DRL-Assisted Fine-Grained Function Placement and Routing of 5G RAN Slice with Reuse Scheme in Elastic Optical Networks

Yunwu Wang [1,2], Min Zhu [1,2*], Xiaofeng Cai [1], Jiahua Gu [1,2], Xiang Liu [1,2], Jiao Zhang [1,2]
[1] National Mobile Communications Research Laboratory, Southeast University, Nanjing, 210096, China
[2] Purple Mountain Laboratories, Nanjing, 211111, China, [*] *Corresponding author e-mail address: minzhu@seu.edu.cn*

***Abstract*—The fine-grained functional split is an effective way to solve the baseband function processing centralization and optical bandwidth saving in radio access networks (RANs). In this paper, to improve computing resource utilization, we investigate how to realize the fine-grained function placement and routing of 5G RAN slice with function reuse scheme in elastic optical networks (EONs). We first formulate a mixed integer linear programming (MILP) model to solve the problem exactly. The main optimization goal in the MILP model is to jointly minimize the average cost of computing, bandwidth resources and end-to-end latency. Then, a heuristic-assisted deep reinforcement learning (HA-DRL) algorithm is proposed to obtain a near-optimal solution. In particular, the longest common subsequence-based path policy is utilized in the DRL to reduce the size of the action space and accelerate the training process. Finally, we evaluate the proposed MILP model and HA-DRL algorithm via extensive simulation. The results show that the proposed MILP model and HA-DRL algorithm outperform the benchmarks in terms of average cost, including the number of used processing pools (PPs), maximum frequency slot index (MFSI) on the lightpath and end-to-end latency of each slice request.**

Keywords—Flexible functional split, fine-grained function placement and routing, 5G RAN slice, heuristic-assisted DRL.

I. INTRODUCTION

To support the service scenarios of mobile applications in terms of enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (uRLLC) and massive machine type communication (mMTC), next-generation radio access network (NG-RAN) architecture has been introduced [1]. These heterogeneous services will bring new challenges such as bandwidth, latency and networking flexibility for the NG-RAN, which accelerates its evolution towards fine-grained units (FUs) [2]. In the 5G FU-based RAN architecture, the baseband unit (BBU) is further split into a set of FUs, taking into account the functional split options of 3GPP standard [3]. During the FU baseband processing function (BPF) placing process, the BPFs of the same type in different requests can be placed to the same processing pools (PPs) to share the common instance (i.e., FU) activated. Hence, the function reuse (FuRe) scheme can significantly reduce the amount of the required FUs activated on PPs. However, it inevitably increases the instance processing latency due to instance competition mentioned in our previous work [4]. Therefore, it is desirable to design an effective baseband function placement and routing (BFP&R) with the FuRe scheme in 5G FU-based RAN architecture, which not only reduces the consumed instance resources but also satisfies the end-to-end latency requirement of each slice request.

Some recent works have investigated the BFP&R issue in NG-RAN architecture through heuristic algorithms. In [5], the effective management policy for the agile distributed unit (DU) and centralized unit (CU) deployment was investigated, where a mixed integer linear programming (MILP) model and a graph-based heuristic algorithm were proposed to achieve energy-efficient BFP&R. Besides, the authors in [2] illustrated that the FU-based RAN architecture could benefit the baseband processing centralization and optical bandwidth saving for the BFP&R issue. However, the above works do not consider the FuRe scheme for the BFP&R in 5G RAN architecture, and hence the instance processing latency is simply set as a fixed value. Recently, deep reinforcement learning (DRL) method has been successfully applied to solve multi-resource management problems. The authors in [6] adopted the DRL to generate the BFP&R policy. This work mainly optimized the resource allocation while ignoring the end-to-end latency of slice requests. In [7], the DRL-assisted BFP&R was further considered the end-to-end latency of the 5G RAN slices over traditional metro-aggregation wavelength division multiplexing (WDM) networks. However, the DRL-assisted fine-grained BFP&R with the FuRe scheme in 5G FU-based RAN architecture hasn't yet been discussed.

In this paper, the fine-grained BFP&R policy of 5G RAN slice with the FuRe scheme was addressed by leveraging MILP model and the proposed DRL-based algorithm in the elastic optical networks (EONs). The main contributions of this paper are listed as follows. 1) This paper investigates resource-efficient fine-grained BFP&R of 5G RAN slice with the FuRe scheme, jointly considering fine-grained BFP&R with the FuRe scheme, PP selection, and end-to-end latency control. 2) We formulate a MILP model for the fine-grained BFP&R with the FuRe scheme in EONs, where the FU processing latency caused by the FuRe scheme is solved by M/M/1 model, which was not considered by most existing works. 3) We propose a heuristic-assisted DRL (HA-DRL) fine-grained BFP&R algorithm that minimizes the required computing and bandwidth resources, meanwhile satisfying the given end-to-end latency requirements. 4) The results show that our proposed algorithm can achieve higher resource efficiency and minimize end-to-end latency, compared with the benchmarks.

