AMF Optimal Placement based on Deep Reinforcement Learning in Heterogeneous Radio Access Network

Hao Jin (hjin@bupt.edu.cn)  
Beijing University of Posts and Telecommunications  
https://orcid.org/0000-0002-0764-850X

Wenzhe Pang  
Beijing University of Posts and Telecommunications

Chenglin Zhao  
Beijing University of Posts and Telecommunications

Research

Keywords: Network Function Virtualization, Mobility Management, AMF Placement, Deep Reinforcement Learning

Posted Date: July 1st, 2020

DOI: https://doi.org/10.21203/rs.3.rs-14323/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
AMF Optimal Placement based on Deep Reinforcement Learning in Heterogeneous Radio Access Network

Hao Jin*, Wenzhe Pang† and Chenglin Zhao

Abstract
To support various service requirements such as massive Machine Type Communications, Ultra-Reliable and Low-Latency Communications in 5G scenario, Network Function Virtualization (NFV) plays an important role in the 5G network architecture to manage and orchestrate network services. As the key network function responsible for mobility management, Access and Mobility Management Function (AMF) can be deployed flexibly at the edge of the radio access network to improve the performance of mobility management based on NFV. In this paper, the optimal placement of AMF is addressed based on Deep Reinforcement Learning (DRL) in a heterogeneous radio access network, which aims to minimize the network utility including the average delay of mobility management requests at AMF, the average wired hops to relay the requests and the cost of AMF instances. By considering time-varying features including user mobility and the arrival rate of user mobility management requests, an AMF optimal placement approach is proposed for the long term optimization. Simulation results show that the performance of the proposed DRL based AMF optimal placement approach outperforms that of the baselines.

Keywords: Network Function Virtualization; Mobility Management; AMF Placement; Deep Reinforcement Learning

1 Introduction
5G service requirements motivate the evolution of the core network since the existing core network architecture is difficult to meet the demands of new services, such as enhanced Mobile Broadband (eMBB), massive Machine Type of Communication (mMTC), Ultra-Reliable and Low-Latency Communication (URLLC) as well as vehicular network [1,2]. As one of the functions of existing core network, mobility management becomes an indispensable technology which should be worked out. In order to deal with the above challenges, Network Function Virtualization (NFV) and Software Defined Networking (SDN) are enabled in the 5G core network architecture [3,4]. Mobility management based on NFV is addressed to meet various service requirements of 5G.

A comprehensive survey is presented in [4] for the up-to-date solutions based on SDN and NFV on the current EPC network architecture. Among the solutions concerning mobility management, two solutions are categorized based on NFV, one is based on Mobility Management Entity (MME) including the solution of virtual mobility management (vMME) and Distributed mobility management (DMME); the other is based on the NG core architecture, which presents new virtual function components for the network architecture [5,6]. Both of the solutions introduce important challenges, mainly related to the placement of the telco-specific VNFs (i.e. MME, PGW, SGW, AMF, SMF, and AUSF) to ensure optimal connectivity for users and simultaneously reduce the deployment cost [7].

In the perspective of research on virtual mobility management, there has been some research issues on the placement of vMME/vEPC NFs. Some of them focus on the mapping of vEPC [7–13], which mainly related to the optimization of mobility management on data plane, while others concentrate on the mapping of vMME [7,11–15] and optimal placement of mobility management related VNFs. For example, in [7], the placement of virtual network slice is proposed over
a federated cloud by deploying the virtual instances of key vEPC NFs considering both 4G and new NFs of 5G. In [8], the concept of soft Evolved Packet Core (softEPC) is presented to instantiate the virtual network functions of EPC flexibly and dynamically over a physical network topology. Regarding the optimization objects, operation cost, load balancing and delay/response time are major objectives in these issues [7,11,12,14].

In the aspect of mobility management related new 5G architecture, according to the 5G network architecture, the main function of mobility management is decomposed into several components, including Access and Mobility Management Function (AMF), Session Management Function (SMF), Unified Data Management (UDM), User Plane Function (UPF) and Authentication Server Function (AUSF). Different placement of AMF, SMF, UDM, UPF and AUSF brings about different performance of mobility management, which sheds a light on the research direction of mobility management [5,6]. In [7], the placement of virtual network slice is addressed by deploying AMF, SMF, UPF and AUSF in 5G. In [16], based on Bayesian coalition formation game, the optimal coalition is formulated to deploy AMF, UPF and SMF collaboratively, the number of VNF instances to instantiate and the virtual resources allocated for each instance are optimized to maximize the profit of each cloud provider and guarantee the Quality of Service (QoS).

Among the network function components related to virtual mobility management in new 5G architecture, AMF is the most important VNF, and its deployment and performance optimization becomes the bottleneck of mobility management [15]. As a typical NFV based virtual network component on control plane, AMF is served to process mobility management requests from mobile users, with different characteristics of mobility management events in different service scenarios. The instances of AMF can be adjusted by optimization algorithms, which are designed from the aspect of QoS for mobility management and OpEx/CapEx of operators for AMF. In [15], from the perspective of scalability of 5G core network, auto-scaling of AMF instances is addressed to cope with the challenges of mobility management posed by massive number of connected devices. With the static threshold method, the decision is made to scale-out or scale-in the AMF instances by setting threshold parameter of network load to avoid wasting resources. In [13], EPC is modeled as a simplified queuing model based on VNF and an auto-scaling algorithm is proposed to obtain the tradeoff between auto-scaling and cost in 5G network. Although the above static threshold based method is effective and easy in auto-scaling of instances, it leads to service level agreement violations, and the thresholds are difficult to configure and change frequently [17]. That is to say, the intrinsic dynamics of user mobility and the arrival rate of user mobility management requests are important factors which should be taken into consideration on the optimization of mobility management.

To overcome the above threshold configuration problem, machine learning (ML) based approach is used [18–20]. In [18], [19] and [20], the traffic load of AMF is forecasted by Convolutional Neural Network (CNN), Deep Neural Network (DNN) and Long Short Term Memory Network (LSTM) based on real datasets in mobile network, respectively. With the forecast traffic result, AMF instances are optimally scaled. The ML method achieves better performance compared with threshold-based solutions. However, the historical datasets are required to train the model and some ideal assumptions is needed. It is difficult to cope with the optimization of virtual mobility management with complete knowledge of the scenarios by the traditional approaches due to inherent dynamics of user mobility and mobility management requirements. As a model-free method without any datasets and prior information of the environment, reinforcement learning (RL) has been used as a promising approach to solve resource and service management problems in mobile network [21,22]. For example, in [23], a Deep Reinforcement Learning (DRL) based approach is proposed to optimize different slice request allocation to the network slices. In [24], a dynamic resource reservation and virtual radio access network resource slicing algorithm is proposed based on DRL to control the increase or decrease of the slice resource based on QoS and user resource utilization. In [25], an accelerated two-step RL algorithm is proposed for the proper VNF sizing and placement in a real network under various environments. In [26], a DRL approach is proposed to automatically make a decision optimally allocating the network resources through observing the network states in the scenario of in-network caching and device-to device (D2D) communications. In [27], a deep learning based real-time energy-efficient power control approach is presented which can operate in an online fashion even in high mobility scenarios.

