variables \(z^d_{vdi}\) and \(z^d_{edi}\) to express the constraint. If a node between \(s_d\) and \(v\) on demand \(d\)'s path allocates the \(f_{di}\), \(z^d_{vdi} = 1\), otherwise \(z^d_{vdi} = 0\). If a node between \(s_d\) and \(i_e\) on demand \(d\)'s path allocates \(f_{di}\) and the path contains link \(e\), \(z^d_{edi} = 1\), otherwise \(z^d_{edi} = 0\). The constraint is given by:
\[
z_{vdi} \leq z^d_{vdi}(i-1), \quad \forall e, \forall d, \forall i \geq 1, \tag{7}
\]
\[
z^d_{vdi} = z_{vdi} + \sum_{(e,i=v)} z^d_{edi}, \quad \forall e, \forall d, \forall i. \tag{8}
\]
\[
x_{ed} + z^d_{edi} - 1 \leq z^d_{edi} \leq x_{ed}, \quad \forall e, \forall d, \forall i. \tag{9}
\]
\[
z^d_{edi} \leq z^d_{edi} \leq 1 \tag{10}
\]
Condition (7) provides a node \(v\) supplies \(f_{di}\) if and only if one node on demand \(d\)'s path from \(s_d\) to \(v\) satisfies \(f_{di}(i-1)\). Condition (8) assures that \(z^d_{vdi} = 1\) if and only if a node on demand \(d\)'s path from \(s_d\) to \(v\) supplies \(f_{di}\). Constraints (9) and (10) guarantees that \(z^d_{edi} = 1\) if and only if demand \(d\)'s path contains link \(e\), and either \(i_e\) or its previous node on demand \(d\)'s path satisfies VNF \(f_{di}\).

Third, the delay constraint assures that the SFC delay must be fulfilled. We define the SFC delay of demand \(d\) to be the sum of the processing delay of all VNFs belonging to \(S_d\) and the routing delay at every node on demand \(d\)'s path. The condition is given by:
\[
\sum_{v} \beta_{v} x_{ed} b_{d} + \sum_{i} \mu_{v} f_{di} \sum_{i} z_{vdi} \leq \alpha_{d}, \quad \forall d. \tag{11}
\]

**C. RESOURCE AVAILABILITY**

The total traffic of service demands traveled through an NFVI link cannot be larger than its bandwidth capacity. The constraint is as follows:
\[
\sum_{d} b_{d} x_{ed} \leq c_{e}, \quad \forall e. \tag{12}
\]

The total cores that a node gives to all demands cannot be larger than the node capacity. The condition is given by:
\[
\sum_{v, d, i} b_{d} z_{vdi} \eta_{v} f_{di} \leq \rho_{v}, \quad \forall v. \tag{13}
\]

**D. FEASIBLE MIGRATION**

A feasible migration assures that the \(ith\) VNF of demand \(d\) (i.e., \(f_{di}\)) moves from node \(v\) to node \(v'\) if and only if node \(v\) provides \(f_{di}\) in the current SFC placement strategy and node \(v\) provides \(f_{di}\) in the new SFC placement strategy. We denote by \(\pi = \pi_{vdi}\) the current SFC allocation for the service demand set. If node \(v\) provides VNF \(f_{di}\), \(\pi_{vdi} = 1\), otherwise \(\pi_{vdi} = 0\). The condition is given by:
\[
m_{v'} f_{di} \leq \pi_{vdi}, \quad \forall v, \forall v', \forall d, \forall i. \tag{14}
\]
\[
m_{v} f_{di} \leq \pi_{vdi}, \quad \forall v, \forall v', \forall d, \forall i. \tag{15}
\]
\[
\pi_{vdi} + z v_{di} - 1 \leq m_{v} f_{di}, \quad \forall v, \forall v', \forall d, \forall i. \tag{16}
\]