II. ARCHITECTURE AND PROBLEM FORMULATION

A. Network Scenario

We consider flexible EONs as the 5G RANs to support "any-to-any" connection between radio units (RUs)/PPs and PPs. The EON is presented as a directed graph $G(R, L)$,

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10278613

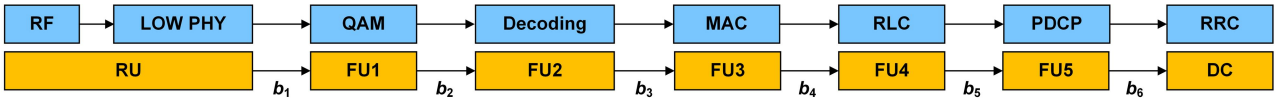

Fig. 1. 5G FU-based RAN architecture, where split options follow the 3GPP specification [3].

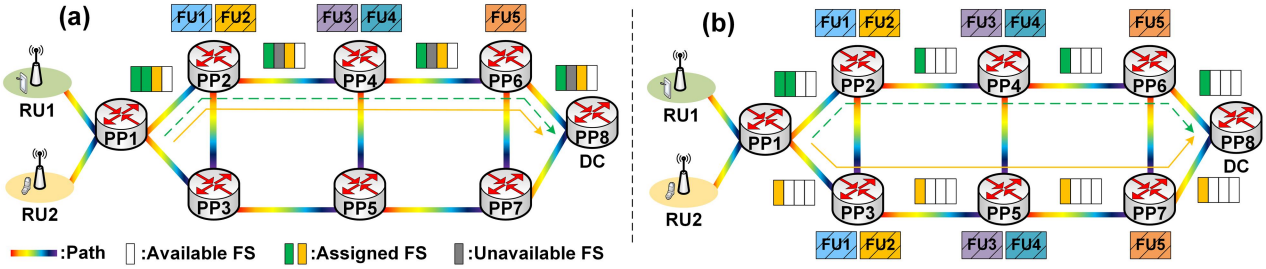

Fig. 2. An example of RU-DC deployment in EONs: (a) Deployment policy 1; (b) Deployment policy 2.

where R and L denote the sets of PP nodes and fiber links, respectively. The upstream and downstream have a similar processing function according to the 5G RAN function split from 3GPP specification [1]. Therefore, in this paper, we only focus on the FU BFP&R for data plane in upstream. As shown in Fig. 1, in 5G FU-based RAN architecture, a 5G RAN slice contains five FUs (i.e., FU1-FU5), where FU1 is responsible for demodulation, FU2 is responsible for channel decoding, FU3 is responsible for media access control (MAC) that multiplexes data from different radio bearers, FU4 is responsible for radio link control (RLC) that includes segmentation and reassembly for higher layers, and FU5 is responsible for packet data convergence protocol (PDCP) that addresses problem of security. And b_1, b_2, b_3, b_4, b_5 and b_6 denote the bandwidth requirement before steering through the RU, FU1, FU2, FU3, FU4 and FU5, respectively.

B. Computational Complexity

To manage computational resource, we calculate the computational complexity of FU. The computational complexity of FU is expressed in Giga operations per second (GOPS), which can be defined as [2]:

$$C_{FU_i} = \alpha_i \cdot (3 \cdot A + A^2 + 1/3 \cdot M \cdot C \cdot Ly) \cdot PRB / 5 \qquad (1)$$

where α_i is the FU factor, A is the number of utilized antenna, M is the modulation bits, C is the coding rate, Ly is the number of multiple input multiple output (MIMO) layers, and PRB is the number of physical resource blocks. The M and C are determined from modulation and coding scheme (MCS) in [8].

C. Bandwidth Requirement

After steering through a FU instance, the bandwidth requirement of a request would change. The number of required frequency slots (FSs) can be obtained by $\lceil b_i / (ML_m \times B_{FS'}) \rceil$, $i \in \{1,\cdots,5\}$, where $B_{FS'}$ is the bandwidth for each FS, i.e., 12.5GHz [4], and ML_m is the level of the modulation format $m \in [1, 2, 3, 4]$, corresponding to binary phase-shift keying (BPSK), quaternary PSK (QPSK), 8-quadrature amplitude modulation (QAM), and 16-QAM, respectively. And b_i can be calculated by parameters in [2].

D. Latency Model

We calculate the required latency from five aspects: **1) Transmission latency:** The transmission latency on fiber links which is linear with the length of fiber links. **2) OEO switching latency:** For an arriving 5G RAN slice request via a light-path, optical signal should be switched to the electronic domain before its BFP in the PP, which results the latency of optical-electric-optical (OEO) and electric switching (i.e., $T_{oeo} = 20us$ [9]). **3) Processing latency:** We adopt the processing latency model present in [4]. By assuming the request traffic as the input queue and the processing central processing unit (CPU) of each FU instance as the single server, the processing of request traffic data in each FU can be modeled as an M/M/1 queue. Therefore, the average processing latency of each FU s in PP r can be calculated as $T_{s,r} = 1/(CK_s - U_{s,r} \times C_{reach})$, $\forall s$, r, where CK_s is the capacity limit of FU s, $U_{s,r}$ is the number of reuse of FU s in PP r, and C_{reach} is the reaching capacity of the slice requests. **4) Virtualization platform latency:** We assume that starting a FU instance requires a virtualization platform latency T_v (i.e., $T_v = 52us$ [2]). Since the FU instance is a kind of reusable resource, the system will not take action to start a new instance, where the FU instance of same type has already been started in the PP. **5) Interface encapsulation latency:** A reconfigurable and general interface for data encapsulation is introduced in our network architecture. In this paper, we evaluate the interface encapsulation/de-capsulation latency referred to enhanced common public radio interface (eCPRI) [2]. The general interface latency is expressed as: $T_{encap} = L_p / b_i$, where L_p denotes the length of a frame.

