VNF Placement and Resource Allocation in SDN/NFV-Enabled MEC Networks

Conference Paper · April 2020

DOI: 10.1109/WCNCW48565.2020.9124910

4 authors:

Nahida Kiran
Beijing University of Posts and Telecommunications

Xuanlin Liu
Beijing University of Posts and Telecommunications

Sihua Wang
Beijing University of Posts and Telecommunications

Changchuan Yin
Beijing University of Posts and Telecommunications

VNF Placement and Resource Allocation in SDN/NFV-enabled MEC Networks

Nahida Kiran, Xuanlin Liu, Sihua Wang, and Yin Changchuan
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications, Beijing, China
Email: kiranbupt@yahoo.com, {kiran, xuanlin.liu, sihuawang, ccyin}@bupt.edu.cn

***Abstract*—Network function virtualization (NFV), software defined networks (SDNs), and mobile edge computing (MEC) are emerging as core technologies to satisfy increasing number of users' demands in 5G and beyond wireless networks. SDN provides clean separation of the control plane from the data plane while NFV enables the flexible and on-the-fly creation and placement of virtual network functions (VNFs) and are able to be executed within the various locations of a distributed system and, in our case, in the NFV-enabled MEC nodes. VNF placement and resource allocation (VNFPRA) problem is considered in this paper which involves placing VNFs optimally in distributed NFV-enabled MEC nodes and assigning MEC resources efficiently to these VNFs to satisfy users' requests in the network. Current solutions to this problem are slow and cannot handle real-time requests. To this end, we propose an SDN-NFV infrastructure to tackle the VNFPRA problem in wireless MEC networks. Our aim is to minimize the overall placement and resource cost. Two algorithms are proposed: (i) an optimal solution formulated as a mixed integer program (MIP) problem (ii) a genetic based heuristic solution. The superior performance of the proposed solution is confirmed in comparison with two existing algorithms such as Random-Fit Placement Algorithm (RFPA) and First-Fit Placement Algorithm (FFPA). The results demonstrate that a coordinated placement of VNFs in SDN/NFV enabled MEC networks can satisfy the objective of overall reduced cost. Simulation results also reveal that the proposed scheme approximates well with our optimal solution returned by gurobi and also achieves reduction on overall cost compared to other methods.**

I. Introduction

Mobile edge computing (MEC) technology has been introduced to reduce resource demands and user experienced delays by moving computing, networking, and storage capabilities from the core cloud to the mobile edge network. By creating a cloud-like environment which has computing, storage, and networking functions on the base stations that are adjacent to the users, users can place requested contents or virtual network functions (VNFs) on adjacent clouds (which are why they are called "edge cloud"), and overall distributed resources can be efficiently managed and delays can be reduced [1]. This MEC-enabled architecture that enables cloud computing capabilities at the edge of the access network is widely accepted by the literature in order to serve the increased requirements of the 5G services [2].

MEC implementation can also potentially leverage Network Function Virtualization (NFV) platforms, which eliminates the restrictions applied by the legacy LTE networks, e.g., the fixed placement of the network functionalities [3]. NFV enables the flexible and on-the-fly creation and placement of virtual network functions (VNFs), satisfying the diverse application requirements and optimizing the management of the heterogeneous (network, computational, and storage) resources. Hence, application and network functionalities are handled as VNFs and managed by an NFV Orchestrator (NFVO) [4], and are able to be executed within the various locations of a distributed system and, in our case, in the NFV-enabled MEC nodes.

The combination of SDN, NFV and MEC technologies are widely considered to be essential in the development of 5G. MEC, together with SDN and NFV will play to address the challenges 5G aims to undertake. MEC can be considered to be a kind of NFV. We consider MEC nodes to be any NFV-enabled network nodes on which MEC services can be hosted with allocated VNFs. MEC involves placing compute power in new locations. This computation may involve network functionality and end-user applications. None of the studies on the integration of MEC and SDN explores the use of NFV to deploy specific VNFs to offer final user services, such as VoIP, video streaming, or web browsing.