With the above investigation, to the best of our knowledge, the optimal placement of AMF based on DRL remains a challenging issue taking into account the time varying features including user mobility and the arrival rate of user mobility management requests. In this paper, the optimal placement of AMF based on DRL is addressed in the heterogeneous radio access network. The main contributions are listed as follows:

1. The optimal placement of AMF in the heterogeneous radio access network is formulated as a
Markov Decision Process (MDP) problem to minimize the network utility including the average delay of mobility management requests at AMF, the average wired hops to relay requests and the cost of AMF instances.

2. The time-varying features are considered including user mobility and arrival rate of user mobility management requests.

3. To solve the problem of large state and action space, an algorithm based on DRL is proposed called AMF-OP-DRL, in which DNN is used to estimate the value function of Q-Learning. The convergence performance of AMF-OP-DRL algorithm is presented under different batch sizes and learning rates. The performance of AMF-OP-DRL is evaluated compared to baseline algorithms. The impact of some factors to the accumulated network utility are analyzed, including the arrival rate of mobility management requests, the number of VNF instances on each AMF server as well as the weight coefficients in the network utility.

The rest of this paper is organized as follows. In Section 2, the system model for AMF optimal placement in HetNet is described. In Section 3, the optimization problem is formulated to place AMF. The DRL based algorithm is described in Section 4. In Section 5, the performance of the optimization algorithm is evaluated by simulation. Finally, conclusions are given in section 6.

2 System model

In order to meet mobility management demands of various services for 5G, AMF can be placed nearby Radio Access Network (RAN). By allowing universal virtual machines (VMs) to host AMF, the number of AMF instances can be orchestrated according to the mobility management requirements and AMF resources.

In this section, in the scenario of heterogeneous radio access network, the system model for AMF optimal placement is provided, and the framework of AMF orchestration based on MANO is illustrated. The modeling of the placement of AMF servers with multiple instances is presented based on a queuing system.

2.1 Scenario of AMF optimal placement in heterogeneous radio access network

Fig.1 gives the system model of AMF optimal placement in the scenario of a heterogeneous radio access network with several macro base stations (MBSs) and small-cell base stations (SBSs).

In Fig.1, MBSs are deployed to provide wide area coverage, and SBSs are provided to enhance network coverage and increase transmission capacity. Assuming that MBSs and SBSs are deployed and managed by the same network operator, both MBSs and SBSs are equipped with universal servers, and each server is enabled with several AMF instances. There are wired links between BSs. The AMF servers can be placed on MBSs and SBSs in the RAN, which are orchestrated and managed based on MANO.

Since each User Equipment (UE) accesses AMF via its attached SBS, that is to say, the UE interacts with AMF either directly when an AMF server is placed on its attached SBS, or it interacts with AMF by one-hop or multi-hop of BSs, which means that all BSs are relay nodes to access AMF.

2.2 The framework of AMF optimal placement based on MANO

The framework of AMF optimal placement based on MANO is shown in Fig. 2. In the framework, the related resources of the physical infrastructure include computation resource and network resource. The computation resource resided in the MBSs and SBSs are abstracted as the virtual computation resource for AMF, and the network resource refers to those networking links available for transmitting AMF information, which are virtualized by the virtualization layer of the framework.

According to the principles of NFV management and orchestration [3, 28], AMF orchestration modules based on MANO mainly includes NFV Orchestrator for AMF, VNF Manager for AMF and Virtualized Infrastructure Manager for AMF.

NFV Orchestrator for AMF is a use case of NFV Orchestrator, which is responsible for fulfilling resource orchestration functions and network service orchestration functions on AMF. It supports lifecycle management of AMF as network service, AMF instances automation management, policy management and evaluation, as well as enhancement for AMF service instances and AMF instances. It collects usage information of NFVI resources by AMF instances or groups of AMF instances, and supports management of the relationship between AMF instances and NFVI resources allocated to those AMF instances by using NFVI resources repository and information received from Virtualized Infrastructure Manager for AMF [3].

VNF Manager for AMF is in charge of lifecycle management of AMF instances, including AMF instantiation, AMF instance software update/upgrade, modification, scaling out/in and up/down and termination, as well as AMF instance assisted or automated healing and AMF lifecycle management change notification, etc. [3, 29].

Virtualized Infrastructure Manager for AMF is designed to orchestrate the allocation/upgrade/release/reclamation of NFVI resources (including the optimization of such resources usage), and it manages the
association of virtualized resources to physical compute, storage and networking resources. It collects performance and fault information of hardware, software and virtualized resources; and forwards performance measurement results and faults/events information related to virtualized resources. It manages in a repository inventory related information of NFVI hardware and software resources, and discovery of the capabilities and features (e.g. related to usage optimization) of such resources as well [3, 29].

Based on the layered functionality and producer-consumer paradigm, Virtualized Infrastructure Manager for AMF and NFV Orchestrator for AMF collect monitoring parameters for AMFs periodically from NFVI resource repositories (e.g. the arrival rates of mobility management requests on AMFs for different service scenarios). The monitoring parameters are processed and selected as the inputs of the AMF optimization algorithm for AMF instance automation management, which is resided in the NFV Orchestrator for AMF. Based on the AMF optimization algorithm result, VNF Manager for AMF adjusts AMF instances (e.g. AMF instance placement optimization with some rules and constraints); while Virtualized Infrastructure Manager for AMF orchestrates association of virtual resources of AMFs to physical computation and networking resources (e.g. the optimal AMF placement and the number of AMF instances in the physical infrastructure).

From the above framework, NFV Orchestrator for AMF and virtualized infrastructure manager for AMF are of paramount importance to the optimal placement of AMF. Different placement result brings about different cost of AMF deployment and different performance of virtual mobility management.

One of the important criteria to evaluate the performance of virtual mobility management is the response time of mobility management requests on the control plane, which mainly caused by propagation delay, transmission delay and processing delay of mobility management requests. Propagation delay depends on the distance to transmit the requests, and transmission delay of requests is the delay caused by physical layer communication, while the processing delay includes queuing delay and serving delay of requests. Regarding the processing of requests on AMF, queuing delay of requests is affected by the capacity of AMF. The serving delay of requests is related to the average serving time for every mobility management request by one AMF instance. If AMF is placed in the radio access network, backhaul delay, queuing delay within the relay routers or the aggregation delay in the backhaul and CN are avoided [30].

According to the above analysis and the system model shown in Fig. 1, it is a challenge to investigate the tradeoff between the cost of AMF placement, the communication cost caused by AMF placement and the insurance of response time of mobility management requests. In the following subsection, the serving process of AMF with multiple instances is formulated based on a queuing system, including mobility management request model and AMF queuing model. The key notations in the modelling of AMF optimal placement are shown in Table 1.

2.3 Mobility management request model

Based on the system model of Fig. 1, assuming that there are \( N_{BS} \) BSs, including MBSs and SBSs. \( N_{AMF} \) universal servers are deployed at BSs, and each AMF server supports multiple active instances. \( N_{UE} \) UEs are resided in the radio access network. Let \( BS \) denote the set of BSs, and \( UE \) denote the set of UEs who can access the AMF servers.

