

DOI:10.20079/j.issn.1001-893x.240618004

## 面向近海通信的移动边缘网络动态服务缓存策略\*

庄颖瑜, 潘春雨, 李学华

(北京信息科技大学 现代测控技术教育部重点实验室, 北京 100101)

**摘要:**为了实现低时延低能耗的近海通信,将动态服务缓存更新机制引入复杂神经网络当中,基于近海通信场景对复杂的神经网络结构进行巧妙设计,提出了基于双重深度 Q 网络(Double Deep Q Network, DDQN)的移动边缘动态服务缓存策略(Mobile Edge Dynamic Service Caching Policy, MEDSCP)。该策略先通过用户终端任务卸载决策博弈得到最佳卸载决策集,然后利用移动边缘计算(Mobile Edge Computing, MEC)和动态服务缓存更新来减少近海通信环境下任务执行的时延能耗成本,旨在提高近海通信的任务处理效率并扩展该行业的发展潜力。仿真实验结果表明,所提的 MEDSCP 策略与现有工作相比,能在保证训练效果的前提下实现算法的快速收敛,还能有效降低近海通信的时延能耗加权和。

**关键词:**近海通信;深度强化学习;移动边缘计算;博弈论;动态缓存

开放科学(资源服务)标识码(OSID):



微信扫描二维码  
听独家语音释文  
与作者在线交流  
享本刊专属服务

中图分类号:TN929.5;TP181 文献标志码:A 文章编号:1001-893X(2025)11-1754-12

## Dynamic Service Caching Strategy for Mobile Edge Computing to Aid Offshore Communications

ZHUANG Yingyu, PAN Chunyu, LI Xuehua

(Key Laboratory of Modern Measurement & Control Technology, Ministry of Information Industry,  
Beijing Information Science and Technology University, Beijing 100101, China)

**Abstract:** In order to achieve low-latency and low-energy offshore communication, the dynamic service cache update mechanism is introduced into the complex neural network, and the mobile edge dynamic service caching policy (MEDSCP) based on double deep Q network (DDQN) is proposed by cleverly designing the complex neural network structure based on offshore communication scenarios. The policy firstly obtains the optimal offloading decision set through the user terminal task offloading decision game, and then utilizes mobile edge computing (MEC) and dynamic service caching update to reduce the delay and energy cost of task execution in the offshore communication environment, aiming to improve the efficiency of task processing in offshore communication and to expand the development potential of this industry. Simulation experimental results show that the proposed MEDSCP strategy can achieve fast convergence of the algorithm while guaranteeing the training effect, and also effectively reduce the delay-energy weighted sum of offshore communications compared with existing work.

**Key words:** offshore communication; deep reinforcement learning; mobile edge computing; game theory; dynamic cache

\* 收稿日期:2024-06-18;修回日期:2024-11-8

基金项目:北京市自然科学基金-海淀原始创新联合基金(L212026);北京市教委科技计划一般项目(KM202211232011)

通信作者:潘春雨 Email:chunyu@bistu.edu.cn

## 0 引言

随着我国海洋经济的快速发展以及近海活动的不断增多,近海移动数据量呈现爆发式增长<sup>[1]</sup>,加之近海智能终端设备的不断迭代升级,用户对数据传输速率和通信质量提出了更高的要求。且相较于内陆稳定的通信环境,近海无线通信会面临移动平台(如船只和浮动平台)的高移动性带来的挑战,这就要求通信系统必须具备动态适应能力,能够快速调整以应对平台的不断变化<sup>[2-3]</sup>。尤其在近海港口的航运场景下对计算密集型和时延敏感型的海洋应用和服务的需求更为明显<sup>[4]</sup>,当船只在进出港口时,近海用户只能向岸基基站请求并获取信息,请求数量较频繁时会使得岸基基站负载过重,进而引发网络拥塞<sup>[5]</sup>。

为了能够处理庞大的移动数据流量,学术界着眼于移动边缘计算(Mobile Edge Computing, MEC)技术,将网络存储功能下沉到网络边缘,就近为用户提供所需的服务,从而缓解网络回程链路的压力,降低网络传输时延,提升系统吞吐量。近年来,已有大量文献研究 MEC 技术的应用与优化。从用户的角度来看,将计算任务卸载到 MEC 服务器进行处理不仅能够显著降低用户设备的能耗,同时还能够缩短任务执行的时间,提高计算任务的执行效率<sup>[6-7]</sup>,满足近海移动终端设备长续航的需求。

在近海终端执行计算任务时,需要将用户数据输入到相应功能的应用程序数据库中。本文中的近海移动通信系统包含多种应用程序数据,用于处理多种不同的用户数据,这些应用程序被概括为虚拟网络功能(Virtual Network Function, VNF)。现有的关于计算卸载的文献可以分为两种主要情况:一是 MEC 服务器具有无限的存储空间,可以为用户存储所有需要的 VNF<sup>[8-9]</sup>;二是 MEC 服务器的缓存大小有限,无法缓存所有用户所需的 VNF。由于第二种情况更具有实际的应用价值,因此本文考虑第二种情况,让终端按需上传用户数据和匹配的 VNF<sup>[10]</sup>。

目前,关于任务计算结果缓存<sup>[11-12]</sup>或基于任务 VNF 缓存的 MEC<sup>[13-14]</sup>工作大多数集中于静态缓存网络边缘的数据。事实上,近海用户对计算任务的需求随时间动态变化。静态缓存策略无法保持缓存数据的高可重用性。因此,设计基于动态学习的方法来预测用户的任务需求并动态调整缓存内存以提高缓存数据的可重用率非常重要。Chen 等人<sup>[15]</sup>提出了服务动态缓存技术,在 MEC 服务器上动态更新

服务缓存以降低单位时隙内的执行能耗。但是在近海通信场景中,计算任务需要低时延,且移动终端需要长续航。Liu 等人<sup>[16]</sup>和 Gao 等人<sup>[17]</sup>虽然提出了基于任务分区和卸载的联合设计方案,但未能有效结合动态用户请求预测来优化资源分配,这将导致系统可能无法预见并处理高峰时段的数据请求,从而未能最大限度地降低延迟和能耗。Wang 等人<sup>[18]</sup>和 Singh 等人<sup>[19]</sup>提出的模型没有充分考虑回程链路的压力,高密度的数据流量可能导致回程链路的瓶颈问题未能得到有效缓解。Chen 等人<sup>[20]</sup>和 Wu 等人<sup>[21]</sup>虽然提出了缓存优化和卸载决策的方案,但这些方案未能综合考虑动态任务负载和缓存容量的限制,导致在高负载条件下的性能可能不足。

因此,本文提出基于移动边缘计算和深度神经网络训练得到的多用户实时任务卸载和动态服务缓存策略联合优化算法。该算法首先将原问题分解为用户终端任务卸载决策和动态服务缓存更新策略两个子问题,先寻求用户终端任务卸载决策的纳什均衡解,然后利用近海通信任务类型数量庞大且重复出现的特点,重新设计基于马尔可夫模型建立的深度神经网络结构,将同种计算卸载任务类型归类为一个神经元,降低神经网络的训练复杂度,得到最佳的移动边缘动态服务缓存策略。

## 1 系统模型

本文的系统模型需要依托于近海场景进行描述。相比蜂窝通信的信道传输链路,近海无线通信的通信链路受到以下几个关键因素的影响:①多径传播:近海环境中,水面反射和散射造成多路径传播现象,会导致信号到达接收端的时间延迟和相位变化,增加信号干扰;②环境干扰:海洋环境中的风浪、盐雾、潮汐等自然因素会对信号传播产生影响,导致信号衰减和失真;③移动性:近海通信中,船只和浮动平台的移动性较高,需要考虑多速率和多种移动模式对信道的影响。