To benefit from the advantages of SDN, NFV, and MEC, VNFs need to be provisioned with sufficient resources in edge servers without impacting network quality of service (QoS). This paper proposes an SDN/NFV-enabled MEC model for optimal virtual network function placement and resource allocation (VNFPRA) with the goal to minimize both the overall deployment cost and resource cost. A single VNF might be enough to handle certain amount of requests before it becomes overloaded. But, when there is a sudden spike of requests, one VNF may not be enough to process all the received requests. In such case, we investigate the impact of replicating VNFs from the neighboring MEC nodes or from the remote cloud with the help of centralized SDN controller. When receiving a VNF request, NFV-enabled MEC node either process the request and interact directly with the user if the requested VNF is available in it, if not, SDN controller decides whether the requested VNF should be forwarded. Whenever a new VNF request from user equipment (UE) enters, the first NFV-enabled MEC node receives a description packet, which is forwarded to the SDN controller. This first packet contains the description of the required VNF that the UE requires, required bandwidth and the destination node. If the required VNF is not in the NFV-enabled MEC node, the controller finds the optimal route and allocation for the requested VNF.

The main contributions in this work are as follows: (i) We formulate and solve the VNF placement and resource allocation (VNFPRA) problem as a mixed integer program (MIP) problem taking into account the possibility of NFV-

This work was supported in part by the National Natural Science Foundation of China under Grant 61671086, in part by Beijing Natural Science Foundation and Municipal Education Committee Joint Funding Project under Grant KZ201911232046, and in part by the 111 Project under Grant B17007.

978-1-7281-5178-6/20/$31.00 ©2020 IEEE

enabled MEC service instantiations; (ii) We propose a novel SDN/NFV-enabled MEC infrastructure for low-cost MEC resource allocation framework; (iii) We propose a genetic based heuristic solution for VNFPRA problem which is theoretically proven to be sound and complete. (iv) Our results demonstrate that a coordinated placement of VNFs in SDN/NFV enabled MEC networks can satisfy the objective of overall reduced cost. Simulation results also reveal that the proposed scheme approximates well with our optimal solution returned by gurobi and also achieves reduction on overall cost compared to other methods.

The paper is organized as follows. Section II addresses the related work. Section III describes the proposed system model. Section IV addresses the problem formulation. In Section V, we present details of cost-aware VNFPRA for NFV-enabled MEC. In Section VI and VII, we show our simulation results and conclude the paper, respectively.

II. RELATED WORK

Recently, with the advent of SDN [5] and NFV [6], the concept of NFV-enabled MEC emerged whereby services can be hosted at any conventional network node that has virtualized resources (e.g., switches, routers, access points, etc.). Within this new model, the on-demand network resource allocation in the form of VNFs becomes achievable. This has the advantage of reducing investments in specialized hardware, while allowing flexible scaling of the capacity of the functions [7]. Additionally, routers and switches can use their unused computational resources to implement these VNFs [8].

NFV has become an essential technology for delivering network services to the end users. The network services accessed by users are becoming more dynamic and diverse with the rapid growth of user data traffic [9]. NFV can also be interconnected with other technologies such as internet of things (IoT), cloud computing, and MEC. It can also provide the necessary flexibility to offer diverse and dynamic network services through the Service Chaining (SC) model. SDN and NFV are complementary in nature. They have intersected objectives of substituting proprietary and closed network elements (switches) with commodity hardware that can be controlled by standardized software [10]. The resources in MEC nodes are restricted compared to that in remote data centers (DCs) [11]. By combining NFV and MEC technologies these resources can be utilized efficiently because network operators can assign resources for VNFs according to the MEC resource usage requirement.

Although there are some methodologies for managing NFV and edge resources, there is still a chain of open issues and further gaps with regard to their well-organized integration and edge infrastructure management in NFV ecosystems. In order to address these gaps, we have taken a mixed environment which is consistent with current state-of-the-art SDN and NFV architectures. This paper addresses the interaction of the three technologies (SDN, NFV, and MEC) with each other and builds a new framework for optimal VNF placement and resource allocation to minimize the overall deployment and resource cost.

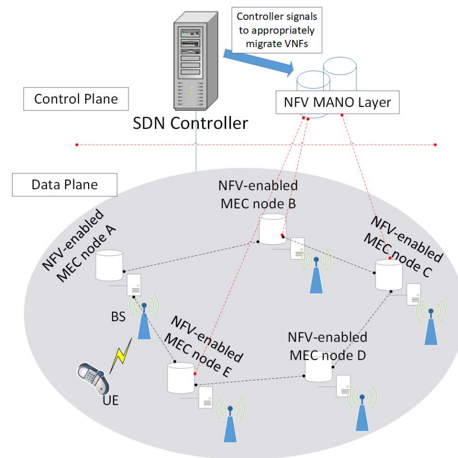

Fig. 1. System model.