**E. OBJECTIVE FUNCTION**

We denote by \(\gamma_{v} f_{di}\) the migration cost when migrating VNF \(f_{di}\) from \(v\) to \(v'\). We assume that the NFV system uses the minimum-weight path to move the data and state of a VNFI instance its new placement. We denote by \(\rho_{v}\) the cost of the path from \(v\) to \(v'\). \(\gamma_{v} f_{di}\) is the size of the data and state of VNF type \(f\). The migration cost of a VNF instance is as follows:
\[
\gamma_{v} f_{di} = \rho_{v}. \tag{17}
\]

The migration cost of a migration solution is as follows:
\[
U_{m1} = \sum_{v, v', d, i} m_{v} f_{di} \gamma_{v} f_{di}. \tag{18}
\]

The deployment cost of a new VNF placement is as follows:
\[
U_{m2} = \sum_{d, v, i} b_{d} z_{vdi} \eta_{v} f_{di} k_{v}. \tag{19}
\]

We define the reallocation cost to be the sum of the migration cost and the deployment cost, which is given by:
\[
U_{m3} = U_{m1} + U_{m2}. \tag{20}
\]

We use the objective function \(U_{m} = U_{m1}\) when minimizing the migration cost. The objective function \(U_{m} = U_{m2}\) minimizes the deployment cost for a new set of demand parameters. The objective function \(U_{m} = U_{m3}\) optimizes the total of the deployment cost and the migration cost.
F. ILP MODEL FOR SFC MIGRATION
The FM problem is to find an SFC migration strategy for minimizing a cost function under the dynamics of bandwidth demand and constraints on the VNF sequence, SFC delay and available system resource. The FMO model provides the optimal solution of SFC migration and its explicit paths on NFVI regarding dynamic routing for the FM problem with a given set of bandwidth demands. The model is as follows:

Minimize $U_m$ subject to (1) – (16).

G. ILP MODEL FOR SFC PLACEMENT
We derive the FPO model from the FMO model to find the optimal solution to the FP problem. We remove the constraints on a feasible migration solution (i.e., Eq. (14)-(16)), and compute the objective function for the FPO problem. The output of the FPO model is the VNF placement matrix $z = (z_{vdi})$ and the routing decision matrix $x = (x_{ed})$. The objective function is given by:

$$U_p = \sum_{d,v,i} b_dz_{vdi} \eta_d x_{dv}.$$  \hspace{1cm} (21)

The FPO model is as follows:

Minimize $U_p$ subject to (1) – (13).

V. APPROXIMATION ALGORITHMS FOR SFC MIGRATION
In the preceding section, we proposed the FPO and FMO models to obtain the optimal SFC placement and migration in an NFV network. For a scenario with 100 nodes and 1000 demands, the FPO and FMO have billions of variables. It takes several hours for an ILP solver to provide the optimal result of such a large ILP model. While we only need to solve the FP problem initially, the FM problem must be addressed frequently in the system’s running time. Hence, we develop approximation results based on heuristics and reinforcement learning for SFC migration in an extensive NFV network.

A. HEURISTIC ALGORITHM
Our proposed heuristic algorithm for SFC migration, namely FMH, is based on the Simulated Annealing (SA) [19]. SA starts with a feasible solution, then searches in the problem’s solution space to find the optimum. In order to apply SA to the FM problem, we need to define the formulation of a migration solution and design a searching procedure for a neighbor solution. We denote an SFC migration solution by $O_m = \{(d, i, v, v') : d \in D, i \in S_d, v \in V, v' \in V\}$, where $v$ and $v'$ are old and new target NFVI nodes that provides the $i$th VNF of SFC $S_d$. We propose the Replace($d, i, v, v', v''$, $O_m$) function to move between neighbor solutions. In the function, we replace the target NFVI node $v'$ in a migration solution by $v''$. In the searching procedure, we use the Replace function for a random item (i.e., $(d, i, v, v')$) in the current migration solution and a random NFVI node $v''$.

We present the FMH algorithm in Algorithm 1. For each temperature $T$, FMH executes the For loop to search for a better migration solution. We denoted the initial value of $T$ by $T_0$ and the number of iterations of the For loop by $\phi$. We use the cooling function $C(T) = \tau T$ ($\tau \in (0, 1)$) to reduce the temperature $T$ after one iteration of the While loop. FMH finishes when $T$ is less than the stop temperature $T_s$. We can improve the quality of an approximation migration solution by selecting an appropriate number of iterations of the While loop and the For loop.