Assuming that the orchestration of AMF runs over equal length time steps, mobility management requests are collected from UEs at the beginning of each time step, and the orchestration for AMF is made at each time step. At time step \( k \), the mobility management requests of UE \( i \) is denoted by \( \lambda_i(k) \), which follows Poisson distribution. In mobility management, the events processed frequently are Service Request (SR), Service Release Request (SRR) and Handover Request (HR) [31–33], which are represented by \( \lambda_{SRi}(k) \), \( \lambda_{SRRi}(k) \) and \( \lambda_{HRi}(k) \), respectively. Thus, the total requests from UE \( i \) are modeled by the sum of the requests for the three kinds of mobility management events as (1),

\[
\lambda_i(k) = \lambda_{SRi}(k) + \lambda_{SRRi}(k) + \lambda_{HRi}(k)
\] (1)

2.4 AMF queuing model

Assuming that all of the packet size of mobility management requests are the same, then the propagation and transmission delay of one request packet within one hop of BSs is identical. Let \( H_{BS-BS_j} \) be the wired hops between BS \( i \) and BS \( j \).

When a user sends the mobility management request, the request arrives at its serving BS first, and then it is transmitted to an AMF server for processing. If the AMF server is placed on the serving BS, the request is processed by the AMF server without communication cost between the serving BS and the BS deployed with the AMF server, otherwise, the request is relayed to the closest BS deployed with an AMF sever and processed if no AMF server is placed on the serving BS.

The serving AMF with multiple active instances on the BS is modeled as a \( M/M/c \) queue, where \( c \) represents \( c \) AMF active instances on each AMF server, which means there are \( c \) running AMF instances.
Assuming that each AMF instance serves the mobility management requests with serving rate $\mu$ and the serving time follows an exponential distribution. Let $\tau_s = 1/\mu$ denote the average serving time of the requests. The mobility management requests from UEs follow Poisson distribution. Let $\lambda$ denote the request arrival rate of the AMF queuing model, and $\tau_q$ represent the average queuing time, and $\rho = \lambda/\mu$. The average queuing time can be obtained as (2) [34],

$$\tau_q (\lambda) = \frac{C (c, \rho)}{\mu c - \lambda}$$

(2)

Where

$$C (c, \rho) = \frac{\rho^c c}{\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c c}{c!} - \rho}$$

(3)

The average waiting time of mobility management requests is derived as (4):

$$\tau_w (\lambda) = \tau_q (\lambda) + \tau_s$$

(4)

Let $S_0$ be the saturation rate of the $M/M/c$ system, according to the stability condition of $M/M/c$ queuing system, $S_0 < 1$, namely $\lambda/\mu < 1$ [35].

From the above modelling of AMF queuing in the heterogeneous radio access network, it indicates that difference placement of AMF servers with multiple instances lead to different waiting time of mobility management request and different communication cost.

3 Problem formulation

In order to characterize the dynamics of AMFs in the heterogeneous radio access networks, including the placement of AMF servers with active instances on the MBSs and SBSs, the mobility of users as well as the delay caused by AMFs on mobility management request processing, a Markov process is used to model the optimal placement problem. It motivates the adoption of the Markov Decision Process (MDP) to formulate the above concerned problem. In addition, since our aim is to achieve the tradeoff among the cost of AMF placement and the insurance of response time of mobility management requests under some constraints including the minimum number of placement for AMF server and maximum arrival rate of mobility management requests, the reward setting of the MDP will be closely related to the Delay, Hops and CostAMF models.

In the following subsection, the motivation of problem formulation based on MDP is analyzed, and then the formulation of AMF optimal placement is addressed.

3.1 Motivation of problem formulation based on MDP

From the AMF queuing model and mobility management request model, it reveals that the mobility of mobile users plays a key role in the following ways. On the one hand, the arrival rate of mobility management requests of mobile users influences the delay caused by AMF queuing model, which potentially affects decision of active AMF server placement, as well as the cost of AMF placement and the hops to relay the requests. On the other hand, since less arrival rate of user requests on AMF servers leads to the traffic offloading of requests to other active AMF servers, which provides the possibility to turn off some AMF servers on some BSs to save AMF consumption cost. Facing with the dynamics of user mobility, mobility management requests and AMF placement, it is natural to formulate the AMF optimal placement problem from the perspective of MDP.

MDP provides a formalism for reasoning about planning and acting in the face of uncertainty, which is defined by a tuple $(S, A, \{TP\}, r)$. $S$ is the set of possible states, $A$ is the set of available actions, and $(TP)$ provides the transition probabilities to each state if action $a$ is taken in state $s$, and $r$ is the reward function. The process of MDP is illustrated as follows. At an initial state $s(0)$, the agent takes an action $a(0)$; then the state of the system transits to the next state $s(1)$ according to the transition probabilities $\{TP_s(0)\}_{0 \leq i \leq 1}$, and the agent receives a reward $r(0)$. With the process continuing, a state sequence including $s(0)$, $s(1)$, $s(2)$, . . . , is generated. The agent in a MDP aims to maximize a discounted accumulative reward [22].

3.2 The formulation of AMF optimal placement

In this subsection, the AMF placement is formulated as a MDP in the heterogeneous RAN, and the reward function is called network utility, which is defined based on the AMF instance cost, the average waiting time of mobility management requests on AMF as well as the total communication cost to relay mobility management requests. The modelling is described as follows.

At a given placement decision time step $k$, let $U(k)$ denote the serving binary indicator set between UEs and BSs, namely,

$$U(k) = \{u_{i,j}(k) \mid 1 \leq i \leq N_{UE}, 1 \leq j \leq N_{BS}\}$$

where $u_{i,j}(k) = 1$ indicates that UE $i$ is served by BS $j$. In other words, UE $i$ is in the coverage area of BS $j$, otherwise, $u_{i,j}(k) = 0$.

Let $P(k)$ denote the AMF placement on BSs at time step $k$, $P(k) = \{p_j(k), 1 \leq j \leq N_{BS}\}$. If $p_j(k) = 1$, then one AMF server with $c$ active instances is placed
on BS \( j \), otherwise, \( p_j(k) = 0 \). \( p_j(k) \) can be modeled as a two-state Markov chain [26]. The transmission from \( p_j(k) \) to \( p_j(k+1) \) depends on the system action at time step \( k \). For simplicity, let \( \lambda_{B_j}(k) \) represent the mobility management request arrival rate to BS \( j \) at time step \( k \), and \( \lambda_{B_j}(k) = \sum_{i=1}^{N_{BS}} \eta_{i,j} \lambda_i(k) \). The change of \( \lambda_{B_j}(k) \) is modeled as a Markov process, \( \lambda_{B_j}(k) \) is computed at the beginning of every time step. The state transition from \( \lambda_{B_j}(k) \) to \( \lambda_{B_j}(k+1) \) depends on the distribution of users and the arrival rate of mobility events at time step \( k+1 \).