E. Problem Formulation

In 5G FU-based RAN architecture, the FUs can be placed into the same or different PPs and chained in the predefined order: [RU, FU1, FU2, FU3, FU4, FU5, data center (DC)] to generate a service chain for the BFP of the slice request. Thus, a RU-DC slice request is denoted as: [RU, (b_1, FU1), (b_2, FU2, (b_3, FU3), (b_4, FU4), (b_5, FU5), b_6, DC]. Fig. 2 shows an example of deploying the RU-DCs in a 8-node EONs. Assume there are two requests Req_1[1, (2, FU1), (1, FU2), (1, FU3), (1, FU4) (1, FU5), 1, 8] and Req_2[1, (1, FU1), (1, FU2), (1, FU3), (1, FU4), (1, FU5) 1, 8]. Fig. 2(a) shows a policy where Req_1 takes path 1→2→4→6→8 and deploys FU1 and FU2 on PP2, FU3 and FU4 on PP4 and FU5 on PP6. Req_2 goes 1→2→4→6→8 and reuses FU1, FU2 and FU3, FU4 and FU5 that have been deployed on PP2, PP4 and PP6, respectively. The FuRe scheme of FUs inevitably increased processing latency for Req_1 and Req_2. To reduce the

processing latency, another policy is presented in Fig. 2(b), where Req_2 takes path 1→3→5→7→8 and instantiates a new FU1 and FU2 on PP3, a new FU3 and FU4 on PP5 and a new FU5 on PP7. In this case, two requests need to be deployed in more PPs (i.e. 6 PPs). Therefore, for the efficient resource utilization and satisfactory end-to-end latency, it is necessary to design a well BFP&R policy with the FuRe scheme that maximizes the resource utilization while ensuring the required end-to-end latency.

III. MILP Formulation

In this section, we formulate a MILP model to determine the placement of FUs with the FuRe scheme to minimize the computing, bandwidth resources and the end-to-end latency.

(A) Notations:

B, L	Set of RUs and optical links
R,F	Set of PPs, and FSs in each link
S	Set of service functions (SFs) (e.g., s = 1 for RU, s = 2 for FU1,…, s = 6 for FU5, s = 7 for DC)
C_{max}	Computational capacity of each PP r
T_b	End-to-end latency requirement of RU b
TC_b	Transmission latency from RU b to its directly connected PP r
TE_s	Encapsulation latency after SF s processed
T_v	Latency of virtualization platform in each PP r
$T_{i,j}$	Transmission latency of link $e(i,j)$, $i,j \in R$
T_{oeo}	Latency of OEO operation
$T_{s,r}$	The processing latency of SF s in PP r
$T_{b,s}$	The processing latency of SF s for RU b
CK_s	Computational demand of SF s
RB_s	FSs demand of SF s
$MFSI$	Number of the maximum FS index
Num	A large positive integer

(B) Variables:

E_r	Equals 1 if DC is deployed at PP r
$F_{b,r}$	Equals 1 if RU b is directly connected to PP r
D_r	Equals 1 if PP r is used
$H_{b,r}$	Equals 1 if RU b is processed in PP r
$Q_{s,r}$	Equals 1 if SF s is processed in PP r
$U_{s,r}$	Number of reuse of FU s in PP r
$O_{b,s,r}$	Equals 1 if SF s of RU b is processed in PP r
$Z_{b,s,r}$	Equals 1 if the last SF s of RU b to be processed in PP r
$\Omega_{i,j,f}$	Equals 1 if FS f of link $e(i,j)$ is used
$X_{b,s,i,j}$	Equals 1 if RU b is carried on the link $e(i,j)$ with the previous SF s being processed
$\Psi_{b,s,i,j,f}$	Equals 1 if RU b is carried on the FS f on the link $e(i,j)$ with the previous SF s being processed
$L_{s,r,b}$	Equals 1 if $U_{s,r} = b$, where $b \in B$

(C) Minimum Objective:

$$\alpha \cdot \sum_{r} D_r + \beta \cdot MFSI + \gamma \cdot \left[\sum_{b,s,i,j} X_{b,s,i,j} \cdot T_{i,j} + \sum_{b,r} H_{b,r} \cdot T_{oeo} + \sum_{b,s<|S|,r} Z_{b,s,r} \cdot 2 \cdot TE_s + \sum_{b,s<|S|} T_{b,s} + \sum_{s<|S|,r} Q_{s,r} \cdot T_v \right] \quad (2)$$

The MILP objective is to simultaneously minimize the computing, bandwidth resources and the end-to-end latency. The first part is to minimize the number of used PPs. The second part is to minimize the $MFSI$ and the last part is to minimize the end-to-end latency. We set $\alpha = 1/|R|$, $\beta = 1/|F|$, $\gamma = 1/(\sum T_b)/|R|$ [2]. The weights α, β, γ are used to control the contribution of each resource to the objective function.