蜂窝通信的信道表达式为

$$h_{\text{cellular}}(t) = \alpha e^{j\phi} \delta(t - \tau) \quad (1)$$

式中: $\alpha$  表示信道增益; $e^{j\phi}$  表示信道的相位偏移。

相较于式(1)提到的蜂窝通信,近海通信的信道表达式考虑了更多的影响因素:

$$h_{\text{offshore}}(t) = \sum_{i=1}^N \alpha_i e^{j\phi_i} \delta(t - \tau_i) \quad (2)$$

式中: $N$  表示路径总量。

## 1.1 网络模型

本文研究基于 MEC 技术的多终端近海通信系统,如图 1 所示。系统配置了一个配有移动边缘计算服务器的基站和若干个进行海上通信的用户,且该服务器能够从云端的软件库获取执行部分特定任务类型的 VNF。 $K$  个用户设备随机分布其中,用  $\mathcal{K} = \{1, 2, \dots, K\}$  表示。假设用户能够生成的计算任务有  $W$  种,用  $\mathcal{W} = \{1, 2, \dots, W\}$  表示。假设系统的每个时隙长度都为  $\tau$ ,时隙的序列表示为  $\mathcal{T} = \{1, 2, \dots, T\}$ 。在每个时隙开始时,每个终端设备都会决定是否要生成一个计算任务,这些计算任务属于任务类别库  $\mathcal{W}$  中的某个任务类别。本文假设无论采用何种任务执行方式,每个计算任务都必须在对应的时隙结束之前被执行完成。为了满足这个要求,可以通过调整每个时隙的长度(即  $\tau$  的大小)来确保任务能够在规定的时限内完成。另外,每个时隙结束时,MEC 服务器会更新服务缓存空间,从云端的软件库主动缓存相应 VNF,有效的缓存策略有助于提高任务的执行效率和系统性能。

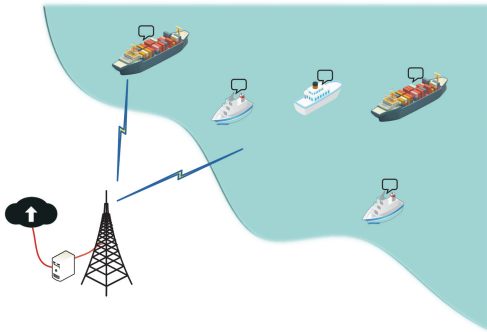


图 1 近海通信系统模型

每个计算任务  $w \in \mathcal{W}$  被定义为  $(D_w, V_w, U_w)$ , 其中,  $D_w$  代表任务类型  $w$  的计算任务数据大小,  $V_w$  代表任务类型  $w$  的 VNF 数据大小,  $U_w$  代表执行任务  $w$  所需要的 CPU 周期。 $b_w^{(t)} \in \{0, 1\}$  代表任务类型  $w$  的 VNF 在时隙  $t$  的服务器缓存状态,  $b_w^{(t)} = 1$  代表 MEC 服务器已在此时隙内缓存执行该任务类型的 VNF;  $b_w^{(t)} = 0$  代表 MEC 服务器在此时隙内没有缓存执行该任务类型的 VNF。MEC 服务器的缓存容量用  $C$  来表示。在任意时隙,服务器缓存的数据大小都不应超过 MEC 的缓存容量  $C$ :

$$\sum_{w \in \mathcal{W}} b_w^{(t)} V_w \leq C, \forall t \in \mathcal{T} \quad (3)$$

在时隙  $t$  的动态服务缓存策略可以用  $B_t = \{\beta_1^{(t)}, \beta_2^{(t)}, \dots, \beta_W^{(t)}\}$  表示。令  $\beta_w^{(t)} \in \{-1, 0, 1\}$  代表  $t$

时隙 MEC 对任务类型  $w$  的 VNF 缓存更新动作,  $\beta_w^{(t)} = 1$  表示执行任务类型  $w$  的 VNF 将在时隙  $(t+1)$  被添加到 MEC 服务器的缓存中;  $\beta_w^{(t)} = 0$  表示在时隙  $t+1$ , MEC 服务器缓存中执行任务类型  $w$  的 VNF 缓存状态不变;  $\beta_w^{(t)} = -1$  表示在时隙  $t+1$  时, 将执行任务类型  $w$  的 VNF 从 MEC 服务器中删除。因此,任务类型  $w$  在时隙  $t+1$  的缓存状态表示为  $b_w^{(t+1)} = b_w^{(t)} + \beta_w^{(t)}$ , 其中 MEC 服务器不能移除未缓存过的 VNF, 所以应满足  $\beta_w^{(t)} > -b_w^{(t)}$ 。

将  $t$  时隙的近海用户终端请求用  $U_t = \{\mu_1^{(t)}, \mu_2^{(t)}, \dots, \mu_K^{(t)}\}$  表示, 令  $\mu_k^{(t)} \in \overline{\mathcal{W}} (\overline{\mathcal{W}} = \{0\} \cup \mathcal{W})$  表示用户  $k$  在时隙  $t$  的计算任务请求状态,  $\mu_k^{(t)} = 0$  表示用户  $k$  在时隙  $t$  不生成计算任务请求,  $\mu_k^{(t)} = w (w \in \mathcal{W})$  表示用户  $k$  在时隙  $t$  生成执行任务类型  $w$  的请求。假设用户在时隙  $(t+1)$  的任务请求状态只受到该用户在时隙  $t$  的任务请求状态影响。

## 1.2 通信模型

本文假设用户与基站之间的近海通信采用无线传输方式进行, 设该场景下通信系统内的可用带宽为  $B$  Hz, 并平均分配给  $M$  个信道使用, 每个信道的带宽为  $B/M$ 。信道序列用  $\mathcal{M} = \{1, 2, \dots, M\}$  表示, 假设每个用户在每个时隙内仅能使用一条信道和基站通信。用  $\alpha_{k,t}$  表示在时隙  $t$  时刻, 用户  $k$  使用的信道序号。用  $\mathcal{L}_t = \{\alpha_{1,t}, \alpha_{2,t}, \dots, \alpha_{K,t}\}$  表示时隙  $t$  的近海用户任务计算决策集。用  $\mathcal{J} = \{1, 2, \dots, K-1\}$  表示不包含用户  $k$  的用户序列集。分别用  $h_k$  和  $p_k$  表示用户  $k$  进行近海通信的信道增益和传输功率。因此, 用户  $k$  在时隙  $t$  进行近海通信的最低上行链路速率可以表示为

$$r_{k,t} = \frac{B}{M} \text{lb} \left( 1 + \frac{p_k h_k}{\sum_{n \in \mathcal{J}, \alpha_{n,t} = \alpha_{k,t}} p_n h_n + \sigma^2} \right) \quad (4)$$

式中:  $\sigma^2$  表示无线通信信道中存在的加性高斯白噪声的方差。当多个用户同时通过同一个信道上传数据时会受到严重的噪声干扰, 这将增加用户进行近海通信的能耗和任务执行时延。因此, 在面对这种情况时, 部分用户的近海通信将考虑在本地执行计算任务。

近海无线通信的信道模型增益设置为

$$h_{k,t} = g_k(t) d_k^{-n} \quad (5)$$

式中:  $d_k$  是终端  $k$  和基站之间的距离;  $n$  是路径损耗因子;  $g_k(t)$  服从参数为 1 的指数分布, 表示  $t$  时隙终端  $k$  到 MEC 服务器的小尺度衰落信道功率增益。