According to the assumption in the AMF queuing model, we assume that the proportion of BS \( j \)'s mobility management requests processed by AMF \( m \) is \( \eta_{BS\rightarrow AMF_{j,m}} \), and \( \eta_{BS\rightarrow AMF_{j,m}} \in [0,1] \). If \( \eta_{BS\rightarrow AMF_{j,m}} > 0 \), some of BS \( j \)'s mobility management requests are processed by \( m-th \) AMF server. Let \( \lambda_{AMF_{m}}(k) \) denote the total mobility management request arrival rate to the \( m-th \) AMF server, then \( \lambda_{AMF_{m}}(k) = \sum_{j=1}^{N_{BS}} \eta_{BS\rightarrow AMF_{j,m}} \lambda_{B_j}(k) \). Based on (4), the average waiting time of mobility management requests on the \( m-th \) AMF server can be computed as (5):

\[
D_{AMF_{m}}(k) = \tau_q(\lambda_{AMF_{m}}(k)) + \tau_s \tag{5}
\]

Thus, the average delay of each mobility management request at all the instances of AMF servers is obtained as (6),

\[
Delay = \frac{1}{N_{AMF}} \sum_{m=1}^{N_{AMF}} D_{AMF_{m}}(k) \tag{6}
\]

The average hops of requests among BSs for each UE due to AMF placement can be obtained as (7):

\[
Hops = \frac{1}{N_{UE}} \sum_{j=1}^{N_{BS}} \sum_{m=1}^{N_{AMF}} \eta_{BS\rightarrow AMF_{j,m}} \lambda_{B_j}(k) H_{BS\rightarrow AMF_{j,m}} \tag{7}
\]

Considering the cost of AMF, which can be modeled by the sum of the number of AMF instances in the network [13], the cost of AMF instances is got as (8),

\[
Cost_{AMF} = c N_{AMF} = c \sum_{j=1}^{N_{BS}} p_j(k) \tag{8}
\]

Where \( c \) is the number of instances in one AMF server.

Let \( Network-Utility \) denote the total utility for AMF placement, which includes the average hops to relay the requests for each UE, the cost of the number of AMF instances and the average delay of requests on AMF, then the normalized network utility is obtained as (9):

\[
Network-Utility = w_1 \frac{Delay}{D_{max}} + w_2 \frac{Hops}{H_{max}} + w_3 \frac{Cost_{AMF}}{A_{max}} \tag{9}
\]

Where \( w_1, w_2 \) and \( w_3 \) are the weight coefficients which can be configured by the network operator, respectively. \( D_{max}, H_{max} \) and \( A_{max} \) are the maximum value of \( Delay \), \( Hops \) and \( Cost_{AMF} \), respectively.

Give a decision time \( k \), the modelling of the placement of AMF is an optimization problem to minimize the \( Network-Utility \) defined in (9), which is formulated as follows,

\[
\min \ Network-Utility \ 
\text{s.t. } C1 : p_j(k) \in \{0,1\}, \forall j \in \{1,2,\cdots,N_{BS}\}
\]

\[
C2 : \sum_{j=1}^{N_{BS}} p_j \geq 1
\]

\[
C3 : \lambda_{AMF_{m}}(k) < c \lambda_{max}, \forall m \in \{1,2,\cdots,N_{AMF}\} \tag{10}
\]

Where \( C1 \) indicates that each BS chooses whether to place the AMF server on its side or not. \( C2 \) means that there exists at least one AMF server in the network, and \( C3 \) represents that the maximum request arrival rate of each AMF server cannot exceed \( c \lambda_{max} \).

Based on the above analysis on \( \lambda_{B_j}(k) \) and \( p_j(k) \), the state, action and reward of the MDP is defined as follows:

- **State**: the state vector can be described as \( s(k) = [\lambda_{B_j}(k), P(k)] \), where \( \lambda_{B_j}(k) \) is a vector representing the mobility request serving rate of every BS at time slot \( k \). \( P(k) \) is a vector denoting the placement state of AMF servers on every BS.
- **Action**: The action space is defined as: \( a(k) = (\alpha_j(k)) \), where \( \alpha_j(k) \in \{0,1\} \), and \( \alpha_j(k) = 1 \) means the AMF server is active at BS \( j \), otherwise, \( \alpha_j(k) = 0 \). The size of the action space is \( 2^{N_{BS}} \), in order to reduce the action space, the agent controls the on-off state of one AMF server in each time step [36].
- **Reward**: After an action is taken in each time step, the agent receives the immediate reward \( r(k) \) named network utility, which is defined as (11)

\[
r(k) = \begin{cases} 
- (Network-Utility) & \text{if } \lambda_{AMF_{m}}(k) < c \lambda_{max}, \\
- Penalty & \text{otherwise}
\end{cases} \tag{11}
\]
where $Penalty$ can be selected as a value that is greater than the maximum value of $Network$ $Utility$ when the arrival rate of each AMF server does not exceed $c\lambda_{max}$.

Assuming that the action is drawn from a stochastic policy: $\pi(a(k) \mid s(k)) = TP(a = a(k) \mid s = s(k))$, in which the transition probability set $TP$ maps each state-action pair $(s(k), a(k))$ to $(s(k+1), a(k+1))$, then the optimization problem can be modeled as a MDP problem: $(S, A, TP, r)$, where $S$ and $A$ are the state space and action space, and $r$ is the reward function. The objective of the MDP problem is to maximize the discounted accumulated reward starting from state-action pair $(s(k), a(k))$, which is defined as the state-value function in (12):

$$V_{AMF}^\pi(s(k)) = \mathbb{E}_\pi \left[ \sum_{n=0}^{\infty} \gamma^n r(k+n) \mid s(k) \right] \quad (12)$$

Where $\gamma \in [0, 1]$ is a discount factor penalizing the future rewards. Therefore, the optimal state value function $V_{AMF}^{\pi^*}$ under the optimal policy $\pi^*$ can be obtained as follows:

$$V_{AMF}^{\pi^*} = \max_{a(k) \in A} \sum_{s(k+1)} TP^{a(k)}_{s(k)\rightarrow s(k+1)} (r(k) + \gamma V_{AMF}^{\pi^*}(s(k+1))) \quad (13)$$

Hence, the value iteration algorithm can be adopted to compute $V_{AMF}^{\pi^*}$. However, the transition probability set $TP$ is not known, it is a model-free problem. Therefore, the remaining task is to find a way of obtaining a good policy for the formulated MDP problem.

4 A DRL based solution to the AMF optimal placement

According to [22], Reinforcement Learning algorithms, especially Q-learning, are widely used to solve MDP problems in the cases when the state space, explicit transition probabilities, and the reward function are not essential. Although Q-learning is also widely adopted in the research of wireless networks for network control without knowing the transition probability and reward function in advance, it has the following three limitations: (1) Traditional Q-learning stores the Q-values in a tabular form. (2) To achieve the optimal policy, Q-learning usually needs to revisit each state-action pair infinitely [22]. (3) The state for Q-learning is often manually defined [36,37].

The above three limitations make it impractical for Q-learning to solve optimization problem with large system state and action spaces. DRL overcomes these problems and has the potential to achieve better performance based on the following three facts. Firstly, DQN is used to store learned Q-values in the form of connection weights between different layers. Secondly, with the help of replay memory and generalization capability by NNs, DRL has good performance via less interaction with complex environments. Thirdly, DR-L avoids manual input by directly learning representation from high dimensional raw network data with DQN [36,37]. Considering these benefits, the NFV orchestrator for AMF uses DRL to learn the control policy of on-off states of AMF servers by interacting with the dynamic environment to minimize the discounted and accumulative network utility for $K$ decision steps, and that is to maximize the long term reward $\mathbb{E} \left[ \sum_{k=0}^{K-1} \gamma^k r(k) \right]$. Based on the above analysis, a DRL based approach to the AMF optimal placement problem (AMF-OP-DRL) is presented.