(D) Constraints:

- **Routing constrains:**

$$\sum_{i \neq r, s<|S|} X_{b,s,i,r} - \sum_{r \neq j, s<|S|} X_{b,s,r,j} = \begin{cases} -1, & \text{if } F_{b,r} = 1 \\ 1, & \text{if } E_r = 1 \\ 0, & \text{others} \end{cases}, \ \forall b, r \quad (3)$$

$$\sum_{s<|S|} X_{b,s,i,j} + \sum_{s<|S|} X_{b,s,j,i} \leq 1, \ \forall i, j (i \neq j), b \quad (4)$$

Eq. (3) ensures that a link should be established from the RU to DC and avoid loop formation with Eq. (4).

- **Capacity constrains:**

$$\sum_{b,s<|S|} X_{b,s,i,j} \cdot RB_s \leq |F|, \ \forall i, j (i \neq j) \quad (5)$$

$$f \cdot \Omega_{i,j,f} \leq MFSI \leq |F|, \ \forall i, j (i \neq j), f \quad (6)$$

$$\sum_{1<s<|S|} Q_{s,r} \cdot CK_s \leq C_{max}, \ \forall r \quad (7)$$

Eq. (5) and (6) ensure that data carried on link $e(i,j)$ and $MFSI$ cannot exceed the number of FSs in each link, while each PP computational capacity is ensured in Eq. (7).

- **Latency constrains:**

$$\left[\sum_{s,i,j} X_{b,s,i,j} \cdot T_{i,j} + \sum_{r} H_{b,r} \cdot T_{oeo} + \sum_{s<|S|,r} Z_{b,s,r} \cdot 2 \cdot TE_s + \sum_{s<|S|} T_{b,s} + \sum_{s<|S|,r} Q_{s,r} \cdot T_v + TC_b \right] \leq T_b, \ \forall b \quad (8)$$

$$T_{s,r} = 1/(CK_s - U_{s,r} \cdot C_{reach}), \ \forall s, r \quad (9)$$

Eq. (8) ensures that the end-to-end latency requirement from RU to DC is satisfied. Eq. (9) calculates the processing latency of each FU s in PP r based on the reuse time considering the M/M/1 model due to the instance competition [4]. Since Eq. (9) is a nonlinear constraint, we could linearize it by introducing an auxiliary variable $G_{s,r} = T_{s,r} \times U_{s,r}$, $\forall s, r$.

$$Num \cdot (Q_{s,r} - 1) \leq U_{s,r} - \sum_{b} b \cdot L_{s,r,b} \leq -Num \cdot (Q_{s,r} - 1), \ \forall s, r \quad (10)$$

$$T_{s,r} = \sum_{b} 1/b \cdot G_{s,r} \cdot L_{s,r,b}, \ \forall r, s \quad (11)$$

$$1 + Num \cdot (Q_{s,r} - 1) \leq \sum_{b} L_{s,r,b} \leq 1 - Num \cdot (Q_{s,r} - 1), \ \forall s, r \quad (12)$$

$$Num \cdot (Q_{s,r} - 1) \leq T_{s,r} \cdot C_s - G_{s,r} \cdot C_{reach} - 1 \leq -Num \cdot (Q_{s,r} - 1), \ \forall s, r \quad (13)$$

Eq. (10)-(13) list the constraints for the variable $G_{s,r}$. Note that Eq. (11) is also a nonlinear constraint, which is further linearized by introducing an auxiliary variable $P_{s,r,b} = G_{s,r} \times L_{s,r,b}$, $\forall s, r, b$.

$$G_{s,r} + Num \cdot (L_{s,r,b} - 1) \leq P_{s,r,b} \leq G_{s,r}, \ \forall s, r, b \quad (14)$$

$$Num \cdot (Q_{s,r} - 1) \leq T_{s,r} - \sum_{b} 1/b \cdot P_{s,r,b} \leq -Num \cdot (Q_{s,r} - 1), \ \forall s, r \quad (15)$$

Eq. (14)-(15) list the constraints for the variable $P_{s,r,b}$.

$$T_{b,s} = \sum_{r} O_{b,s,r} \cdot T_{s,r}, \ \forall b, 1 < s < |S| \quad (16)$$

Eq. (16) calculates the processing latency of each FU s in request RU b. It is worth note that Eq. (16) is also a nonlinear constraint. Next, we linearize it by introducing an

auxiliary variable $a_{b,s,r} = O_{b,s,r} \times T_{s,r}$.

$$0 \le a_{b,s,r} \le T_{s,r},\ \forall b, 1 < s < |S|, r \tag{17}$$

$$T_{s,r} + Num \cdot (O_{b,s,r} - 1) \le a_{b,s,r} \le Num \cdot O_{b,s,r}, \forall b, 1 < s < |S|, r \tag{18}$$

Eq. (17)-(18) list the constraints for variable $a_{b,s,r}$. Then, we convert Eq. (16) into Eq. (19).