对于近海终端  $k$  在时隙  $t$  的任务请求  $\mu_k^{(t)}$  到时

隙  $t+1$  的任务请求  $\mu_k^{(t+1)}$  的转移概率如公式 (6) 所示:

$$P[\mu_k^{(t+1)}=j|\mu_k^{(t)}=i]=\begin{cases} R, i \in \overline{\mathcal{W}}, j=0 \\ (1-R) \frac{1/j^\delta}{\sum_{m=1}^n 1/m^\delta}, i=0, j \in \mathcal{W} \\ \frac{1-R}{N}, \text{其他情况} \end{cases} \quad (6)$$

式中:  $R$  表示当前时隙存在任意任务请求,而下一时隙没有任何请求的概率;  $i=0, j \in \mathcal{W}$  时,状态转移概率服从 Zipf 分布(长尾分布),该分布由参数  $\delta$  确定;每个计算任务与  $N$  个人为设定的计算任务相关联。

如图 2 所示,环境参数  $\delta$  对于终端  $k$  在时隙  $t$  的近海通信任务请求  $\mu_k^{(t)}$  到时隙  $t+1$  的任务请求  $\mu_k^{(t+1)}$  转移概率服从 Zipf 分布,该分布由环境参数  $\delta$  控制 Zipf 分布的不均匀程度; $\delta$  越大,分布倾斜程度越大,即某些任务类型出现的概率越大;相反, $\delta$  越小,所有终端选择不同任务类型的概率越趋近等概率。

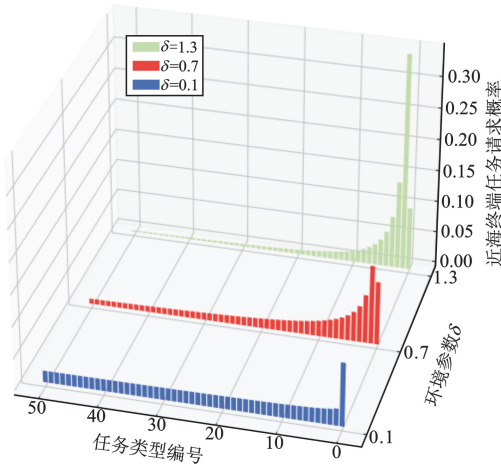


图 2 环境参数  $\delta$  对计算任务分布的影响

### 1.3 计算模型

在网络模型中,MEC 服务器主动从云端的软件库缓存选择的 VNF,以降低下一时隙系统执行任务所需的时延能耗成本。每个时隙开始时,近海用户首先向 MEC 服务器发送任务请求类型,服务器返回是否已缓存执行该任务类型的 VNF。如果配置了对应类型的 VNF,近海用户终端会选择将计算任务卸载到 MEC 服务器上执行计算卸载(有缓存的计算卸载模式)。如果没有配置该类型的 VNF,用户终端将根据动态服务缓存更新策略决定是通过本地执行计算任务(本地计算模式),还是将任务数据和

VNF 同时卸载到 MEC 服务器上进行计算(无缓存的计算卸载模式)。由于请求信息的数据量明显小于计算任务的数据量,因此本模型忽略了终端向 MEC 服务器发送任务请求和服务器返回是否已缓存相应的 VNF 两个过程所产生的时延能耗。

本文的时延表示信息从发送端到接收端所需的时间,能耗是指通信过程中消耗的电能量。在信息传输的过程中,降低传输时延将带来能耗的提升,而降低能耗将导致时延的增加。因此,本文综合考虑时延和能耗两个指标,探究时延和能耗的均衡。“时延能耗加权和”是一个综合考量时延和能耗之间权衡关系的物理量,其物理意义可以解释为:时延能耗加权和表示为时延与能耗分别与对应权重系数乘积的加权和,反映在通信系统中,为了在时延和能耗之间达到一个平衡,可以通过施加权重系数进行必要的折中考虑。

#### 1.3.1 本地计算模式

如果用户终端通过动态服务缓存更新策略决定在本地执行计算任务,则本地计算模式的时延可表示为

$$T_{k,w}^L = \frac{U_w}{f_k^L} \quad (7)$$

式中:  $f_k^L$  表示用户设备  $k(k \in \mathcal{K})$  的 CPU 每秒时钟周期数(即表征 CPU 的计算能力);  $U_w$  表示执行任务  $w$  所需要的 CPU 周期数。由于每个任务都需在其生成的时隙  $\tau$  内完成,所以  $f_k^L$  应满足  $f_k^L \geq U_w/\tau$ 。

本文假定近海用户通信设备的工作负载相同,能耗与其执行计算任务的时间成正比,因此本地计算卸载模式的能耗为

$$E_{k,w}^L = p_k T_{k,w}^L \quad (8)$$

式中:  $p_k$  表示用户  $k$  进行近海通信的传输功率。

本文构建的近海通信模型的能耗和时延权重可根据通信场景的不同进行配置,其中,  $\varphi$  是时延的权重因子,  $\theta$  是能耗的权重因子,则本地计算卸载模式的整体成本可以记作

$$Z_{k,w}^L = \varphi T_{k,w}^L + \theta E_{k,w}^L \quad (9)$$

#### 1.3.2 有缓存的计算卸载模式

如果用户设备  $k(k \in \mathcal{K})$  选择通过卸载来执行任务  $w$ ,并且该计算任务  $w$  的 VNF 已经被 MEC 服务器缓存,则近海用户终端  $k$  仅需将计算任务的数据卸载到 MEC 服务器,由 MEC 服务器执行计算任务,并将结果返回给近海用户终端。该模式的时延可以表示为

$$T_{k,w,t}^O = \frac{U_w}{f_c} + \frac{D_w}{r_{k,t}} \quad (10)$$

式中:  $f_c$  表示 MEC 服务器的计算能力;最低上行链路速率  $r_{k,t}$  由式(4)得到,且时延需满足  $T_{k,w,t}^0 < \tau$ 。有缓存的计算卸载模式的能耗为

$$E_{k,w,t}^0 = p_k \frac{D_w}{r_{k,t}} \quad (11)$$

有缓存的计算卸载模式的整体成本可以记作

$$Z_{k,w,t}^0 = \varphi T_{k,w,t}^0 + \theta E_{k,w,t}^0 \quad (12)$$

### 1.3.3 无缓存的计算卸载模式

如果用户设备  $k (k \in \mathcal{K})$  选择通过计算卸载来执行任务  $w$ ,但是该计算任务  $w$  的 VNF 没有被 MEC 服务器缓存,则用户终端  $k$  需将计算任务数据和相应 VNF 同时卸载到 MEC 服务器,由服务器执行计

算任务,并将结果返回到近海用户终端。该模式的时延可以表示为

$$T_{k,w,t}^N = \frac{U_w}{f_c} + \frac{D_w + V_w}{r_{k,t}} \quad (13)$$

和有缓存的计算卸载模式相同,也需要满足  $T_{k,w,t}^N < \tau$ 。该模式的能耗为

$$E_{k,w,t}^N = p_k \frac{D_w + V_w}{r_{k,t}} \quad (14)$$

无缓存的计算卸载模式的整体成本可以记作

$$Z_{k,w,t}^N = \varphi T_{k,w,t}^N + \theta E_{k,w,t}^N \quad (15)$$

3 种计算模型的时延能耗成本对比如表 1 所示。

表 1 计算模型时延能耗对比

计算模式	适用情况	时延 $T$	能耗 $E$	成本 $Z$
本地计算 Local	适用于用户设备通过无线传输干扰严重的情形	$\frac{U_w}{f_k}$	$p_k T_{k,w}^L$	$\varphi T_{k,w}^L + \theta E_{k,w}^L$
有缓存的计算卸载 Offloading	适用于用户设备选择通过计算卸载来执行任务,且 MEC 服务器缓存有该任务类型的 VNF 的情形	$\frac{U_w}{f_c} + \frac{D_w}{r_{k,t}}$	$p_k \frac{D_w}{r_{k,t}}$	$\varphi T_{k,w,t}^0 + \theta E_{k,w,t}^0$
无缓存的计算卸载 No-Offloading	适用于用户设备选择通过计算卸载来执行任务,但 MEC 服务器没有缓存该任务类型的 VNF 的情形	$\frac{U_w}{f_c} + \frac{D_w + V_w}{r_{k,t}}$	$p_k \frac{D_w + V_w}{r_{k,t}}$	$\varphi T_{k,w,t}^N + \theta E_{k,w,t}^N$