III. SYSTEM MODEL

Our system model consists of a control plane and a data plane as shown in Fig. 1. We assume that the network is made of five small base stations (BS) and each BS is attached with an NFV-enabled MEC node connected directly to each other that are capable of being controlled by an SDN controller. As in SDN-enabled technologies, only the centralized controller will have the full view of the global network topology and it will be a responsive agent, which will react based on the requests/packets that it receives. We also assume that the SDN controller can communicate with the management and orchestration (MANO) layer of the NFV. As stated in [12], virtual network function manager (VNFM) is the critical component of the NFV architecture, also called MANO. In this system, we consider each MEC server has VNFs which are managed by a VNFM locating in the core network. VNFM is implemented so that it can start and stop VNFs based on local demand. By doing this, we can reduce deployment costs, power consumption as well as improve service readiness to operate a network component.

In order to effectively manage the network, there are two main decisions we need to make (i) **VNF placement**, i.e., how to allocate VNFs and assign optimal set of resources by considering the resource availability in MEC nodes. The cost incurred in deploying VNF services and delivering them to the end-users must be minimized for cost-efficiency , and (ii) **Resource allocation**, i.e., how the computational capabilities available at NFV-enabled MEC nodes are allocated to the VNFs. We use $\mathcal{G}(\mathcal{V}, \mathcal{L})$ to denote the network with N users, where $\mathcal{V} = \{1, 2, \cdots, V\}$ is the set of NFV-enabled MEC nodes and $\mathcal{L}$ is the set of links between the NFV-enabled MEC nodes. In this paper, we assume that any end user can be within the coverage of one MEC node. Total demand that each VNF may satisfy is limited. For simplicity, we assume that one end user sends only one request to the MEC node and one VNF can handle 50 requests. Requests sent from users should be first received by the NFV-enabled MEC node j, where $j \in \mathcal{V}$ and it interacts directly with the user if the requested VNF is available in it. But, when there is a sudden spike of requests and jth MEC node cannot process these requests because of capacity limit, it can replicate required VNFs from the neighboring

MEC nodes/remote cloud by VNFM. Each MEC node $j \in \mathcal{V}$ is attributed with a vector of available resource capacities Y (e.g. CPU, Memory). We consider that each NFV-enabled MEC node $j \in \mathcal{V}$, has maximum computing capacity C_j^{max} and maximum storage capacity S_j^{max}, which is the maximum capacity to compute and store VNFs. With $\mathcal{F}$, we denote the set of VNFs of different types where $\mathcal{F} = \{1, 2, \cdots, F\}$. Each VNF $i \in \mathcal{F}$ on MEC node $j \in \mathcal{V}$, if activated, consumes some physical resources (i.e., DRAM memory, CPU cycles) and has a predetermined resource requirement of computing capacity c^i (CPU cycles) and storage capacity s^i (memory) to process it.

IV. PROBLEM FORMULATION

A_{ij} is an allocation matrix where

$$A_{ij} = \begin{cases} 1, & \text{if VNF } i \text{ is allocated in MEC node } j \\ 0, & \text{otherwise.} \end{cases}$$

Initially we migrate the VNFs from remote cloud and store them in distributed MEC nodes. This migration requires placement cost. After that we replicate the VNFs from neighboring MEC nodes if needed rather than replicating from remote cloud. This replication may cause link usage cost (replication cost) to replicate VNFs from adjacent MEC nodes.

When a MEC node does not have the VNF application requested by some users, we should determine the whether application VNF should be replicated from another MEC node or not. Let $l(j, k)$ denote the link between MEC node j and MEC node k and every link $l \in \mathcal{L}$ has capacity c_l and per unit link usage cost of u_l. We denote by y_l^i the decision variable stating whether the link is used to replicate VNF i from NFV-enabled MEC node k to NFV-enabled MEC node j. Bandwidth demand for replicating VNF i over the link l is denoted as b_l^i.

$$y_l^i = \begin{cases} 1, & \text{if VNF } i \text{ is replicated over the link } l \text{ ;} \\ 0, & \text{otherwise.} \end{cases}$$

Based on above components, we address the VNF placement and resource allocation problem subject to various constraints. We assume that we have three different types of costs incurred as follows:

(1). **VNF Placement/Allocation Cost**:

This is the cost incurred if any VNF is placed/hosted in NFV-enabled MEC node. The optimal VNF placement involves determining how many VNF instances are needed to meet the end user's workload and on which servers they should be deployed. This cost can be defined as

$$\sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{V}} A_{ij} C_i^j, \tag{1}$$

where C_i^j is the cost of placing VNF i at NFV-enabled MEC node j. A_{ij} is the allocation matrix.