In the For loop, we run the searching procedure (i.e., lines 7-12) to select a neighbor solution. FMH moves to the neighbor solution if the objective value of the current solution is worse than that of the neighbor solution (i.e., lines 12-17). Otherwise, FMH selects the neighbor solution with a probability (i.e., lines 18-24). The purpose of moving to a worse solution is to cope with a local optimum. When the temperature $T$ reduces, the probability decreases. Hence, after some consecutive iterations, FMH will converge to the optimal migration solution.

The complexity of FMH is determined by the two nested loops (i.e., the While loop defined in line 5 and the

Algorithm 1 Heuristic Algorithm for SFC Migration

1: function FMH($G, \Gamma, \pi$)
2: Initialize $T$, $T_0$, $T_f$, \phi, $\tau$, $O_m$
3: $O_m \leftarrow O_m$
4: while $T \geq T_n$ do
5:  for $k \leftarrow 1$ to $\phi$ do
6:  repeat
7:   Select $(d, i, v, v')$ in $O_m$ randomly
8:   Select $v''$ in $V$ randomly
9:   $O_m \leftarrow$ Replace$(d, i, v, v', v'', O_m)$
10:  Compute $x'$
11:  until $O'_m$ is feasible
12:  if $U_m(O_m) > U_m(O'_m)$ then
13:    $O_m \leftarrow O'_m$
14:  else
15:    $\Delta \leftarrow U_m(O_m) - U_m(O_m)$
16:   $\varepsilon \leftarrow$ Random(0,1)
17:  if $\exp(-\Delta/T) > \varepsilon$ then
18:    $O_m \leftarrow O'_m$
19:    $x' \leftarrow x'$
20:  end if
21:  end if
22: end for
23: $T \leftarrow C(T)$
24: end while
25: return $m$, $x^*$
26: end function
We propose a reinforcement learning based approximation algorithm for the FM problem, called FMR. We use the Soft Actor-Critic (SAC) algorithm, which is an actor-critic method of reinforcement learning [20]. The key idea of SAC is to use the policy’s entropy to increase the training speed. We first describe the formulation of SAC, then apply it to the FM problem.

A Markov decision process formulates the SAC algorithm with the state space, action space, probability density, and reward function. Each state in the state space can change to another state by an action in the action space with a probability defined by the probability density function. Let \( a_t \) be an action in the action space \( A \), \( s_t \) be a state in the state space \( S \). The goal of SAC is to search for a policy \( \omega(s_t, a_t) \) for maximizing the learning objective. We define the learning objective to be the expected sum of rewards and the policy’s entropy. We denote by \( \lambda \) the temperature hyperparameter of SAC, which is used to control the relationship between the entropy and the reward. Let \( h \) be the entropy function. The learning objective of SAC is given by:

\[
J(\omega) = \sum_t \mathbb{E}_{(s_t, a_t)} [r(s_t, a_t) + h(\omega(\cdot | s_t))\lambda] \tag{22}
\]

To apply SAC to the FM problem, we define the following:

- A state in FMR is a resource allocation for service demands. We denote a state by \( s_t = ((d, f, v, b_d)) \). The formulation shows that VNF \( f \) of service demand \( d \) with bandwidth \( b_d \) is provided by node \( v \).
- An action in FMR represents a transition between different states or a possible migration result. We denote an action by \( a_t = (v_1, v_2, \ldots, v_{|s_t|}) \). The formulation shows that \( v_i \in V \) is a migration solution for the \( i \)th item in the current state.
- We define the reward of a migration solution as \( U_{m}^t = -U_{m} \) since SAC maximizes the learning objective. As a result, the learning policy produced by FMR minimizes the reallocation cost.