## 2 问题描述

为了应对近海通信场景中终端的低能耗需求和计算任务卸载的低时延需求,本文提出了一种联合优化用户终端卸载决策和 MEC 服务器动态更新 VNF 缓存的方法。该方法能够有效减少系统内计算任务卸载的总执行时延和终端总能耗的加权平均值。

本文将终端  $k$  在时隙  $t$  的时延能耗总成本表示为

$$Z_{k,t} = \sum_{w \in \mathcal{W}} \Theta(\mu_k^{(t)} = w) \{ \Theta(\alpha_{k,t} = 0) Z_{k,w}^L + \Theta(\alpha_{k,t} \in \mathcal{M}) ((1 - b_w^{(t)}) Z_{k,w,t}^N + b_w^{(t)} Z_{k,w,t}^0) \} \quad (16)$$

式中:  $\Theta(\cdot)$  表示如果括号内的等式成立,则值为 1,否则值为 0。因此本文将问题表述为

$$P: \min_{\mathcal{L}, \beta} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{k \in \mathcal{K}} Z_{k,t} \quad (17)$$

- s. t. C1:  $\sum_{w \in \mathcal{W}} (b_w^{(t)} + \beta_w^{(t)}) D_w \leq C, \forall t \in \mathcal{T}$   
 C2:  $b_w^{(t+1)} = b_w^{(t)} + \beta_w^{(t)}, \forall w \in \mathcal{W}, \forall t \in \mathcal{T}$   
 C3:  $b_w^{(t)} \in \{0, 1\}, \forall w \in \mathcal{W}, \forall t \in \mathcal{T}$   
 C4:  $\beta_w^{(t)} \geq -b_w^{(t)}, \forall w \in \mathcal{W}, \forall t \in \mathcal{T}$   
 C5:  $\alpha_{k,t} \in \{0, 1, \dots, M\}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}$   
 C6:  $\beta_w^{(t)} \in \{-1, 0, 1\}, \forall w \in \mathcal{W}, \forall t \in \mathcal{T}$   
 C7:  $\Theta(\alpha_{k,t} \in \mathcal{M}) ((1 - b_w^{(t)}) Z_{k,w,t}^N + b_w^{(t)} Z_{k,w,t}^0) \leq \tau$   
 C8:  $\Theta(\alpha_{k,t} = 0) Z_{k,w}^L \leq \tau$

在问题 P 中,C1 表示在任意时隙内,MEC 服务

器缓存的 VNF 总数据大小不能超过服务器缓存容量上限;C2 表示执行计算任务  $w$  的 VNF 在时隙  $t+1$  的服务器缓存状态取决于时隙  $t$  的服务更新动作作用于时隙  $t$  的缓存状态;C3 表示 MEC 服务器的缓存状态有且仅有缓存和未缓存两种;C4 表示服务器不能将该时隙未缓存的 VNF 删除;C5 表示计算任务执行方式,其中  $\alpha_{k,t} = 0$  表示终端  $k$  通过本地计算执行任务;C6 表示服务器更新动作包括  $\beta_w^{(t)} = 1$  添加该 VNF,  $\beta_w^{(t)} = 0$  该 VNF 的缓存状态不变,  $\beta_w^{(t)} = -1$  将该 VNF 从缓存中删除;C7 表示采用计算卸载方式需要满足任务的时延标准,即在生成任务的时隙内执行完成;C8 表示采用本地计算方式需要满足任务的时延标准,即在生成任务的时隙内执行完成。

## 3 移动边缘动态服务缓存策略 MEDSCP

由于在该场景下时延能耗总成本问题 P 的复杂性使得无法在多项式时间内找到有效算法来实现最优解。解决问题 P 的困难主要来自近海用户终端计算任务决策和 VNF 在不同时隙的缓存状态,以及缺乏终端在未来的服务请求信息。为了解决这两大困难,本文将原问题分解为两个子问题,即近海用户终端计算任务决策的博弈问题和 MEC 服务器动

态更新 VNF 缓存策略问题。首先,通过一种分散式算法得到近海用户终端任务计算决策博弈问题的纳什均衡解。之后,将 MEC 服务器动态更新 VNF 缓存策略问题重新定义为马尔可夫决策过程,并与近海通信场景进行结合,训练 MEC 服务器得到最优的移动边缘动态服务缓存策略 (Mobile Edge Dynamic Service Caching Policy, MEDSCP)。

### 3.1 用户终端任务卸载决策

基于问题 P 可知对于任意时隙  $t$ ,近海用户终端的计算决策仅影响当前时隙内的时延能耗加权和而不影响其他时隙,并且计算决策与任意时隙的 VNF 缓存决策互不影响,所以本文设计了一个有效的算法来实现近海用户终端任务卸载决策,并且可以推广到解决任意时隙的用户终端任务卸载决策。本文将时隙  $t$  中的任务卸载问题从问题 P 中分解出来:

$$P1: \min_{\mathcal{L}_t} f_t(L_t) \sum_{k \in \mathcal{K}} Z_{k,t} \quad (18)$$

s. t. C2, C3, C7, C8。

首先需要明确问题 P1 是否满足达到纳什均衡的条件,即对于最优的卸载决策  $\alpha_{k,t}^*$ ,在其他用户终端决策  $\mathcal{L}_{-k,t}$  给定的情况下,用户终端  $k$  无法通过改变自己的卸载决策来进一步减少其时延能耗加权和。具体地,如果用户终端  $k$  改变自己的决策为  $\alpha'_{k,t}$  (其中  $\alpha'_{k,t} \neq \alpha_{k,t}^*$ ),则时延能耗加权和的值将不会比最优卸载决策  $\alpha_{k,t}^*$  更小,即

$$f_t(\alpha'_{k,t}, \mathcal{L}_{-k,t}) \geq f_t(\alpha_{k,t}^*, \mathcal{L}_{-k,t}) \quad (19)$$

为了证明 P1 是纳什均衡问题,令  $\mathcal{L}_{-k,t} = \{\alpha_{1,t}, \dots, \alpha_{k-1,t}, \alpha_{k+1,t}, \dots, \alpha_{K,t}\}$  表示除了终端  $k$  之外

的所有其他终端在时隙  $t$  的卸载决策集。让终端  $k$  选择最优的卸载决策  $\alpha_{k,t}^*$ :

$$\alpha_{k,t}^* = \operatorname{argmin}_{\alpha_{k,t}} f_t(\alpha_{k,t}, \mathcal{L}_{-k,t}) \quad (20)$$

经过多次求解,得到最终的最优卸载决策集  $\mathcal{L}_t^* = \{\alpha_{1,t}^*, \alpha_{2,t}^*, \dots, \alpha_{K,t}^*\}$ 。对于每个用户终端  $k$ ,根据以上定义和最优决策的选择条件,若其他用户的决策保持不变,那么

$$f_t(\alpha_{k,t}, \mathcal{L}_{-k,t}) \geq f_t(\alpha_{k,t}^*, \mathcal{L}_{-k,t}) \quad (21)$$