(2). **Resource/Computation Cost**:

This is the cost per utilized processor and memory of the NFV-enabled MEC node. This cost can be expressed as

$$\sum_{j \in \mathcal{V}} \sum_{i \in \mathcal{F}} \sum_{Y} A_{ij} R_i^j C_Y^j, \tag{2}$$

where R_i^j is the number of VNF requests for VNF i from MEC node j, which is known or given. C_Y^j is the cost of resource Y(e.g. CPU, Memory) at MEC node j.

(3). **Link Usage/Replication Cost**:

This is the cost per utilized capacity of the virtual links. This can be defined as

$$\sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{V}} \sum_{l \in \mathcal{L}} u_l^i b_l^i y_l^i, \tag{3}$$

where u_l^i is the link usage cost for replicating VNF i. b_l^i is the bandwidth allocated to vnf i over the link l.

The objective function minimizes the overall resource cost. These are the placement cost incurred if required VNF is placed/hosted in NFV-enabled MEC node, cost per utilized memory and CPU of the NFV-enabled MEC node, and the cost per utilized capacity of the links.

$$\min_{A_{ij}, y_l^i} \quad \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{V}} A_{ij} C_i^j + \sum_{j \in \mathcal{V}} \sum_{i \in \mathcal{F}} \sum_{Y} A_{ij} R_i^j C_Y^j + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{V}} \sum_{l \in \mathcal{L}} u_l^i b_l^i y_l^i \tag{4}$$

subject to

$$\sum_{i=1}^{F} A_{ij} c^i \le C_j^{max}, \qquad \forall j \in \mathcal{V}, \tag{4a}$$

$$\sum_{i=1}^{F} A_{ij} s^i \le S_j^{max}, \qquad \forall j \in \mathcal{V}, \tag{4b}$$

$$\sum_{j=1}^{V} A_{ij} \ge 1, \qquad \forall i = 1, 2, \cdots, F, \tag{4c}$$

$$\sum_{l \in \mathcal{L}} y_l^i b_l^i \le c_l, \qquad \forall i = 1, 2, \cdots, F, \tag{4d}$$

$$A_{ij} \in \{0, 1\}, \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{V}, \tag{4e}$$

$$y_l^i \in \{0, 1\}, \qquad \forall i \in \mathcal{F}, \quad \forall l \in \mathcal{L}, \tag{4f}$$

where C_i^j is the cost of placing VNF i at NFV-enabled MEC node j. C_Y^j is the cost of resource Y at NFV-enabled MEC node j where $Y = [y_1, y_2]$. y_1 is the number of CPU resource and y_2 is the number of memory resource. (4a) and (4b) represents sum of computing capacity of NFV-enabled MEC node j for activating VNFs cannot exceed the maximum computing capacity C_j^{max} and storage capacity S_j^{max} of each NFV-enabled MEC node. (4c) indicates that at least one VNF should be existed/hosted on NFV-enabled MEC node. (4d) makes sure that the bandwidth b_l^i allocated to VNF i should not exceed the capacity c_l of the link. (4e) limits the allocation matrix to either 0 or 1. (4f) is the decision variable.

V. COST-AWARE VNF PLACEMENT AND RESOURCE ALLOCATION FOR NFV-ENABLED MEC

To find the optimum placement of MEC nodes required for the deployment of VNFs, the problem addressed in this paper can be formulated as a Mixed Integer Program (MIP) problem that takes into consideration the virtualization of MEC applications. To solve the scalability problem of the exact MIP solution, two algorithms are used: (i) an optimal solution formulated as MIP problem and (ii) a genetic based heuristic solution for optimal VNF placement and replication. Since the allocation of VNFs is known to be NP-hard [13],

	MEC n1	MEC n2	MEC n3	MEC n4	MEC n5	MEC n6	MEC n7	MEC n8	MEC n9	MEC n10
VNF1	0	0	1	0	0	0	0	0	0	0
VNF2	0	1	0	0	0	0	0	0	0	0
VNF3	1	0	0	0	0	0	0	0	0	0
VNF4	1	0	0	0	0	0	0	0	0	0
VNF5	0	0	0	1	0	0	0	0	0	0
VNF6	0	0	0	0	0	0	0	1	0	0
VNF7	0	0	0	0	0	0	0	0	0	1
VNF8	0	0	0	1	0	0	0	0	0	0
VNF9	0	0	0	0	0	0	0	0	0	1
VNF10	0	0	0	0	0	0	1	0	0	0

Fig. 2. Chromosome representation.

for large networks, heuristics are important to decrease the computation time, while MIP models are only feasible for small networks.