We describe the outline of FMR in Algorithm 2. The main steps of FMR are presented in Algorithm 2. We use the actor network’s policy to compute a migration action with regard to the current state of SFC allocation for service demands. We move to the new state of SFC allocation by executing the action in the NFV environment. We use the critic network with regard to the new state of SFC allocation, its reward and the preceding state to evaluate the value of the new state, which is used to optimize the parameters of the actor and critic networks. We implement the actor and critic networks as neural networks.

**B. REINFORCEMENT LEARNING BASED APPROXIMATION ALGORITHM**

We propose a reinforcement learning based approximation algorithm for the FM problem, called FMR. We use the Soft Actor-Critic (SAC) algorithm, which is an actor-critic method of reinforcement learning [20]. The key idea of SAC is to use the policy’s entropy to increase the training speed. We first describe the formulation of SAC, then apply it to the FM problem.

For loop defined in line 6). The While loop executes until the temperature is greater than or equal to the stop temperature \( T_{n} \). When we use the cooling function \( C(T) = \tau T \), the While loop runs in \( O \left( \log \left( \frac{T_{0}}{T_n} \right) \right) \), where \( T_{0} \) is the start temperature. The average time of the worst case, in which the migration solution does not change after the repeat until loop (i.e., lines 7 – 12), is \( \frac{|\Gamma||F||V|^2}{2} \) because the number of all possible random results is \( |\Gamma||F||V|^2 \). Thus, FMR has a complexity of \( O \left( \log \left( \frac{T_{0}}{T_n} \right) \right) \).

**VI. EVALUATION**

We evaluate the performance of our proposed solution approaches, including FPO, FMO, FMH, and FMR for SFC migration with explicit paths in NFV. Our proposed ILP model (FMO) allows us to find the optimal SFC migration for a set of service demands. To evaluate the performance of our proposed approximation algorithms (FMI, FMR), we use the optimal solution obtained by FMO as a benchmark solution since we could not find an approximation algorithm in previous studies for the SFC migration problem with explicit dynamic paths and a similar set of input and output parameters. We also discuss NSP’s strategies for optimizing the operating cost in the long term.

**A. SCENARIOS AND PARAMETERS SETTING**

Our evaluation includes eight network topologies: Two realistic networks reported in the Abilene dataset [21] and the Geant dataset [22], and six synthetic networks generated the
FIGURE 3. The convergence of FMR.

Barabási-Albert (BA), Waxman (WA) and Erdős-Rényi (ER) models [23]. The Abilene dataset presents a US network of 12 nodes and 15 links. The Geant dataset describes a European network including 22 nodes and 36 links. We generate two topologies of 50 and 200 nodes for a random graph model by using the FNSS tool [24]. The parameters of creating a BA topology are four initial nodes and four links added to a new node. The parameter of generating a WA topology is the link density probability of 0.9. The parameter of creating an ER topology is the edge creation probability of 0.2. We denote the small and large topologies of BA, WA, and ER by BAS, BAL, WAS, WAL, ERS, and ERL, respectively.

We randomly select the arrival and departure nodes of each service demand. The number of service demands in the Abilene and Geant networks is 15, and that in the other topologies is 100. The volume of the bandwidth demand is varied between 1 and 5 Gbps. We randomly select an SFC for a service demand in a set of four VNF types. The number of cores required by a VNF type to process 1 Gbps is varied between 1 and 2 cores. The SFC delay is randomly chosen between 1 ms and 30 ms. The bandwidth volume of all links is 80 Gbps. The processing capacity of all nodes is 200 cores. The routing delay and processing delay parameters are varied from 10 to 100 µs. The cost for providing one core at a node is 1 $/h. The guidance of selecting the values of some parameters has been reported in several previous studies [5], [25].

We implement FMR by adapting the implementation of SAC (i.e., tf_agents.agents.sac) in TensorFlow [26]. We use the default implementation of the temperature hyperparameter, which is automatically adjusted [27]. The number of layers and the number of neurons in each layer used in the actor and critic networks are two layers and 32 neurons because the policy did not significantly improve when we increased the number of layers and neurons.