Based on RL, the Q function in the model is defined as follows in (14),

$$Q_{AMF}^\pi(s(k), a(k)) = \sum_{a(k+1)} TP^{a(k)}_{s(k)\rightarrow s(k+1)} (r(k) + \gamma V_{AMF}^{\pi^*}(s(k+1))) \quad (14)$$

Comparing (12) with (13), the optimal state value function can be obtained by calculating the optimal Q function $Q_{AMF}^{\pi^*}(s(k), a(k))$, the optimal Q function is obtained by the iterative formula defined in (15) [22],

$$Q_{AMF}(s(k), a(k)) \leftarrow (1-\alpha) Q_{AMF}(s(k), a(k)) + \alpha \left[ r(k) + \gamma \max_{a' \in A} Q_{AMF}(s(k+1), a') \right] \quad (15)$$

Where $\alpha$ is the learning rate. Based on (15), the optimal policy can be obtained by computing, storing and updating the Q values of the state-action pairs in the Q-table.

Due to high state space, it is difficult to get enough samples to traverse each state. To solve the problem, we resort to use DNN as an approach to approximate the Q function [37]. Based on DNN, the Q function $Q_{AMF}^{\pi^*}(s(k), a(k))$ can be represented as $Q_{AMF}(s(k), a(k) ; \theta)$, and $\theta$ are the weight sets. According to the neural network structure proposed in [37], the replay memory and target DQN are also considered, since the replay memory breaks the correlations between samples, and the parameters of the
target network is fixed and periodically updated, which stabilizes the training process and accelerates the convergence. Fig. 3 shows the training process of AMF placement.

The goal of the DRL agent is to learn the placement policy of the AMF by interacting with the environment to minimize the discounted accumulated network utility in $K$ decision steps, which can be represented by (16):

$$\max_{\pi} \mathbb{E} \left[ \sum_{k=0}^{K-1} \gamma^k r(k) \right]$$

DRL is trained by iteratively minimizing the loss function at each training step, the loss function can be denoted as (17):

$$L_\theta = \mathbb{E}_{MEM} \left[ (r(k) + \gamma \max_{a(k+1)} Q_{AMF}(s(k+1), a(k+1); \theta^\ast)) - Q_{AMF}(s(k), a(k); \theta) \right]^2$$

(17)

Where $\theta$ are the parameters of the evaluation network, and $\theta^\ast$ are the parameters of the target network. MEM refers to the replay memory. The detailed procedure of the proposed AMF-OP-DRL algorithm is shown in the Algorithm 1.

**Algorithm 1** Optimal Placement of AMF based on DRL(AMF-OP-DRL)

1: **Input**: $\lambda_P(k), P(k)$.
2: Offline training:
3: Initialize the evaluation network with weights $\theta$.
4: Initialize the target network with weights $\theta^\ast = \theta$.
5: Initialize the replay memory $MEM$ to capacity $N$, each element consists of the state, action, reward and next state.
6: for epoch = 0, 1, ..., COUNTS − 1 do
7: Initialize the initial observation and pre-process the initial observation to the beginning state $s_0$.
8: for decision steps $k = 0, 1, ..., K - 1$ do
9: Select an action $a(k)$ randomly with probability $\epsilon$ or $a(k) = \arg\max_{a \in \mathcal{A}} Q_{AMF}(s(k), a; \theta)$ with $1 - \epsilon$.
10: According to executed action $a(k)$ and user movement around the network, the DRL agent observes the reward $r(k)$ and the next state $s(k+1)$.
11: Store the transition $(s(k), a(k), r(k), s(k+1))$ into the replay memory $MEM$.
12: Sample a mini-batch transitions randomly from $MEM$ and perform a gradient update on the loss function defined in (15) with respect to $\theta$.
13: Reset $\theta^\ast = \theta$ every COUNTS steps.
14: end for
15: **Output**: the placement policy of AMF: $P(k + 1)$.

Where COUNTS is the parameter for epochs, and $\theta^\ast$ is reset by $\theta$ for every COUNTS steps.

### 5 Results and discussion

In this section, the performance of the AMF-OP-DRL algorithm is evaluated by simulation compared with several baseline algorithms, including simulation parameters in 5.1 and performance evaluation in 5.2.

#### 5.1 Simulation parameters

In the simulation, 12 BSs are distributed in a rectangular area of size 560m × 390m, with $4 \times 3$ grid. The coverage area of each BS is 140m × 130m [33].

##### 5.1.1 Mobility management related parameter settings

There are 12000 UEs moving in the area, user mobility pattern follows Random Walk (RW) model, in which each user choose a random direction from $[0, 2\pi]$ and the speed ranges between $[v_{min}, v_{max}]$. In the simulation, the direction is simplified as four directions, at each time step, every UE moves in four directions with the same probability [38]. The user speed ranges between [1, 3] m/s.

The arrival rate of mobility management requests follows Poisson distribution. The arrival rate of HR is 0.0012 procedures/s/user, the arrival rate of SR is 0.0045 procedures/s/user and the arrival rate of SRR is 0.0045 procedures/s/user [31]. Each AMF server is equipped with fixed number of AMF instances.

##### 5.1.2 Parameter settings of the AMF $M/M/c$ queuing system

According to the modeling of AMF $M/M/c$ queuing system, when the queuing system is stable, $Sa < 1$.

Based on the formula (4), the impact of $c$, $\mu$ to the average waiting time is presented in Fig. 4 and Fig. 5, respectively.

From Fig. 4 and Fig. 5, it indicates that when $\lambda$ changes, the value of $Sa$ is different depending on $c$ and $\mu$. Based on the constraints of the average waiting time of requests, the parameter of $c$, $\mu$ and $\lambda_{max}$ can be obtained.

In the simulation, we choose $\mu = 10$ procedures/s/user, $c = 5$ and $\lambda_{max} = 9$ procedures/s, then $Sa = 0.9$. That is to say, whenever the arrival rate is greater than 45 procedures/s, the AMF server cannot process those mobility management requests beyond its processing capacity, the system is saturated.

The cost of one AMF instance is set as 1. $w_1$, $w_2$ and $w_3$ are set as 1, 1 and 5, respectively. According to (10), in order to ensure that the reward value is small enough when the placement decision is overloaded, we choose the threshold of $Penalty$ when the arrival rate of each AMF server is $c\lambda_{max}$, namely $Penalty$ is set as 7 in the simulation. It can be set larger than the threshold as well.
5.1.3 Parameter settings of DQN
The DQN is enabled with an input layer, a hidden layer and an output layer. There are 24 neurons in the input layer, 16 neurons in the hidden layer and 13 neurons in the output layer [39]. Table 2 shows the parameter settings of DQN.

The detailed layout architecture of the DNN in the simulation is given in Fig. 6.

5.2 Performance evaluation of AMF-OP-DRL algorithm
In this subsection, the performance of AMF-OP-DRL algorithm is evaluated, including the convergence performance of AMF-OP-DRL, the impact of arrival rate to the number of AMF servers, the impact of weight coefficients to Delay, Hops and AMF cost, as well as the performance comparison with baseline algorithms.