$$T_{b,s} = \sum_{r} a_{b,s,r},\ \forall b, 1 < s < |S| \tag{19}$$

- **FS allocation constrains:**

$$\sum_{f} \psi_{b,s,i,j,f} = RB_s \cdot X_{b,s,i,j},\ \forall b, s, i, j (i \ne j) \tag{20}$$

$$\sum_{f \le f' \le \min(f + RB_s, |F|)} \psi_{b,s,i,j,f} \ge RB_s + Num \cdot (\psi_{b,s,i,j,f} - \psi_{b,s,i,j,f-1} - 1),\ \forall b, s, i, j (i \ne j), f \tag{21}$$

$$\sum_{b,s<|S|} \psi_{b,s,i,j,f} \le \Omega_{i,j,f} \le Num \cdot \sum_{b,s<|S|} \psi_{b,s,i,j,f},\ \forall i, j (i \ne j), f \tag{22}$$

Eq. (20) ensures that a RU is allocated FSs equal to the number of FS requested. Eq. (21) ensures that a set of continuous FS' are selected for RU b after placing SF s. Eq. (22) ensures that the spectrum non-overlapping limitation.

- **FU placement constrains:**

$$O_{b,s,r} = \begin{cases} F_{b,r}, s = 1 \\ E_r, s = |S| \end{cases},\ \forall b, r \tag{23}$$

$$\sum_{r} O_{b,s,r} = 1,\ \forall b, s \tag{24}$$

$$\sum_{j, s \le s' < |S|} X_{b,s',r,j} = O_{b,s,r},\ \text{if } F_{b,r} = 1,\ \forall b, s, r \tag{25}$$

$$2 \cdot O_{b,s,r} \le \sum_{i, 1 \le l_1 < s} X_{b,l_1,i,r} + \sum_{j, s \le l_2 < |S|} X_{b,l_2,r,j} \le O_{b,s,r} + 1,\ \text{if } F_{b,r} = 0,\ \forall b, s, r \tag{26}$$

$$2 \cdot Z_{b,s,r} \le O_{b,s,r} - O_{b,s+1,r} + 1 \le Z_{b,s,r} + 1,\ \forall b, 1 < s < |S| - 1, r \ne DC \tag{27}$$

Eq. (23) ensures that the source and destination nodes of the slice request are its RU and the DC, respectively. Eq. (24) ensures that each SF is placed once in EON. Eq. (25) and (26) ensure that the SF s of RU b can be processed in its direct adjacent PP or other PPs. Eq. (27) is relevant to the interface encapsulation.

- **Reuse constrains:**

$$U_{s,r} = \sum_{b} O_{b,s,r}, \forall 1 < s < |S|, r \tag{28}$$

$$Q_{s,r} \le U_{s,r} \le Num \cdot Q_{s,r}, \forall 1 < s < |S|, r \tag{29}$$

Eq. (9)-(10) count the value of the reuse times of each SF s on each PP r.

- **PP node constrains:**

$$D_r \le \sum_{b, 1 < s < |S|} O_{b,s,r} \le Num \cdot D_r,\ \forall r \tag{30}$$

$$-H_{b,r} \le \sum_{i \ne r} \psi_{b,s,i,r,f} - \sum_{r \ne j} \psi_{b,s,r,j,f} \le H_{b,r},\ \forall b, r, s < |S|, f \tag{31}$$

Eq. (30) ensures that if PP r is used. Eq. (31) ensures that bypassing data shouldn't change its FS f of link $e(i,j)$.

Algorithm 1: HA-DRL policy

1. **For** each episode **do**
2. Initialize state S_t;
3. **For** each 5G FU-based RAN request RU-DC **do**
4. Find a path p^{LBP} according to **Algorithm 2**;
5. Decompose request RU-DC into 5 elements;
6. Obtain current state S_t;
7. **For** each element **do**
8. Generate a random decimal $c \in [0,1]$;
9. **If** $c > \varepsilon$, **then**
10. Select a random action a_t from the path p^{LBP};
11. **Else**
12. Select $a_t = \text{argmax}_a Q(S_t, a)$, where the a_t from the path p^{LBP};
13. **End if**
14. **If** the PP a_t available resource is not enough, **then**
15. **Node mapping fails;**
16. **Else**
17. **Node mapping succeeds**;
18. **End if**
19. **If** node mapping succeeds, **then**
20. Execute **Algorithm 3** according to a_t**;**
21. **If** link mapping succeeds, **then**
22. Execute **Algorithm 4** according to a_t**;**
23. **If** latency check passed, **then**
24. Deploy the FU instance and assign FSs required with **Algorithm 3** and get reward r_t and next state S_{t+1};
25. **Else** Get penalty r_t and next state S_{t+1};
26. **End if**
27. **Else** Get penalty r_t and next state S_{t+1};
28. **End if**
29. **Else** Get penalty r_t and next state S_{t+1};
30. **End if**
31. Record sequence (S_t, a_t, r_t, S_{t+1}) in the memory;
32. **End for**
33. Randomly select mini-batch of sequence (S_t, a_t, r_t, S_{t+1}) from memory Λ;
34. Update DNN with the loss function if the number of data exceeds the threshold.
35. **End for End for**

Algorithm 2: LBP policy

Input: $Req(s, d)$;
Output: p^{LBP};
1. Calculate $P_{s,d}\{p_1, p_2, ..., p_k\}$ according to s and d;
2. **for** $k = 1$ to K **do**
3. Get FUs already deployed on p_k as sequence Φ_k;
4. Calculate LCS degree between Φ_k and FUs [10];
5. **End for**
6. Select p_k with maximum LCS degree;
7. $p^{LBP} = p_k$;

IV. HA-DRL METHODOLOGY

In this section, a HA-DRL algorithm is developed for the FU BFP&R with the FuRe scheme in EONs, where the state, action spaces, and reward function are defined as follows.