A. Proposed Algorithms for our Optimization Problem

To solve the scalability problem of the exact MIP problem, we propose an optimal solution formulated as MIP problem and a heuristic based genetic algorithm for VNF placement and resource allocation (GA-VNFPRA) which is a heuristic algorithm used to search for an optimal solution by simulating the natural evolutionary process (selection, mutation and crossover). The algorithm optimizes the placement of VNFs which are distributed on the NFV-enabled MEC nodes. Furthermore, our proposed approaches are compared with two existing approaches (i.e. random fit placement algorithm (RFPA) and first fit placement algorithm (FFPA) approaches). The detailed explanation of the proposed heuristic algorithm for our optimization problem is as follows:

(1). Chromosome Representation:

A chromosome signifies a possible VNF placement solution. We implement a matrix representation for a chromosome. We use binary encoding method to represent the assignment VNF placement solution. The placement matrix solution is an $m \times n$ matrix if there are m VNFs and n NFV-enabled MEC nodes. If the i-th VNF is deployed on the j-th MEC node, the entry (i, j) is set to 1 to guarantee that the VNF_i is deployed only once on the MEC node j and other elements in the i-th row must be equal to 0. This binary encoding technique contributes to simplify the crossover and mutation operation between two matrices. Fig. 2 shows an example of the proposed chromosome for our optimization problem. In this example, there are 10 VNFs. It describes that the VNF1 is deployed on MEC node 3, VNF2 is deployed on node 2, etc.

(2). Population Initializing:

We generate initial population I_p which grows in each generation of genetic algorithm. First, we generate the accessible set S_i^j for each VNF i in MEC node j. S_i^j represents the set of all base stations from which user is getting signals. Then, the MEC node calculates the accessible probability $P_{i,k}^S$ for each VNF i in MEC node k, which signifies how likely user can connect to another MEC node. We calculate $P_{i,k}^S$ by the following equation:

$$P_{i,k}^S = \frac{S_{i,k}^j}{max_{\forall k \epsilon S_i^j} S_{i,k.}^j}. \tag{5}$$

(3). Fitness Function: We should obtain an optimal placement solution considering our main objective, minimizing the overall resource cost. In order to be consistent with the final objective, the fitness of chromosome is calculated based on overall cost. The fitness function evaluates how a chromosome matches our main objective. In our Algorithm, fitness is defined as

$$fitness(a_i) = 1/obj.func, \tag{6}$$

where a_i is an individual in the population and $obj.func$ is our objective function. Considering the objective of this paper, we calculate the placement and resource cost of all the VNFs mapped on each MEC node according to the matrix placement solution. Then, we get the best fitness value for this solution.

Algorithm 1 solves our optimization problem by generating a group of random chromosomes and evaluate each chromosome by equation (4)-(4f). Two best chromosomes whose resource cost in (4) is lower are selected to produce another group of random chromosomes by mutation and recombination. This technique lasts until the acquired result reaches convergence.

Algorithm 1 Genetic Algorithm for VNFPRA

```
/ / Input Parameters:
  𝒢, N, ℱ, R_i^j, C_i^j, C_Y^j, u_l^i
  Constraint:
    • MEC node Capacity
    • VNF placement
    • link Capacity
/ / Initialise generation 0:
  encode VNF placement in MEC nodes into chromosomes;
  k = 0 ;
  P_k = set of β randomly generated chromosomes;
/ / Evaluate P_k:
  calculate Cost(α) for each α element P_k;
  While k < maxGeneration and Cost(α) > minimumCOst do
      / / Select:
      select two chromosome α with lowest cost,
      insert into P_{k+1};
      / / Crossover:
      for j = 1 to (P_k − 2)/2 do
        select chromosome α_a and α_b randomly from P_k;
        one-point crossover to α_a and α_b into α_c and α_d;
        insert α_c and α_d into P_{k+1};
      endfor
      / / Mutate:
      for j = 1 to (P_k − 2)/2 do
        select α_j from P_{k+1};
        mutate each bit of α_j to generate α'_j;
        update α_j in P_{k+1} with α'_j;
      endfor
      / / Evaluate P_{k+1}:
        calculate Cost(α) for each α element P_{k+1};
        k = k + 1;
  endwhile
  return chromosome α with lowest cost ;
  decode chromosome α into MEC nodes and VNF type;
```