因此,用户终端  $k$  无法通过改变自己的卸载决策来进一步优化目标函数,符合纳什均衡的定义。而近海用户终端任务卸载博弈的目标恰恰是找到一个纳什均衡的卸载策略  $\mathcal{L}_t^* = \{\alpha_{1,t}^*, \alpha_{2,t}^*, \dots, \alpha_{K,t}^*\}$ ,对于计算决策  $\mathcal{L}_t^*$ ,每个用户终端都无法再通过改变它的卸载决策来降低时延能耗加权和。如图 3 所示,用户终端任务卸载决策流程可描述为:首先,将所有用户终端的任务卸载决策初始化为 0。接下来,每个终端根据约束条件 C2、C3、C7 和 C8 计算其可用的任务卸载决策集,并找到其最优的用户终端任务卸载决策  $\alpha_{k,t}^*$ 。然后如果  $\alpha_{k,t}^* \neq \alpha_{k,t}$ ,用户终端  $k$  向 MEC 服务器发送更新卸载状态消息,当 MEC 服务器收到所有终端的更新请求消息时,它会随机选择一个终端并向其发送更新许可消息。收到更新许可消息的用户终端将保持其计算决策不变。最后,如果 MEC 服务器没有收到任何终端的更新请求消息,则向所有用户终端发送结束命令。用户终端收到结束命令后,将不再更新其卸载决策,并根据现有的卸载决策来进行计算任务。

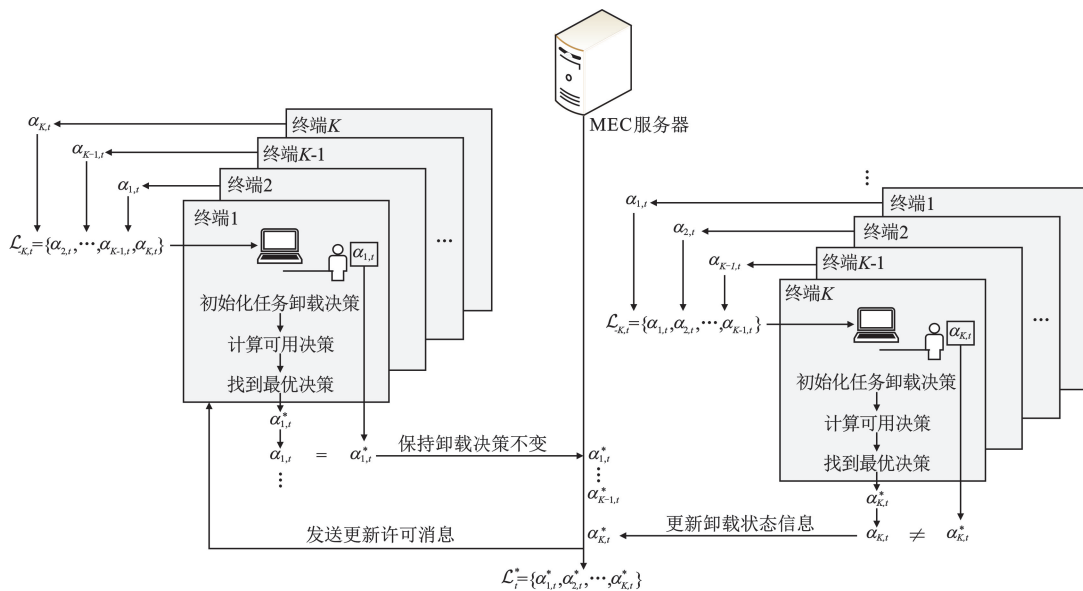


图 3 用户终端任务卸载决策流程

### 3.2 MEC 服务器动态更新 VNF 缓存策略

根据 3.1 节中介绍的用户终端任务卸载决策的博弈算法,可以得到系统任意状态下的终端计算决策解集  $\mathcal{L}_t^*$ 。将  $\mathcal{L}_t^*$  代入问题 P 得到 MEC 服务器动态更新 VNF 缓存策略问题:

$$P2: \min_{\beta_t} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{k \in \mathcal{K}} \hat{Z}_{k,t} \quad (22)$$

s. t. C1, C2, C3, C4, C6。

式(22)中:

$$\hat{Z}_{k,t} = \sum_{w \in \mathcal{W}} \Theta(\mu_k^{(t)} = w) \{ \Theta(\alpha_{k,t}^* = 0) Z_{k,w}^L + \Theta(\alpha_{k,t}^* \in \mathcal{M}) ((1 - b_w^{(t)}) Z_{k,w,t}^N + b_w^{(t)} Z_{k,w,t}^O) \} \quad (23)$$

由于在面对各种各样的任务类型时,  $t$  时隙内很难预测到下一时隙的终端请求状态  $U_{t+1}$ , P2 很难得到解决。根据最低成本广义指派问题 (Lowest cost Generalized Assignment Problem, LGAP) [22] 可以证明, 本文研究的 P2 是 NP-hard 问题。

使用基于双重深度 Q 网络 (Double Deep Q Network, DDQN) 的服务器动态更新 VNF 缓存策略模型来捕捉终端请求计算模型的特征, 并预测更好的 MEC 服务器缓存更新策略。

1) 状态  $S_t$ : 时隙  $t$  的终端请求状态,  $S_t = U_t$ 。

2) 动作  $A_t$ : 服务器为时隙  $t+1$  任务更新缓存状态的动作。

3) 奖励  $R$ : 定义所提算法的平均每时隙时延能耗加权和为奖励  $R$ 。

DDQN 的训练过程包含两个相同的深度神经网络 (Deep Neural Network, DNN) 结构, 其中一个实时更新网络参数的主网络, 另一个是定期复制主网络的网络参数的目标网络。通过两种参数不同的网络评估  $Q$  值, 并选择其中  $Q$  值小的结果作为目标  $Q$  值, 这样可以使目标  $Q$  值的计算更加稳定, 降低了深度 Q 网络 (Deep Q Network, DQN) 中可能出现的过度估计的可能性。

算法公式如下:

$$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)] \quad (24)$$

如图 4 所示, 为了克服大规模任务中异构数据导致的高维和复杂的缓存动作空间, 提高算法收敛效率, 本文修改 DNN 的输入层和输出层神经元连接结构 [22], 然后训练修改后的神经网络得到移动边缘

动态服务缓存策略 MEDSCP。修改 DNN 的步骤是: 根据近海用户终端数量设定第一层 DNN 的神经元数量, 并根据任务类型设定第二层 DNN 的神经元数量。使用  $\mu_k^{(t)} = w$  表示近海用户终端  $k$  在时隙  $t$  的请求的任务类型。传统的做法是将  $U_t$  作为状态信息输入神经网络, 在本文研究的问题中, 为了消除近海用户终端的计算决策顺序对神经网络输出的  $Q$  值的影响, 并提升神经网络的训练速度, 本文采用  $\Theta(\mu_k^{(t)} \in \mathcal{W}), k \in \mathcal{K}$  表示神经网络的状态输入。如果  $w \in \mathcal{W}$  并且  $\mu_k^{(t)} = w$ , 则代表第一层 DNN 的第  $k$  个神经元仅和第二层 DNN 的第  $w$  个神经元相连。如果  $w=0$  并且  $\mu_k^{(t)} = w$ , 则第一层 DNN 的神经元和第二层的所有神经元相连。通过这种方式, 可以降低神经网络的复杂程度, 加快算法的收敛速度。