1) State Representation: A HA-DRL FU BFP&R with the FuRe scheme is proposed in **Algorithm 1**. In lines 1-2, for each episode, we initialize a state for the slice requests to be deployed. In lines 3-4, receiving a RU-DC request $Req(s, d)$, we first calculate the K shortest paths between s and d as $P_{s,d}\{p_1, p_2, \ldots, p_k\}$ and get the FUs already deployed on path p_k as sequence Φ_k. Then, the path p^{LBP} that has maximum LCS degree [10] between Φ_k and FUs is selected as the p^{LBP}. The process is shown in **Algorithm 2**. In line 5, to facilitate the decision making of the DRL-agent, a request RU-DC request is decomposed into 5 units, i.e., (b_1, FU1), (b_2, FU2), (b_3, FU3), (b_4, FU4), (b_5, FU5, b_6) and handle them in turn. In line 6, the DRL agent obtains the current state for the

Algorithm 3: Link mapping policy

Input: The given last PP i and the selected PP a_t, required FS number b_k;
1. Calculate K shortest paths between PP i and PP a_t;
2. **For** k = 1 to K **do**
3. Calculate F_{Pk}^{max};
4. **End for**
5. **If** $|F| \geq (\min\{F_{P1}^{max}, F_{P2}^{max}, \ldots, F_{Pk}^{max}\} + b_k)$, **then**
6. **Link mapping succeeds** and select path p_k with minimum F_{Pk}^{max};
7. Assign FSs indexed from $F_{pk}^{max}+1$ to $F_{pk}^{max}+b_k$ on all links in path p_k;
8. **Else**
9. **Link mapping fails.**
10. **End if**

Algorithm 4: Latency check policy

1. **For** all the requests RU $b \in B$ that are already deployed **do**
2. Calculate the end-to-end latency D_b^{map} of requests RU b;
3. **If** $D_b^{map} < T_b$, **then**
4. **Delay check passed;**
5. **Else**
6. **Delay check failed;**
7. **End if End for**

current FU of the current slice request in the *t-th* time step, which is comprised of the FU request information and the current FU-based RAN information. It can be defined as

$$S_t = \{PP_{used}, C_{avail}, FMSI, T_{total}, A_s, RU\} \tag{32}$$

where PP_{used} denotes which PP nodes have been used for FU placement, C_{avail} is the available computational resource in each PP, *FMSI* is the current *MFSI*, T_{total} is the accumulated latency at present for deploying the previous FU, and A_s is a set of actions for deploying the previous FUs, and *RU* is the starting point of the slice request.

2) Action Definition: In lines 7-13, the agent selects an action with ε-greedy strategy. We utilize LBP policy to reduce the size of the current action space. In lines 14-18, if PP a_t has enough resources to deploy current FU, the node mapping of action a_t succeeds. In lines 19-20, we apply a link mapping policy to deploy the FSs demand as shown in **Algorithm 3**. For each link $l \in L$, the index of the last employed FS on the link l is defined as F_l^{max}. Similarly, the index of the last employed FS on the path p_k is defined as F_{pk}^{max}, where $F_{pk}^{max} = max\{F_1^{max}, F_2^{max}, \ldots, F_l^{max}\}$, $l \in p_k$. In this policy, for two PPs (i.e., PP i and PP j ($i,j \in R$)), we first calculate the K shortest paths between them, denoted as $P_{i,j}\{p_1, p_2, \ldots, p_k\}$. Then, if $|F| \geq (\min(F_{P1}^{max}, F_{P2}^{max}, \ldots, F_{Pk}^{max}) + b_k)$, the link mapping succeeds and we select the path p_k which has minimum F_{Pk}^{max}, and b_k required FSs are assigned on fibers in the path p_k. In lines 21-22, after performing the action a_t to the EON, we need to check that all deployed requests in the EON continue meeting their end-to-end latency requirement. In lines 2-7 of **Algorithm 4**, we calculate the end-to-end latency D_b^{map} of each request RU b that has been deployed in the EON and compare it to its end-to-end latency threshold T_b. If the total latency of each request RU b exceeds its end-to-end latency threshold, the latency check of action a_t fails.