For VNF migration in our model we can use the following fitness function:

$$F_n = \sum_{\forall i \epsilon \mathcal{F}} U_{n,i.}^j \tag{7}$$

Here, F_n is the fitness value for n-th individual and U_i^j is the utility of VNF i, which is defined by the following equation:

$$U_{n,i} = P_{n,i}^j + T_{n,i}^j + L_{n,l}^j, \tag{8}$$

where $P_{n,i}^j$ is the minimum processing time of VNF i for the n-th individual when VNF i is migrated from MEC node j to k. When VNF i is not migrated, it will be equal to k. We calculate $J_{n,i}^j$ by this equation

$$P_{n,i}^{j} = min_{\forall k \epsilon S_{i,j}} \{ \frac{P_{n,i}^{j,k}}{max_{\forall k \epsilon S_{i,j}, t \epsilon I_p} P_{t,i}^{j,k}} \}. \tag{9}$$

Moreover, $T_{n,i}^{j}$ is the transfer time of VNF i for the n-th individual from MEC node j to k. When MEC node j is same as MEC node k, the VNF transfer time will be equal to $T_{i}^{j,j}$ = 0. Using the below equation we calculate minimum VNF transfer time as

$$T_{n,i}^{j} = min_{\forall k \epsilon S_{i,j}} \{ \frac{T_{n,i}^{j,k}}{max_{\forall k \epsilon S_{i,j}, t \epsilon I_p} T_{t,i}^{j,k}} \}, \tag{10}$$

where $L_{n,i}^{j}$ is the VNF load association degree of l type VNF between MEC node j to k, which helps the system to predict the load of the MEC node. Minimum normalized VNF load association can be calculated as

$$L_{n,l}^{j} = min_{\forall k \epsilon S_{i,j}} \{ \frac{L_{n,l}^{j,k}}{max_{\forall k \epsilon S_{i,j}, t \epsilon I_p} L_{t,l}^{j,k}} \}. \tag{11}$$

We use min-max normalization technique to calculate $P_{n,i}^{j}$, $T_{n,i}^{j}$, and $L_{n,l}^{j}$.

(4). Genetic operators:

Once fitness value for each individual is calculated, next we have to choose two individuals for the crossover operation. At the start of the selection process, the selection probability P_n^s for each individual n is calculated using this equation:

$$P_n^s = \frac{F_n}{\sum_{\forall t \epsilon I_P} F_t}. \tag{12}$$

Here, an individual with greater P_n^s has greater chance to be selected. Next, a mutation operation is performed on the new offspring with the mutation probability P^m and then we reduce the offspring which are not compatible with VNF's accessible set. To retain the stochastic property of VNFPRA problem, we replace an individual from I_P with the offsprings such that the offspring fitness values are higher than the individual fitness values. The whole procedure is listed in Algorithm 2.

Algorithm 2 Genetic Algorithm for VNF Migration

Input:
Fitness Function.
Output:
Remapped NFV-enabled MEC node for each VNF.
1. $I_P \longleftarrow PopulationInitialization()$
2. while $(iteration < maxitr \& notconverge)$ **do**
3. Pick out two individuals based on P_n^s.
4. Execute the crossover operation.
5. Reduce the incompatible offspring.
6. $\eta_b \longleftarrow$ Topmost offspring.
7. $\eta_r \longleftarrow$ Random offspring.
8. $\gamma_w \longleftarrow$ Worst individual in I_P.
9. $\gamma_r \longleftarrow$ Random individual in I_P with fitness $< \alpha_r$.
10. Replace γ_w and γ_r in I_P with η_b and η_r respectively.
11. Repeat with iteration $\longleftarrow$ iteration +1.
12. end while
13. Return the best individual.

B. Gurobi Optimizer

This is an exact algorithm which derives the optimal solution to (4) using the Gurobi optimizer [14].

VI. PERFORMANCE EVALUATION

In this section, we assess the performance of our proposed methods for VNFPRA. We develop a simulation model using python programs to evaluate the performance of the proposed algorithms. Each MEC node is randomly deployed in the network. The system initially generates the target users, each of them is randomly distributed in the coverage area of different NFV-enabled MEC nodes. We use 5 MEC nodes, each having [1.5-3.0] GHz processor and [8-64] GB memory. Each MEC supports 5 UEs.