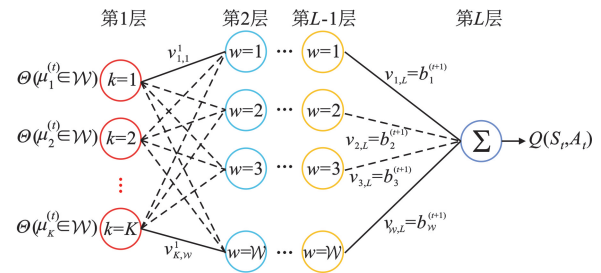


图 4 MEDSCP 神经网络结构

在 DDQN 算法中, 现有研究中 DNN 的输出层神经元数量等于所有可能动作的数量, 即每个神经元对应一个动作, 并输出相应的状态动作值  $Q(S_t, A_t)$ 。然而, 对于问题 P2, 由于不同类型 VNF 的数据大小和任务种类的多样性, 会导致产生大量可能的缓存动作。因此, 为了解决这个问题, 本文修改了 DNN 输出层的结构。设定 DNN 倒数第二层包含的神经元个数等于系统内任务种类个数, 即每个神经元对应一种任务类型。用  $Y = (y_1, y_2, \dots, y_w)$  表示每一个神经元的输出。DNN 的最后一层设定为单一神经元, 后两层神经元之间的连接用  $\mathcal{V} = (v_{1,L}, v_{2,L}, \dots, v_{w,L})$  表示, 将这里面的每个值和预测下一时隙的缓存状态相对应, 即  $v_{w,L} = b_w^{(t+1)}$ ,  $v \in \mathcal{V}$ 。所以, DNN 输出的状态动作价值为

$$Q(S_t, A_t) = \sum_{w \in \mathcal{W}} b_w^{(t+1)} y_w \quad (25)$$

在本文所提算法中, DNN 的训练分为以下几个阶段。在初始训练阶段, 由于缺乏足够的学习经验,

神经网络的参数具有随机性。因此,采用  $\epsilon$ -greedy 策略,增加学习的广度。具体地,先将探索系数  $\epsilon$  的值设定为较大的值,当生成的随机数小于  $\epsilon$  时,采取随机策略,MEC 服务器随机缓存 VNF;当生成的随机数大于  $\epsilon$  时,采取生成最大  $Q$  值的动作。在时隙开始时,终端会生成任务请求  $U_{t+1}$ ,然后根据动态服务更新缓存博弈得到的  $\mathcal{L}_{t+1}^*$  策略完成请求任务,并得到时延能耗加权的成本。根据设定的奖励函数定义,获取当前动作的奖励值,将  $(S_t, A_t, R_t, S_{t+1})$  作为一个数据元组存入经验回放池中。当经验回放池中存储了一定量的数据元组后,算法将采用小批量随机梯度下降方式,从经验回放池中随机抽取一定数量的数据元组并根据公式 (24) 继续进行训练。这种方式有助于减小训练时的内存压力,提高训练效率,并防止模型陷入局部最优解。当训练进行一段时间,即神经网络已经得到部分有效学习后,降低探索系数  $\epsilon$ ,让算法在选择动作时更多地根据学习到的策略执行。

根据上面的步骤进行训练, MEDSCP 策略在满足近海通信任务类型数量庞大且重复出现的条件时能够更好地发挥算法的优势从而降低计算复杂度,降低近海通信所需的时延能耗。分析 MEDSCP 的神经网络结构可知,算法的第二层和倒数第二层神经网络代表的都是近海终端计算任务类型,同种计算任务可归类为一个神经元,当计算任务密集且重复出现时,能够同时减少第二层和倒数第二层 DNN 的神经元数量,从而降低网络复杂度。

假设神经网络层数为  $L$ ,近海用户终端数量设定为第一层 DNN 的神经元数量,假设有  $k$  个神经元;DNN 第二层和倒数第二层(隐藏层)均定义为近海通信的计算任务类型,假设有  $w$  个神经元,DNN 的最后一层修改为单一神经元,而其余未提及的隐藏层(第 3 层到第  $L-2$  层)的神经元个数均为  $n$  个。基于上述假设,考虑满足降低复杂度 ( $w_0$  个任务类型且  $w_0 < w$ ) 的情形给出时间复杂度分析:基于上述的神经网络结构,计算可得前向传播的时间复杂度表示为  $O(k \times w_0 + w_0 \times n + (L-4) \times n^2 + n \times w_0 + w_0)$ ,化简后可得  $O(k \times w_0 + 2w_0 \times n + (L-4) \times n^2)$ 。同理,可得反向传播的时间复杂度。假设训练过程中采用的批次大小为  $batch$ ,训练步数为  $num$ ,则训练的总时间复杂

度为  $O(num \times batch \times (k \times w_0 + 2w_0 \times n + (L-4) \times n^2))$ ,对比常规的 DDQN 算法复杂度  $O(num \times batch \times (k \times w + 2w \times n + (L-4) \times n^2))$ ,当  $w_0 < w$  时,能够有效降低网络复杂度。

表 2 缓存策略对比

任务卸载策略	描述
无缓存的卸载 No-offloading	利用本文提出的基于博弈的多用户实时卸载策略来决定在 MEC 服务器的无 VNF 缓存状态下,终端任务的执行策略
MEDSCP	利用本文提出的移动边缘动态服务缓存策略得到 MEC 服务器的最佳服务缓存策略
有缓存的卸载 MFU-MEC	MEC 服务器根据过去时隙的请求信息计算出最高请求次数的 VNF 进行缓存,当缓存存储已满时,已缓存 VNF 中被使用次数最少的将被新的 VNF 替换
FIFO-MEC	基于先进先出的 MEC 缓存策略,当 MEC 服务器缓存空间不足时,先缓存的 VNF 将优先被替换

#### 4 仿真结果与分析

本节通过仿真实验来评估所提算法的性能。如表 2 所示,将 3 种不同的 MEC 有缓存的卸载策略和无缓存的卸载策略 No-offloading 放到一起对平均每时隙时延能耗加权和进行比较:

1) 基于使用频率更新 MEC (Most Frequently Used Caching-based MEC, MFU-MEC) 的缓存策略<sup>[23]</sup>;MEC 服务器根据过去时隙的请求信息计算出最高请求次数的 VNF 进行缓存。当缓存存储已满时,已缓存 VNF 中被使用次数最少的将被新的 VNF 替换。

2) 基于先进先出的 MEC (First In First Out-based MEC, FIFO-MEC) 缓存策略<sup>[24]</sup>;当 MEC 服务器缓存空间不足时,先缓存的 VNF 将优先被替换。

3) 无缓存卸载策略 No-offloading:利用本文提出的基于纳什均衡博弈的策略来决定在 MEC 服务器的无 VNF 缓存状态下,得到的终端任务的执行策略。

本文首先详细描述了仿真环境,并列出该实验设置的仿真参数表,如表 3 所示。然后,分别从算法层面和策略层面展现了本文所提算法 MEDSCP 的优越性。最后,从服务缓存容量、任务类型数量以及

环境参数三方面出发对平均每时隙时延能耗加权和的影响进行分析。

表 3 仿真参数设置

参数	含义	取值	单位
$K$	近海终端数量	10	台
$W$	任务类型数量	50	种
$P_k$	近海通信设备传输功率	500	mW
$f_k$	近海终端 $k$ 计算能力	1	标准任务块/时隙
$f_c$	MEC 服务器计算能力	20	标准任务块/时隙
$D$	计算任务数据大小	100~500	标准任务块
$V$	VNF 数据大小	1 000~5 000	标准任务块
$\varphi$	时延权重	0.5	无单位
$\theta$	能耗权重	0.5	无单位
$\delta$	环境参数	0.7	无单位

仿真实验采用 Python 实现,假设有  $K$  个近海终端随机分布在  $200\text{ m}\times 200\text{ m}$  的近海场景中,且场景的中心放置一个配置有 MEC 服务器的基站。