3) Reward Description: For each action a_t that satisfies node, link and latency constraints, we will deploy the FU instance and assign FSs required with **Algorithm 3** and set a big reward (lines 23-24), which sums of the newly activated PP X_t, consumed *FMSI* Y_t and end-to-end latency Z_t. On the contrary, we set a reward $r_{\max}$ for an action that cannot fulfill all constraints (line 25-30). Therefore, the reward function is shown in Eq. (33).

$$r_t = \begin{cases} -(\alpha \cdot X_t + \beta \cdot Y_t + \gamma \cdot Z_t), & \text{if } a_t \text{ is valid} \\ -r_{\max}, & \text{if } a_t \text{ is invalid} \end{cases} \tag{33}$$

4) Training Mechanism: In line 31-32, we record data (S_t, a_t, r_t, S_{t+1}) in the memory Λ. To train the HA-DRL model, the deep neural network (DNN) parameters are updated by $loss = E\{[r_t + \mu \times \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]^2\}$. where $r_t + \mu \times \text{argmax}Q(S_{t+1}, a_{t+1})$ is the optimal Q-value, μ denotes discount rate and $Q(S_t, a_t)$ denotes the Q-value before updating. In this work, we implement double deep Q network (DDQN) as our DRL algorithm. In lines 33-34, a mini-batch of sequences (S_t, a_t, r_t, S_{t+1}) is selected from memory Λ to train the DDQN model, then the gradient descent algorithm is used to update the DNN parameters.

V. Performance Evaluation

We perform the simulations to evaluate the performance of our proposed MILP model and HA-DRL algorithm with 9-node network and 30-node topology, respectively [5]. Three baseline benchmarks are considered for comparison, i.e., decentralized LBA (D-LBA), centralized LBA (C-LBA) [10], and shortest-path and random algorithm (SRA). In the simulation, for the wireless part, we consider that each RU includes 32 antennas, 8 MIMO layers, 100 MHz wireless spectrum and MCS equals 23 [2], [8]. For small-scale network, each optical link ranges in [5, 30] *km*, and available FSs is 15. For large-scale network, each optical link ranges in [5, 30] *km*, and available FSs is 250.

Fig. 3 (a) demonstrates the results of the average cost, which is calculated by Eq. (2). We find that the MILP model achieves the lower average cost, followed by the HA-DRL algorithm. The baseline benchmarks, the D-LBA, C-LBA and SRA have higher average costs. This is because the MILP can obtain optimal solution by exhaustive search. Meanwhile, the HA-DRL utilizes LBP policy as performance guarantees, resulting in the HA-DRL solution being the closest to the optimal MILP solution.

To further validate the effectiveness of our proposed HA-DRL algorithm, the simulation results under the 30-nodes network topology are shown in Fig. 3 in terms of average cost (Fig. 3(b)), number of used PPs (Fig. 3(c)), *FMSI* (Fig. 3(d)) and total latency (Fig. 3(e)). As shown in Fig. 3(b), all three algorithms consume more average costs as the number of request increases, because more requests would consume more resources. In Fig. 3(c), with the increasing number of requests, all three algorithms consume more PPs, where PPs consumption is the lowest with the proposed HA-DRL. The SRA consumes the most PP resources, because it lacks the ability to proactively FuRe the already deployed FUs.

Fig. 3(d) gives out results on the *MFSI*. Thanks to the DRL training, the proposed HA-DRL algorithm achieves similar *FMSI* performance as D-LBA and C-LBA algorithms. Note that SRA outperforms other algorithms in terms of *FMSI* since it always chooses the shortest path. However, the overall average cost of SRA remains the worst performance because of too much wastage of PP resources. Furthermore, the proposed HA-DRL achieves a better balance between PP consumption and *FMSI* by the aid of the effective DRL training. We can observe in Fig. 3(e) that total latency increases with the number of RUs because of the more transmitted data traffic. We can also observe that SRA needs