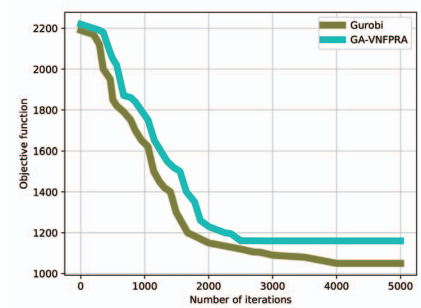

Fig. 3. Objective function vs. number of iterations.

In this work, first of all we compare the proposed MIP model, implemented using the Gurobi Optimizer [14] , with the proposed genetic algorithm for VNFPRA to show how close our solutions are to the optimal values returned by MIP. Regarding MIP approaches, they are trying to go through every possible solution to extract the optimal one but they are characterized by an increased complexity for large scale experimentations. In contrast to MIP, the genetic algorithm is very simple and easy to implement by simply translating the pseudocode into a python code. To better illustrate the efficiency of the proposed approaches, we compare them against the following RFPA, and FFPA baseline algorithms:

(1) Random-Fit Placement Algorithm (RFPA):

Based on a given number of VNFs, RFPA places each VNF at MEC node selected uniformly at random.

(2) First-Fit Placement Algorithm (RFPA):

This algorithm places each VNF at the first available MEC node that has enough capacity to host it.

Fig. 3 presents a comparison of the objective function values obtained for our proposed GA-VNFPRA and optimal solution returned by Gurobi. MIP algorithm is used as a benchmark since it always has the exact optimal solution but is not implementable for large-scale systems (e.g., MIP needs more execution time than GA). When the number of iterations increases, both the objective value of the proposed heuristic GA-VNFPRA and the Gurobi are gradually decreased and eventually tend to converge. Since we are solving a minimization problem, lower values for a specific placement and resource allocation combination mean better performance. Our proposed GA-VNFPRA is shown to approximate well with the optimal solutions provided by the Gurobi. However, after 2500 iterations, the proposed GA-VNFPRA converges to a local optimum and no further improvements can be achieved while gurobi converges after 4000 iterations, this is because it increases the particular solution computational time for large scale experimentations and thus we can only use it as a benchmark approach or an offline solution.

In Fig. 4, we evaluate the performance of the proposed

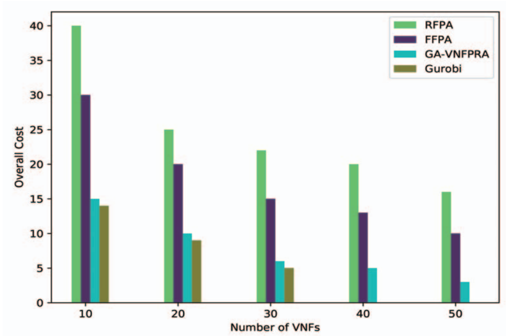

Fig. 4. Overall cost vs. number of VNFs.

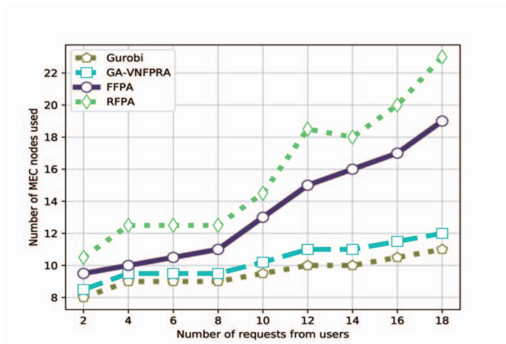

Fig. 5. Number of MEC nodes used.

scheme with the optimal solution provided by Gurobi and two existing approaches. It depicts the overall cost per VNF placement. As was expected, the Gurobi provides the exact optimal solution with lowest cost initially, but with the increasing number of VNFs it takes longer time to produce results and sometimes even can not produce results in a reasonable period of time. The genetic algorithm performs well at much lower cost. Here, we can also see how the cost of the random allocation spikes in comparison with the first fit placement, genetic and the optimum solution. This is because RFPA doesn't consider MEC capacity constraints and also FFPA may not find neighbor MEC nodes that can host the rest of VNF requests, leading to higher cost.