在图 5 中,奖励值的设置为平均每时隙时延能耗加权和的值。可以看出,本文所提算法 MEDSCP 在训练到 600 次时,时延能耗加权和可以收敛到 37 附近;DDQN 算法在未修改 DNN 神经元结构的情况下,经过 750 次迭代训练后才开始表现出时延能耗加权的收敛特征,并且收敛到的时延能耗加和在 39 附近,这表明通过采用所提算法 MEDSCP,能够在保证训练效果的前提下实现算法的快速收敛。这是因为 MEDSCP 修改了 DNN 的输入层和输出层神经元连接结构,合理地减少 DNN 神经元间的冗余连接,加快算法收敛速度,这种改进更为贴近近海通信的场景需求,同时也进一步扩展了深度强化学习算法在多维度动作场景中的应用。

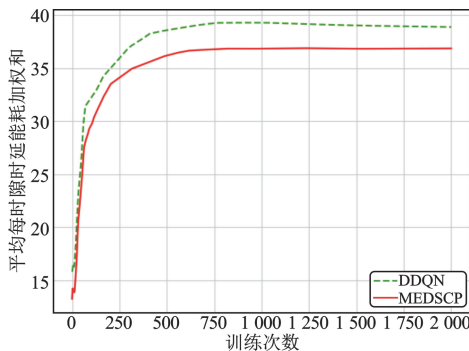


图 5 算法收敛趋势对比

在图 6 中,将所提的 MEDSCP 算法与文献[25]和[26]中所提的优化技术进行了比较。文献[25]提出了一种空中和地面合作的无人机辅助 MEC 网络,采用了基于拉格朗日对偶性的联合优化方法;而文献[26]则提出了一种交替算法,在 MEC 网络中迭代求解两个子问题,实现联合优化。为确保比较的准确性,针对不同的算法采用了相同的网络参数进行训练,然后对系统的平均每时隙时延能耗加权和进行比较。可以看出,所提出的 MEDSCP 算法比文献[25]降低了约 13.51% 的成本,比文献[26]降低了约 8.11% 的成本。这归功于将动态缓存更新机制引入 MEC 网络,并在 DDQN 结构的基础上进行更贴近近海通信场景的修改。对比结果验证了所提方案的优越性。

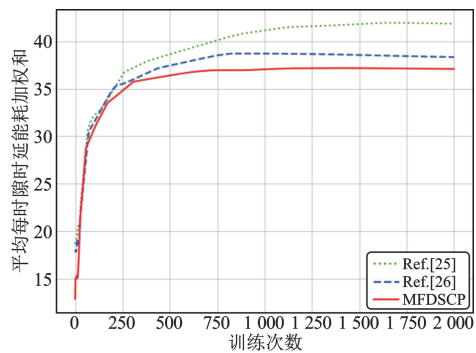


图 6 MEDSCP 与已有文献方案对比

在图 7 中,有缓存的卸载策略包括本文所提的 MEDSCP、根据使用频率更新的 MFU-MEC 以及 FIFO-MEC 3 种。如图所示,在训练初期,3 种有缓存的卸载策略的平均每时隙时延能耗加权和也随之增加。随着训练次数的增加,本文所提的 MEDSCP 策略通过 DNN 的学习能够获得更高的预测准确性,极大地增加有缓存卸载的几率,减少终端传输过程中的数据大小,从而降低系统执行任务的时延能耗加权和,快速收敛在 31 附近。相较于 MFU-MEC 收敛在的 35.5 附近和 FIFO-MEC 收敛在的 37.5 附近,本文提出的 MEDSCP 策略能够在同等情况下更为有效地降低平均每时隙时延能耗加权和,满足近海通信低时延和低能耗的需求。

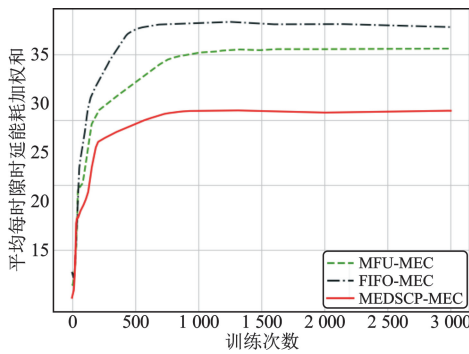


图 7 不同有缓存的卸载策略平均每时隙时延能耗加权和对比如图

在图 8 中,除了无缓存的卸载方案,其余方案平均每时隙时延能耗加权和对比如图都随着缓存大小的增加而降低。这是因为更大的 MEC 服务缓存空间可以存储更多的 VNF,增加了近海终端选择有缓存的计算卸载模式的概率,从而降低了系统的时延和能耗。当缓存大小为 0 时,MEC 服务器无法缓存任何 VNF,所有近海终端只能选择本地计算模式来执行任务。在这种情况下,这些策略之间没有任何区别。当服务缓存容量超过 80 千标准任务块时,由于 MEC 服务缓存容量足以容纳大量 VNF,降低了预测准确度对算法性能的影响,导致所有有缓存的卸载策略的性能相近。而本文所提出的 MEDSCP 策略,能够在相同服务缓存容量的情况下最有效地减少系统的时延能耗加权和对比如图,提高系统性能。

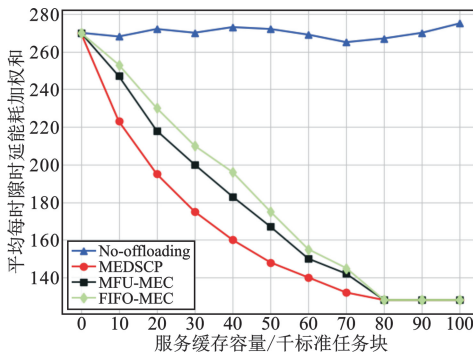


图 8 不同策略下 MEC 服务器缓存容量与平均每时隙时延能耗加权和对比如图的关系

如图 9 所示,假设不同任务执行需要的 VNF 均不相同,其中有缓存的卸载策略的平均每时隙时延能耗加权和对比如图随着任务类型数量的增加而增加。然而,随着任务类型数量的增加,近海终端的任务请求范围会更加广泛,这可能会降低 MEC 服务器缓存策

略的预测准确性,从而降低缓存的 VNF 可重用性,难以降低平均每时隙时延能耗加权和对比如图。此外,根据图片可以观察到,通过更准确地预测近海终端任务需求,本文提出的 MEDSCP 策略在相同任务类型数量的情况下对于平均每时隙时延能耗加权和对比如图的减少明显优于其他策略,更能够满足近海无线通信场景的高需求。

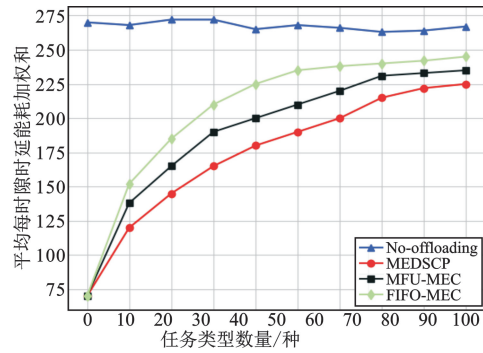


图 9 不同策略下任务类型数量与平均每时隙时延能耗加权和对比如图的关系

图 10 展示了在不同环境参数下,随着近海终端数量变化,平均每时隙时延能耗加权和对比如图的变化情况。不难理解的是,随着近海终端数量的增加,整个通信体系内总的时延能耗加权和对比如图也随之增加。而由于环境参数的增加,本文所提策略的平均每时隙时延能耗加权和对比如图呈现的是逐渐下降的趋势。因为环境参数  $\delta$  的增加会导致大多数近海终端的任务请求会集中在少数任务类型上,提高策略对缓存的任务 VNF 预测的准确性,使缓存任务的 VNF 有更高的使用率,增加近海终端选择有缓存的计算卸载模式的概率,降低系统的平均每时隙时延能耗加权和对比如图。反之, $\delta$  越小,所有终端选择不同任务类型的概率越趋近等概率,降低了缓存方案的预测准确性。