5.2.1 Convergence performance of AMF-OP-DRL
The convergence performance of AMF-OP-DRL algorithm based on DQN under different batch sizes are presented in Fig. 7 when the batch size is 32, 64 and 128, respectively. The performance is the best when the batch size is 32. This is because when the batch size is small, more randomness is introduced and longer training time is needed to achieve convergence, while large batch size tends to get local optimal result.

The impact of DQN learning rate to convergence performance is shown in Fig. 8. The learning rate is set as 0.01, 0.001 and 0.0001, respectively. The DQN achieves the optimal performance when the learning rate is 0.01.

Since the complexity of the DRL algorithm is related to the number of BSs in the network, in order to investigate the convergence performance of the algorithm when the number of BS is large, the number of the BSs is set as 56 and the number of UEs is 12000 in the simulation. The arrival rate of HR, SR and SRR is the same as that in 5.1.1. The number of neurons in the input layer of the DQN is adjusted to 112, and the number of neurons in the output layer is set as 57. There are two hidden layers and each layer has 60 neurons. The hyper parameters of DQN are adjusted according to the convergence performance of the algorithm. The convergence performance is shown in Fig. 9. The algorithm begins to converge when the epoch is around 700. It indicates that AMF-OP-DRL algorithm achieves good performance when the number of BS is large.

5.2.2 The impact of request arrival rate to the number of AMF servers
The impact of request arrival rate to the number of AMF servers is evaluated in Fig. 10, Fig. 11 and Fig. 12, respectively, when \( w_1 : w_2 : w_3 = 1 : 1 : 5 \).

Fig. 10 shows the number of AMF servers under different HR arrival rate with different number of VNF instances on each AMF sever, when the HR arrival rate varies from 0 to 0.007 procedures/s/user, the arrival rate of SR and SRR is 0.0045 procedures/s/user, and the number of AMF instances changes from 4 to 6. It shows that the number of AMF servers increases with the increase of HR arrival rate, and the optimal number of AMF servers is different. When the number of instances is 4 and 5, the number of AMF servers is more sensitive to the increase of HR arrival rate than that when the number of instances is 6, because the processing capacity of AMF servers with 4 instances and 5 instances is less than that with 6 instances. There is a cross point between the number of instances is 5 and 6 on AMF servers when the arrival rate of HR event arrival rate is 0.006 procedures/s/user, which depends on the processing capacity of AMF servers as well as different weights of the optimization objectives. In the case when the number of instances is 6 on AMF servers, when the HR arrival rate increases from 0.005 to 0.006, it reveals that the agent tends to obtain the optimal network utility caused by the increase of the arrival requests with the tradeoff of adding one AMF server.

Fig. 11 is the simulation result of the impact of SR arrival rate to the number of AMF servers, the arrival rate of HR is set as 0.0012 procedures/s/user and SRR is 0.0045 procedures/s/user. Fig. 12 shows the impact of SRR arrival rate to the number of AMF servers when the arrival rate of HR is 0.0012 procedures/s/user and the arrival rate of SR is 0.0045 procedures/s/user. As shown in Fig. 11 and Fig. 12, when the arrival rate changes from 0 to 0.004 procedures/s/user, the number of AMF servers with 5 and 6 instances is the same. Since the load of the AMF servers in both cases is not saturated, the number of AMF servers keeps the same when the arrival rate varies from 0.002 procedures/s/user to 0.004 procedures/s/user. When the arrival rate is 0.005 procedures/s/user, for the AMF servers with 5 instances, the saturation rate is larger than that of those AMF servers with 6 instances, the number of AMF servers becomes different.

5.2.3 The impact of weight coefficients to Delay, Hops and AMF cost
The impact of the weight coefficients to the AMF cost is shown in Fig. 13, in which each point in the figure is obtained from the average value of each training step after the algorithm converges. It is shown that the cost of AMF is the lowest when \( w_1 : w_2 : w_3 = 1 : 1 : 5 \). Since the proportion of the AMF cost in network utility function is the highest, the learning strategy of the
DRL agent is to place the least number of AMF servers under the constraints. When \( w_1 : w_2 : w_3 = 1 : 5 : 1 \), the DRL agent reduces the wired hops to AMF as much as possible, and the AMF cost changes slightly when the user's HR arrival rate increases. That is to say, the policy is preferred to place the AMF servers on every BS, and the user's mobility management requests are processed by the AMF on its serving BS. When \( w_1 : w_2 : w_3 = 5 : 1 : 1 \), the AMF cost changes between the two solutions with \( w_1 : w_2 : w_3 = 1 : 5 : 1 \) and \( w_1 : w_2 : w_3 = 1 : 1 : 5 \), because the average delay of requests at AMF is affected by the arrival rate and the serving rate of the AMF instances, the number of AMF servers needs to achieve a balance.

Fig. 14 is the simulation result on the impact of weight coefficients to the average wired hops. When \( w_1 : w_2 : w_3 = 1 : 1 : 5 \) and \( w_1 : w_2 : w_3 = 1 : 5 : 1 \), the wired hops changes slightly when the number of AMF servers is increasing to cope with the increasing request rate. The average delay of mobility management requests at AMF instances and their serving capacity. The average delay of mobility management request is set higher due to the massive number of UEs distributed in the radio access network.

The impact of weight coefficients to the average waiting time of mobility management request is presented in Fig. 15. When \( w_1 : w_2 : w_3 = 1 : 1 : 5 \), the curve fluctuates because the DRL agent adjusts the number of AMF servers with the increase of HR arrival rate. Since the number of AMF instances on each AMF server is fixed, the fluctuation depends on the number of AMF instances and their serving capacity. The average delay of mobility management requests at AMF increases slightly when \( w_1 : w_2 : w_3 = 1 : 5 : 1 \) and \( w_1 : w_2 : w_3 = 5 : 1 : 1 \), because the increase of the request arrival rate is smaller compared with the serving rate of each AMF server.

With the above analysis of Fig. 13, Fig. 14 and Fig. 15, it indicates that the AMF placement strategy is affected by the weight coefficients of the network utility. Therefore, the weight coefficients can be adjusted according to different service requirements on mobility management. For example, the weights of the wired hops and the average delay of requests at AMF server can be set higher in the URLLC service scenario, while in some mMTC scenario, the weights of the AMF cost is set higher due to the massive number of UEs distributed in the radio access network.

5.2.4 Performance comparison with baseline algorithms

In this subsection, the performance of AMF-OP-DRL algorithm is evaluated. To the best of our knowledge, there is no research issues on AMF optimal placement in the scenario of heterogeneous radio access network yet. Therefore, the proposed AMF-OP-DRL algorithm is evaluated comparing with baseline algorithms including AMF-OP-Greedy, AMF-OP-HBP, AMF-OP-Random and AMF-OP-KAB, which are illustrated as follows,

- **AMF-OP-Greedy**: It traverses all combinations of \( N_{AMF} \) BSs and selects the combination with the lowest accumulated network utility, where \( N_{AMF} \) is the number of BSs selected to place AMF servers, which is learned by the DRL agent at each step.
- **AMF-OP-HBP**: In the algorithm, the AMF servers are placed on BSs ranked top \( N_{AMF} \) in arrival rate. The network utility is calculated according to the learned AMF placement policy.
- **AMF-OP-Random**: The AMFs are randomly placed on \( N_{AMF} \) BSs. The network utility is got based on the selected AMF placement policy.
- **AMF-OP-KAB**: The algorithm is based on the K-armed bandit algorithm. In every step, there are \( N_{AMF} \) actions to choose. Each action corresponds to whether the AMF server is placed on a BS and each action has an average reward value. The agent successively takes an action and then receives a random reward, aiming at maximizing the accumulated reward [22]. For every training step, the AMF placement policy is obtained by the \( \epsilon \)-greedy strategy. That is to say, the algorithm randomly chooses an action when the random number is less than the probability \( \epsilon \), otherwise, it selects the action with the maximum reward value. \( \epsilon \) is set as 0.1 in the simulation.