In Fig. 5 we evaluate the total number of MEC nodes required with increasing number of requests from users. The number of MEC nodes needed to facilitate the incoming requests increase rapidly for RFPA and FFPA while GA-VNFPRA algorithm requires minimum MEC nodes to process requests and also it approximates well with the optimal solution. This is because RFPA doesn't consider the best MEC nodes with efficient available resources, which means the allocation of CPU and memory resource is not optimal. When there are too many VNF requests from users, the optimal allocation of resources in MEC nodes for VNFs will become more important.

In Fig. 6 we study the performance of VNF migration for increasing number of requests using our proposed genetic algorithm for VNF migration (GA-VNFM). GA-VNFM requires minimum VNF migrations while in FFPA and RFPA approaches, the number of VNF migrations rapidly increase with number of requests from users. Compared to these approaches number of VNF migration in our proposed GA-VNFM steadily goes to a stable state.

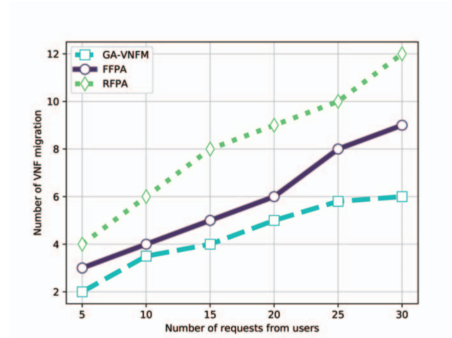

Fig. 6. Number of VNF migration.

VII. CONCLUSION

A framework based on SDN and NFV MANO architecture is presented for deploying VNFs in distributed NFV-enabled MEC nodes. Furthermore, a VNFPRA problem is proposed that requires meeting the optimal placement requirements of VNFs in SDN/NFV-enabled MEC nodes to reduce the deployment and resource cost. This is a critical problem in the field of NFV. This paper proposes a solution for this problem based on two algorithms (i) an optimal solution formulated as MIP problem and (ii) a genetic based heuristic algorithm. Results show the higher performance of the proposed methods in comparison with two existing algorithms in literature. The results also validate the fact that a coordinated placement of VNFs in SDN, NFV, and MEC can satisfy the objectives of overall reduced cost.

REFERENCES

[1] ETSI, ISGMEC, "Mobile-edge computing introductory white paper." *White Paper, Sept (2014).*

[2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657-1681, Third Quarter 2017.*

[3] ETSI NFV, "Network functions virtualisation (NFV); management and orchestration, " *GS NFV-MAN 001, Dec. 2014.*

[4] A. J. Gonzalez, G. Nencioni, A. Kamisinski, B. E. Helvik and P. E. Heegaard, "Dependability of the NFV orchestrator: State of the art and research challenges," *in IEEE Communications Surveys & Tutorials, vol. PP, no. 99, 2018.*

[5] Bouten, Niels, Jeroen Famaey, Rashid Mijumbi, Bram Naudts, Joan Serrat, Steven Latre, and Filip De Turck, "Towards nfv-based multimedia delivery," *in IEEE 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 738-741, 2015.*

[6] R. Mijumbi et al.,"Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys & Tutorials, 2015.*

[7] B Han, V Gopalakrishnan, L Ji, S Lee,"Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag. 53(2), 90-97, 2015.*

[8] R Bonafiglia, I Cerrato, F Ciaccia, M Nemirovsky, F Risso, "Assessing the performance of virtualization technologies for NFV: a preliminary benchmarking," *in Fourth European Workshop on Software Defined Networks (EWSDN), pp. 67-72, 2015.*

[9] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata,"Application of evolutionary mechanism to dynamic virtual network function placement," *IEEE Int. Conf. on Network Protocols (ICNP), Wkshp on Control, Operation and application in SDN protocols (CoolSDN), pp. 1-6, Nov. 2016.*

[10] C Bu, X Wang, M Huang, K Li, "SDNFV-based dynamic network function deployment: Model and mechanism," *IEEE Commun. Lett. 22(1), 93-96, 2018.*

[11] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "Pricing policy and computational resource provisioning for delay-aware mobile edge computing," *in Communications in China (ICCC), 2016 IEEE/CIC International Conference on, pp. 1-6, IEEE, 2016.*

[12] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: State of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network, vol. 28, no. 6, pp. 18-26, 2014.*

[13] Herrera, Juliver Gil, and Juan Felipe Botero,"Resource allocation in NFV: A comprehensive survey," *IEEE Transactions on Network and Service Management 13, no.3*, pp.518-532, August 5, 2016.

[14] Gurobi,"Gurobi optimizer." *[Online]. Available: http://www.gurobi.com*, 2014.