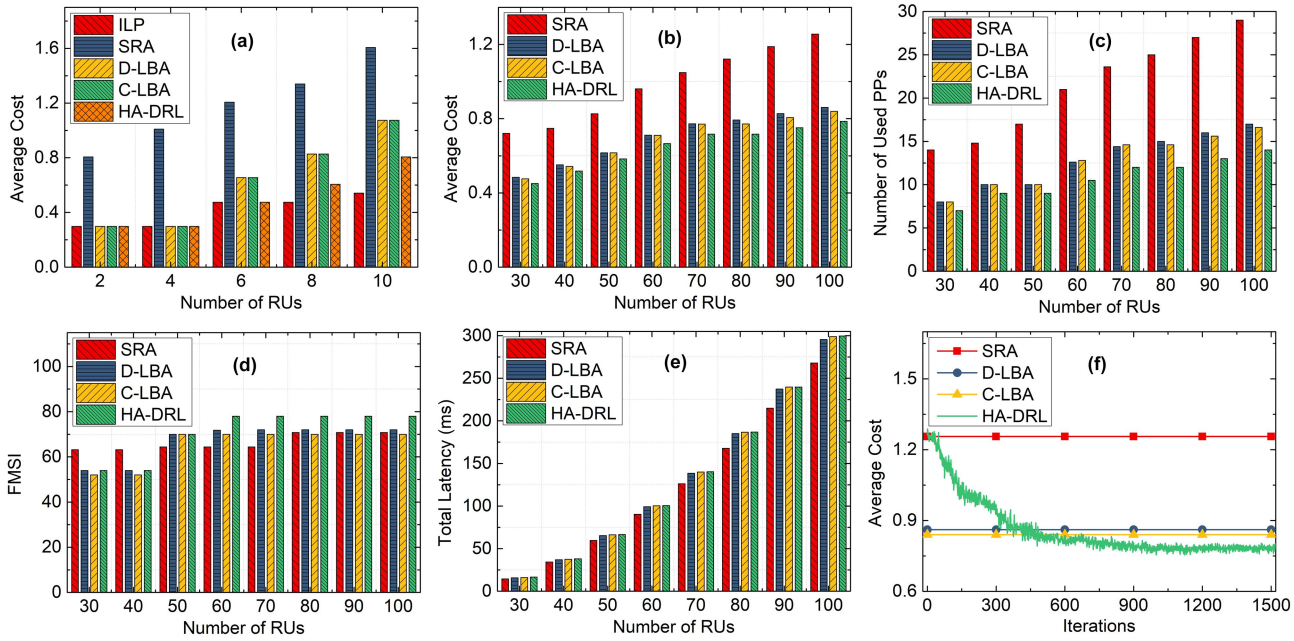

Fig. 3. (a) Average cost in small-scale network; (b) Average cost in large-scale network; (c) Number of used PPs; (d) FMSI; (e) Total latency; (f) Training results of DRL.

the minimum latency followed by D-LBA, C-LBA and HA-DRL. That is because D-LBA and C-LBA and HA-DRL preferentially reuse the already deployed FU, resulting in higher processing latency.

Fig. 3(f) shows the trend of average cost against training iterations for HA-DRL-based algorithm. The training begins with high average cost because of the random exploration, then decreases quickly for improving algorithm, and finally converges to a cost average cost at 0.78. This means that good convergence performance of HA-DRL algorithm is achieved. It is observed that by iterations 500, the average cost obtained by HA-DRL is lower than the other benchmark heuristics and the training curves become flatten and converged after the iteration number is 900.

VI. Conclusion

This paper investigated the resource-efficient FU-BFP&R of 5G RAN slice problem with the FuRe scheme in the EONs. We first introduced the 5G FU-based RAN architecture and formulated the FU-BFP&R issue with the FuRe scheme. Then, a HA-DRL algorithm was proposed to obtain a near-optimal solution. Particularly, the LBP policy was utilized to limit the optional space of DRL agent exploration, which accelerated the training process. In addition, the MILP model was formulation to search for the optimal solution, meanwhile three existing heuristic algorithms were also employed as the benchmarks. Finally, the simulation results validated our proposed MILP model and HA-DRL algorithm can achieve higher resource efficiency and minimize end-to-end latency compared with other benchmarks.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China (62271135，62101121 and 62101126), the Key Research and Development Program of Jiangsu Province (BE2020012), the Major Key Project of Peng Cheng Laboratory (PCL 2021A01-2)， the project funded by China Postdoctoral Science Foundation (2021M702501 and 2022T150486).

References

[1] *Technical Specification Group Radio Access Network; NG-RAN; Architecture Description, V16.0.0, Rel. 16*, 3GPP Standard TS 38.401, Dec. 2019.

[2] Y. Xiao, J. Zhang and Y. Ji, “Can fine-grained functional split benefit to the converged optical-wireless access networks in 5G and beyond?,” *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1774-1787, Sep. 2020.

[3] *Study on new radio access technology: Radio access architecture and interfaces, V14.0.0, rel. 14,* 3GPP Standard TS 38.801, Mar. 2017.

[4] M. Zhu, J. Gu, T. Shen, J. Zhang and P. Gu, “Delay-aware and resource-efficient service function chain mapping in inter-datacenter elastic optical networks.” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 14, no. 10, pp. 757-770, Oct. 2022.

[5] Y. Xiao, J. Zhang and Y. Ji, “Energy-efficient DU-CU deployment and lightpath provisioning for service-oriented 5G metro access/aggregation networks,” *J. Lightw. Technol.*, vol. 39, no. 17, pp. 5347-5361, Sep. 2021.

[6] Y. Xiao, J. Zhang, Z. Gao and Y. Ji, “Service-oriented DU-CU placement using reinforcement learning in 5G/B5G converged wireless-optical networks,” *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020, pp. 1-3.

[7] Z. Gao, S. Yan, J. Zhang, B. Han, Y. Wang, Y. Xiao, D. Simeonidou and Y. Ji, “Deep reinforcement learning-based policy for baseband function placement and routing of RAN in 5G and beyond,” *J. Lightw. Technol.*, vol. 40, no. 2, pp. 470-480, Jan. 2022.

[8] *Technical Specification Group Radio Access Network; NR; Physical Layer Procedures for Data, V16.0.0, Release 16*, 3GPP Standard TS 38.214, Dec. 2019.

[9] Y. Yin *et al.*, “Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks,” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A100-A106, Oct. 2013.

[10] M. Zhu, Q. Chen, J. Gu and P. Gu, “Deep reinforcement learning for provisioning virtualized network function in inter-datacenter elastic optical networks,” *IEEE Trans. Netw. Service Manag.*, doi: 10.1109/TNSM.2022.3172344.