The comparison result is shown in Fig. 16, in which each point is the average value of 100 epochs. It reveals that the accumulated network utility of AMF-OP-HBP and that of AMF-OP-Random are more than twice than that of AMF-OP-DRL algorithm. The AMF-OP-HBP is even worse than AMF-OP-Random, the reason is that AMF-OP-HBP chooses the BSs with the highest arrival rate and brings about greater probability of overload.

AMF-OP-KAB has better performance than that of AMF-OP-HBP and AMF-OP-Random, but it is worse than AMF-OP-DRL. Because the number of state-action pairs is large in AMF-OP-KAB, its average reward function of actions does not fit well compared with the DRL. In AMF-OP-DRL, since DNN is used to approximate the Q function, the performance of the DRL based control algorithm is much closer to AMF-OP-Greedy, but the greedy algorithm traverses all possible combinations of AMF placement, which brings about high time complexity when the number of BS is large.

From the above analysis, AMF-OP-DRL algorithm outperforms the four baseline algorithms.
Regarding the complexity of the proposed algorithm, if there are n BSs in the system, then for the offline trained neural network model, the computational complexity of the AMF-OP-Greedy is $O(n^2)$, while the computational complexity of the AMF-OP-Greedy is $O(n^m \log(n^m))$, where m is the number of AMF servers to be placed.

6 Conclusion

In this paper, a DRL based algorithm is proposed to optimize the placement of AMF in the heterogeneous network to minimize the long-term network utility including the average delay of mobility management requests, the average wired hops to relay mobility management of requests and the cost of AMF instances. The time-varying features including user mobility and the arrival rate of user mobility management requests are considered in the model of the DRL environment. The convergence performance of the algorithm is evaluated under various batch sizes, learning rates and the number of BSs. The impact of the IR/SR/SRR arrival rate and the weight coefficients to the performance of AMF-OP-DRL are investigated. The performance of AMF-OP-DRL is also evaluated compared with four baselines including AMF-OP-Greedy, AMF-OP-HBP, AMF-OP-Random and AMF-OP-KAB. Simulation results show that it outperforms those four baseline algorithms. In the future research work, the joint optimization of the auto-scaling and optimal placement of AMF would be addressed.

Funding

The funding source is the same as described in the acknowledgements.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Author’s information

Hao Jin received the Ph.D. degree from Beijing University of Posts and Telecommunications in 1996. She is currently an associate professor at BUPT, Beijing, China. Her research interests include future network architecture, optimization of mobile wireless communication and AI based MEC.

Wenhe Pang received the BEng degree from Chongqing University of Posts and Telecommunications in 2017. He is a graduate student for master degree in the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include mobility management, MEC and NFV.

Chenglin Zhao received the bachelor’s degree in radio technology from Tianjin University in 1986 and the master’s degree in circuits and systems and the Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1993 and 1997, respectively. He is currently a Professor at BUPT. His current research interests include emerging technologies of short-range wireless communication, cognitive radios, MEC and industrial IoT.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grants 61471062, 61431008 and 61971061, National Science and Technology Major Project (Grant 2017ZX03001014), and also by Beijing Natural Science Foundation (Grant L182039).
13. Y. Ren, T. Phung-Duc, Y. Liu, J. Chen, Y. Lin, in IEEE 7th International Conference on Cloud Networking (CloudNet). ASA: Adaptive VNF scaling algorithm for 5G mobile networks, (Tokyo, Japan, 2018), pp. 1-4

14. H. Jin, Y. Jin, H. Lu, C. Zhao, M. Peng, NFF and SFC: A case study of optimization for virtual mobility management. IEEE J. Sel. Areas Commun. 36(10), 2318-2332 (2018)

15. I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, D. Darche, in 2018 IEEE Annual Consumer Communications & Networking Conference (CCNC). On the scalability of 5G core network: The AMF case, (Las Vegas, NV, USA, 2018), pp. 1-6

16. C. Li, J. Li, Y. Li, Z. Han, Bayesian coalition formation game for virtual 5G core network functions. IEEE Access. 7, 29805-29817 (2019)

17. P. Tang, F. Li, W. Zhou, W. Hu, L. Yang, in IEEE Global Communications Conference (GLOBECOM). Efficient auto-scaling approach in the telco cloud using self-learning algorithm, (San Diego, CA, USA, 2015), pp. 1-6

18. I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, P. Bertin, Improving traffic forecasting for 5G core network scalability: A machine learning approach. IEEE Netw. 32(6), 42-49 (2018)

19. I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, C. Viho, D. Darche, in IEEE Global Communications Conference (GLOBECOM). Smart scaling of the 5G core network: An RNN-based approach, (Abu Dhabi, United Arab Emirates, 2018), pp. 1-6

20. I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, C. Viho, D. Darche, in IEEE Conference on Standards for Communications and Networking (CSCN). An efficient and lightweight load forecasting for proactive scaling in 5G mobile networks, (Paris, France, 2018), pp. 1-6

21. Y. Sun, M. Peng, Y. Zhou, Y. Huang, S. Mao, Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surveys Tutor. 21(4), 3072-3108 (2019)

22. R. Sutton A. Barto, Reinforcement learning: An introduction. MIT press. (1998)

23. Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, L. Wang, Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. IEEE Veh. Technol. Mag. 14(2), 44-52 (2019)

24. G. Sun, Z. T. Gebrekidan, G. O. Boateng, D. Ayepah-Mensah, W. Jiang, Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks. IEEE Access. 7, 45758-45772 (2019)

25. M. Nakanoya, Y. Sato, H. Shimoshinji, in IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Environment-adaptively sizing and placement of NFV service chains with accelerated reinforcement learning. (Arlington, VA, USA, 2019), pp. 36-44

26. Y. He, C. Liang, F. R. Yu, V. C. M. Leung, in IEEE Global Communications Conference (GLOBECOM). Integrated Computing, Caching, and Communication for Trust-Based Social Networks: A Big Data DRL Approach, (Abu Dhabi, United Arab Emirates, 2018), pp. 1-6

27. B. Matthiesen, A. Zappone, E. A. Jorswieck M. Debbah, in IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). Deep Learning for Real-Time Energy-Efficient Power Control in Mobile Networks, (Cannes, France, 2019), pp. 1-5

28. C. Liang, F. R. Yu, Wireless Network Virtualization: A Survey, Some Research Issues and Challenges. IEEE Commun. Surveys Tutor. 17(1), 358-380 (2015)

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

30. M. R. Sama, X. An, Q. Wei, S. Beker, in IEEE Wireless Communications and Networking Conference (WCNC). Reshaping the mobile core network via function decomposition and network slicing for the 5G era, (Doha, Qatar, 2016), pp. 1-7