We wrote FMH in Java. In our evaluation, we use \( \phi = 100 \) and \( T_n = 0.1 \). We assign the parameter \( \tau \) so that the number of FMH’s iterations and FMR’s iterations is similar. The computation of \( \tau \) is given by:

\[
\tau = \left( \frac{T_n}{T_{FMR}} \right)^{\frac{1}{IFMR}}
\]

where \( IFMR \) is the number of iterations of FMR.

We evaluate three metrics, including the reallocation cost, the computation time, and the operating cost. The reallocation cost, expressed by (20), is the sum of the migration cost and the deployment cost (Section IV-E). We compute the migration cost and the deployment cost by (18) and (19). The computation time is the time required by a proposed model and algorithm to find a migration and placement solution in an evaluation scenario. We use the operating cost to evaluate the impact of different SFC migration strategies on the system in the long term. The operating cost of an NFV network in one day is the sum of the migration cost in all migration scenarios and the product of the duration time and the deployment cost in each migration scenario. The reallocation and operating cost unit is a monetary unit ($). Note that the value difference between the cost of various scenarios and solution approaches affects the results of performance comparison rather than their specific value.

B. PERFORMANCE COMPARISON OF SOLUTION APPROACHES

We first analyze the convergence of FMR. Fig. 3 plots the average reallocation cost of ten migration solutions produced
FIGURE 4. Performance comparison of different solution approaches: FPO is the optimal solution of the FP problem, FMO is the optimal solution of the FM problem, FMR is the results produced by FMR’s policy trained with new data, FMR* is the results produced by FMR’s policy without retraining the policy, FMH is the results of the FMH heuristic algorithm.

FIGURE 5. The FMR and FMH performance in an extensive NFV network.

by FMR’s policy that is obtained after some iterations of training. The result shows that FMR can obtain a steady policy after around 7.700 iterations. The number of iterations required for a steady policy slowly increases when the size of a topology grows. For example, it approximately takes 5,000 iterations in the Geant topology with 22 nodes, 5,500 iterations in the BAS topology with 50 nodes, and 7,200 iterations in the BAL topology with 200 nodes.

Second, we compare the computation time and reallocation cost of the optimization model, the approximation solutions based on heuristics and reinforcement learning. To compare with optimal results, we only consider the small topologies, including the Abilene, Geant, BAS, WAS, and ERS topologies. The number of FMR’s iteration that we used is 7,200 iterations. Fig. 4a plots the reallocation cost of FPO, FMO, FMR, and FMH. We observe that FMO is better than FMH and FMR in terms of the reallocation cost, but the difference is not significant. In Fig. 4b, the computation time of FMR and FMH is higher than that of FMO. FMH is slightly better than FMR in terms of the computation time. In Fig. 4, when we use FMR’s policy without retraining the policy with new data, FMR can produce an acceptable result with significantly reduced time in comparison to FMH. This was to be expected since FMR can learn its policy from the past traffic changes while FMH searches for a migration solution in a new state space.

Finally, we evaluate the FMH and FMR performance in a large-size scenario. We use the BAL, WAL, and ERL topologies with 200 nodes. We plot the results produced by
T.-M. Pham: Optimizing Service Function Chaining Migration With Explicit Dynamic Path

FMR without retraining its policy. Fig. 5 shows that FMH is slightly better than FMR in terms of the reallocation cost after the same iterations. We also observe that FMR is significantly better than FMH in computation time. It infers that an NSP should use FMR to find a migration solution when an NFV network requires a prompt response to demand changes.

**FIGURE 6.** Impact of the number of changing demands.

**C. COST EVALUATION OF SFC MIGRATION STRATEGIES**

We first evaluate the impact of an overloaded network on the reallocation cost of an SFC migration strategy. We randomly select a demand and double its bandwidth requirement. Fig. 6 plots the reallocation cost of SFC migration in terms of the number of changing demands. We observe that the reallocation cost increases quickly then slows down. We argue that when the number of changing demands increases until a threshold, most nodes provided VNFs to a service demand are overloaded, and the migration happens in the whole NFVI.