31. J. Prados-Garzon et al., in IEEE International Conference on Communications Workshops (ICCW). Analytical modeling for virtualized network functions, (Paris, France, 2017), pp. 979-985

32. V. Nguyen, K. Grinnemo, J. Taheri, A. Brunstrom, in IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). On load balancing for a virtual and distributed MME in the 5G core, (Bologna, Italy, 2018), pp. 1-7

33. J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, J. M. Lopez-Soler, Modeling and dimensioning of a virtualized MME for 5G mobile networks. IEEE Trans. Veh. Technol. 66(5), 4383-4395 (2017)

34. I. Kleinrock, Queueing Systems, Volume I: Theory. Hoboken, NJ, USA. (1975)

35. M. Zukerman, Introduction to queueing theory and stochastic teletraffic models. arXiv preprint arXiv:1307.2968 (2019)

36. Y. Sun, M. Peng, S. Mao, Deep reinforcement learning-based mode selection and resource management for green fog radio access networks. IEEE Internet Things J. 6(2), 1960-1971 (2019)

37. V. Mnih et al., Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533 (2015)

38. Z. Wang, L. Li, Y. Xu, H. Tian, S. Cui, Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. IEEE Internet Things J. 5(6), 4296-4307 (2018)

39. I. Goodfellow, Y. Bengio, A. Courville, Deep learning. MIT press. (2016)

Figures
Fig. 1 AMF placement in heterogeneous radio access network
Fig. 2 AMF orchestration based on MANO

Fig. 3 Training process of AMF Placement based on Deep Reinforcement Learning
Fig. 4 The impact of arrival rate of mobility management events to the average waiting time ($c = 5$)
Fig. 5 The impact of the arrival rate of mobility management events to the average waiting time ($\mu = 10$)
Fig. 6 The layout architecture of the DNN
Fig. 7 Convergence performance of AMF-OP-DRL under different batch sizes

- Batch size = 32
- Batch size = 64
- Batch size = 128
Fig. 8 Convergence performance of AMF-OP-DRL under different learning rates
Fig. 9 Convergence Performance of AMF-OP-DRL algorithm ($N_{BS} = 56$)
Fig. 10 The impact of the arrival rate of HR event to the number of AMF servers
The number of AMF servers

Fig. 11 The impact of the arrival rate of SR event to the number of AMF servers
Number of instances on AMF = 4
Number of instances on AMF = 5
Number of instances on AMF = 6

Fig. 12 The impact of the arrival rate of SRR event to the number of AMF servers
Fig. 13 The impact of weight coefficients to AMF cost
Fig. 14 The impact of weight coefficients to wired hops
Fig. 15 The impact of weight coefficients to the average waiting time of requests.
Fig. 16 Performance comparison with baseline algorithms
Table 1 Summary of key notations

| Notations      | Description                                                                 |
|----------------|------------------------------------------------------------------------------|
| \( N_{BS} \)  | Number of Base Stations (BSs)                                                |
| \( N_{UE} \)  | Number of User Equipments (UEs)                                              |
| \( N_{AMF} \) | Number of running AMF servers                                                |
| \( k \)       | The decision time step of DRL agent                                          |
| \( \lambda_i (k) \) | The mobility management request arrival rate from UE \( i \) at time step \( k \) |
| \( \lambda_{SR_i} (k) \) | The Service Request (SR) arrival rate of UE \( i \) at time step \( k \)   |
| \( \lambda_{SRR_i} (k) \) | The Service Release Request (SRR) arrival rate of UE \( i \) at time step \( k \) |
| \( \lambda_{HR_i} (k) \) | The Handover Request (HR) arrival rate of UE \( i \) at time step \( k \) |
| \( H_{BS-BS_{i,j}} \) | The wired hops between BS \( i \) and BS \( j \)                           |
| \( c \)       | The number of instances in one AMF server                                    |
| \( \tau_s \)  | The average serving time of mobility management request                      |
| \( \tau_q \)  | The average queuing time of mobility management request                      |
| \( \lambda \) | The request arrival rate of the AMF queuing system                           |
| \( \mu \)     | The serving rate of one AMF instance                                         |
| \( \lambda_{max} \) | Maximum request arrival rate of one AMF instance                             |
| \( U_{i,j} (k) \) | The serving indicator between UE \( i \) and BS \( j \) at time step \( k \) |
| \( p_j (k) \) | The binary indicator for placement of active AMF server on BS \( j \) at time step \( k \) |
| \( \lambda_{B_j} (k) \) | The arrival rate of the mobility management request at BS \( j \) at time step \( k \) |
| \( \lambda_B (k) \) | The vector representing the mobility management request arrival rate of every BS at time step \( k \) |
| \( \eta_{BS-AMF_{j,m}} \) | The proportion of BS \( j \)'s mobility management requests processed by AMF \( m \) |
| \( \lambda_{AMF_m} (k) \) | Total mobility management request rate processed by AMF \( m \) at time step \( k \) |
| \( w_1, w_2, w_3 \) | The weight coefficients of Delay, Hops and AMF cost                        |
| \( D_{max}, H_{max}, A_{max} \) | The maximum value of Delay, Hops and AMF cost                               |
| \( TP \)      | Transition probability set of state-action pairs                            |
| \( \gamma \)  | Discount factor penalizing the future reward                                |
Table 2 DQN Parameters

| Parameters                                         | Value          |
|----------------------------------------------------|----------------|
| The size of Replay memory $MEM$                    | 2000           |
| The initial memory $N$ populate $MEM$              | 1000           |
| Activation function                               | Relu           |
| The learning rate of RMSProp optimizer             | 0.01, 0.001, 0.0001 |
| Discounted factor for future reward               | 0.99           |
| Batch size for DQN update                          | 32, 64, 128    |
| The number of time steps to update target networks | 200            |
| $w1, w2, w3$                                       | 1, 1, 5        |
| $D_{max}$                                          | 0.2525, 0.3724 |
| $H_{max}$                                          | 1.5            |
| $A_{max}$                                          | 60, 72         |
Figures

Figure 1
AMF placement in heterogeneous radio access network

Figure 2
AMF orchestration based on MANO
AMF orchestration based on MANO

**Figure 3**

Training process of AMF Placement based on Deep Reinforcement Learning
Figure 4

The impact of arrival rate of mobility management events to the average waiting time
Figure 5

The impact of the arrival rate of mobility management events to the average waiting time
Figure 6

The layout architecture of the DNN

Figure 7

Graph showing the accumulated network utility over training epochs for different batch sizes.
Convergence performance of AMF-OP-DRL under different batch sizes

Figure 8

Convergence performance of AMF-OP-DRL under different learning rates
Figure 9

Convergence Performance of AMF-OP-DRL algorithm
Figure 10
The impact of the arrival rate of HR event to the number of AMF servers

Figure 11
The impact of the arrival rate of SR event to the number of AMF servers
Figure 12

The impact of the arrival rate of SRR event to the number of AMF servers

The impact of the arrival rate of SRR event to the number of AMF servers.
Figure 13
The impact of weight coefficients to AMF cost

Figure 14
The impact of weight coefficients to wired hops
Figure 15

The impact of weight coefficients to the average waiting time of requests.
Figure 16

Performance comparison with baseline algorithms