We then evaluate the impact of different SFC migration strategies on the system’s operating cost in the long term. In an evaluation scenario, we change the number of migrations in one day. We compute the operating cost of three SFC migration solutions when considering three different objective functions, including the deployment cost, the migration cost, and the reallocation cost. Fig. 7 plots the operating cost of different SFC migration solutions produced by FMR in terms of the number of migrations that happened on one day. We observe that the operating cost is lowest when we optimize the deployment cost in a scenario of ten migrations, while the operating cost is smallest when we optimize the migration cost in a scenario of one hundred migrations. This occurs because the impact of the deployment cost on the operating cost is higher than that of the migration cost when migration happens rarely. The result suggests that to optimize the operating cost in the long term, we should select the deployment cost as an objective function in a low dynamic environment and the migration cost as an objective function in a highly dynamic environment. To balance the operating cost in both environments, we should select the reallocation cost as an objective function.

To summarize, the solution produced by FMH and FMR is close to the optimal solution produced by FMO. When we do not retrain FMR’s policy, FMR can still produce an acceptable migration solution with significantly reduced time compared to FMO and FMH. The results also argue that optimizing the operating cost in the long term depends on selecting an appropriate objective in connection with the rate of migrations.

**VII. CONCLUSION**

We studied the optimization problem of SFC migration with explicit dynamic paths in NFV. We proposed two ILP models
(i.e., FPO and FMO) to find the optimal solution for SFC placement and migration. The key novel feature of the models is the ability to provide the explicit mapping between a logical path and physical links for a service demand in NFVI, where routing is dynamic. They also employ other vital features of SFC migration, such as the VNF sequence and SFC delay, to maximize cost efficiency. We proposed the FMH and FMR algorithms based on SA and SAC, which provide an efficient SFC migration for a large-scale NFV network. The evaluation results show that FPO, FMO, FMH, and FMR can provide an efficient solution for SFC migration. The results also suggest that an NSP should consider the selection of an appropriate objective in connection with the rate of migrations for optimizing the operating cost in the long term. Possible directions of our future work include the consideration of several realistic evaluation scenarios with various physical network parameters, an optimization model of SFC migration with federated NFVI, or a resilient SFC migration as in [25].

**ACKNOWLEDGMENT**

The author would like to thank the anonymous reviewers for many constructive helpful comments.

**REFERENCES**

[1] Network Functions Virtualisation (NFV): Management and Orchestration, document GS NFV-MAN 001 V1.1.1, ETSI, 2014.

[2] C. Bernstein, A. Rahman, J. Zaniga, L. Contreras, P. Aranda, and P. Lynch, *Network Virtualization Research Challenges*, document RFC 8568, IRTF, 2019.

[3] Y. Wang, G. Xie, Z. Li, P. He, and K. Salamatian, “Transparent flow migration for NFV,” in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.

[4] M. Huang, W. Liang, Y. Ma, and S. Guo, “Throughput maximization of delay-sensitive request admissions via virtualized network function placements and migrations,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[5] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function instance migration in network function virtualization architectures,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[6] C. Sun, J. Bi, Z. Meng, T. Yang, X. Zhang, and H. Hu, “Enabling NFV elasticity control with optimized flow migration,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2288–2303, Oct. 2018.

[7] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, “Delay-aware flow migration for embedded services in 5G core networks,” in *Proc. Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[8] K. Hejia and X. Hesselbach, “Offline and online power aware resource allocation algorithms with migration and delay constraints,” *Comput. Netw.*, vol. 158, pp. 17–34, 2019.

[9] J. Zheng, T. S. E. Ng, K. Sripanidkulchai, and Z. Liu, “Pacer: A progress management system for live virtual machine migration in cloud computing,” *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 4, pp. 369–382, Dec. 2013.

[10] L. M. Pham and T.-M. Pham, “Autonomic fine-grained replication and migration at component level on multicloud,” *Vietnam J. Comput. Sci.*, vol. 4, no. 1, pp. 39–49, Feb. 2017.

[11] S. Wang, J. Xu, N. Zhang, and Y. Liu, “A survey on service migration in mobile edge computing,” *IEEE Access*, vol. 6, pp. 25511–25528, 2018.