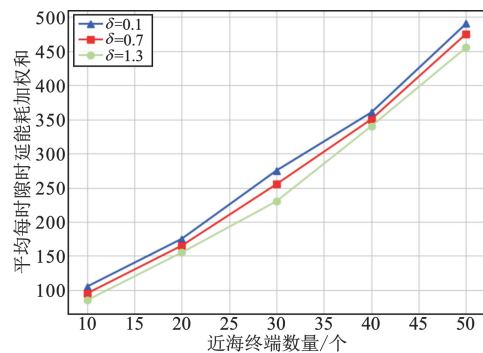


图 10 环境参数  $\delta$  与平均每时隙时延能耗加权和对比如图的关系

## 5 结束语

本文针对近海通信网络对通信系统的低时延与低能耗需求,搭建了 MEC 辅助近海通信框架,对服务器任务软件动态缓存的时延能耗加权和进行了仿真研究,提出了基于 DDQN 的 MEDSCP 算法。通过与 MFU-MEC、FIFO-MEC 和无缓存的卸载算法进行对比,证明 MEDSCP 算法在降低系统时延和能耗方面具有更优秀的性能。这是因为 MEDSCP 算法通过 DNN 的学习能够获得更高的预测准确性,增加近海终端选择有缓存的计算卸载的几率,减少传输过程中的数据大小,从而达到降低所执行任务的时延能耗加权和的目的,满足了近海通信低时延容忍度和近海终端长续航的需求。为了解决深度强化学习(Deep Reinforcement Learning, DRL)算法中动作空间复杂的问题,本文通过修改 DNN 的输入层和输出层神经元连接结构,减少 DNN 神经元间的冗余连接,加快算法收敛速度,扩展 DRL 算法应用于多维度动作的场景。但是,在算法设计和神经网络训练中,会存在一些情况导致算法在理论上满足 MEDSCP 算法复杂度降低的条件,却在实际训练过程中该条件不发挥作用导致算法的复杂度无法降低。比如,学习率等超参数选择不当或未能正确解调,都会导致算法未能利用到复杂度降低的机制,从而理论与实际出现偏差。

综上所述,本文提出的 MEDSCP 算法为近海通信网络下的系统时延与能耗的优化问题提供了新的思路和方法,并为相关研究提供了有价值的参考,但也带来了新的思考:如何才能尽可能地避免这种理论与实际的偏差,这将是后续需要研究的方向。

### 参考文献:

- [ 1 ] 贺硕. 移动边缘网络缓存技术研究[D]. 北京:北京邮电大学,2020.
- [ 2 ] DAI M H, WU Y, QIAN L P, et al. UAV-assisted multi-access computation offloading via hybrid NOMA and FDMA in marine networks [J]. IEEE Transactions on Network Science and Engineering, 2023, 10(1): 113–127.
- [ 3 ] ALQURASHI F S, TRICHILI A, SAEED N, et al. Maritime communications: a survey on enabling technologies, opportunities, and challenges [J]. IEEE Internet of Things Journal, 2023, 10(4): 3525–3547.
- [ 4 ] SU X, MENG L L, HUANG J. Intelligent maritime networking with edge services and computing capability [J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13606–13620.
- [ 5 ] LIU C X, FENG W, CHEN Y F, et al. Cell-free satellite-UAV networks for 6G wide-area Internet of Things [J]. IEEE Journal on Selected Areas in Communications, 2021, 39(4): 1116–1131.
- [ 6 ] YANG B, FAGBOHUNGBE O, CAO X L, et al. A joint energy and latency framework for transfer learning over 5G industrial edge networks [J]. IEEE Transactions on Industrial Informatics, 2022, 18(1): 531–541.
- [ 7 ] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading [J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628–1656.
- [ 8 ] TRAN T X, POMPILO D. Adaptive bitrate video caching and processing in mobile-edge computing networks [J]. IEEE Transactions on Mobile Computing, 2019, 18(9): 1965–1978.
- [ 9 ] CHEN M, HAO Y X. Task offloading for mobile edge computing in software defined ultra-dense network [J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587–597.
- [ 10 ] WEN W L, CUI Y, QUEK T Q S, et al. Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing [J]. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7879–7894.
- [ 11 ] YANG X L, FEI Z S, ZHENG J C, et al. Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing [J]. IEEE Transactions on Vehicular Technology, 2019, 68(11): 11018–11030.
- [ 12 ] XING H, CUI J J, DENG Y S, et al. Energy-efficient proactive caching for fog computing with correlated task arrivals [C] // 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications. Cannes: IEEE, 2019: 1–5.
- [ 13 ] DAI J M, ZHANG Z L, MAO S W, et al. A view synthesis-based 360° VR caching system over MEC-enabled C-RAN [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(10): 3843–3855.
- [ 14 ] YAN J, BI S Z, DUAN L J, et al. Pricing-driven service caching and task offloading in mobile edge computing [J]. IEEE Transactions on Wireless Communications, 2021, 20(7): 4495–4512.
- [ 15 ] CHEN Z X, YI W Q, ALAM A S, et al. Dynamic task software caching-assisted computation offloading for multi-access edge computing [J]. IEEE Transactions on Communications, 2022, 70(10): 6950–6965.
- [ 16 ] LIU Y, YAN J J, ZHAO X H. Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network [J]. IEEE Transactions on Vehicular Technology, 2022, 71(4): 4225–4236.
- [ 17 ] GAO M J, SHEN R J, SHI L, et al. Task partitioning and

- offloading in DNN-task enabled mobile edge computing networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(4):2435-2445.
- [18] WANG H H, WANG Y, MA Y H, et al. Resource allocation for OFDM-based maritime edge computing networks [C]//2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics. Chengdu:IEEE, 2020:983-988.
- [19] SINGH S, DHILLON H S, ANDREWS J G. Offloading in heterogeneous networks: modeling, analysis, and design insights [J]. IEEE Transactions on Wireless Communications, 2013, 12(5):2484-2497.
- [20] CHEN C L, BRINTON C G, AGGARWAL V. Latency minimization for mobile edge computing networks [J]. IEEE Transactions on Mobile Computing, 2023, 22(4):2233-2247.
- [21] WU M R, LI K X, QIAN L P, et al. Secure computation offloading and service caching in mobile edge computing networks [J]. IEEE Communications Letters, 2024, 28(2):432-436.
- [22] QIU Y, LIANG J B, LEUNG V C M, et al. Online reliability-enhanced virtual network services provisioning in fault-prone mobile edge cloud[J]. IEEE Transactions on Wireless Communications, 2022, 21(9):7299-7313.
- [23] CHEN Z X, YI W Q, ALAM A S, et al. Dynamic task software caching-assisted computation offloading for multi-access edge computing [J]. IEEE Transactions on Communications, 2022, 70(10):6950-6965.
- [24] WU P Y, LI J, SHI L, et al. Dynamic content update for wireless edge caching via deep reinforcement learning [J]. IEEE Communications Letters, 2019, 23(10):1773-1777.
- [25] XU Y, ZHANG T K, LIU Y W, et al. UAV-assisted MEC networks with aerial and ground cooperation [J]. IEEE Transactions on Wireless Communications, 2021, 20(12):7712-7727.
- [26] ZHAN C, HU H, SUI X F, et al. Completion time and energy optimization in the UAV-enabled mobile-edge computing system [J]. IEEE Internet of Things Journal, 2020, 7(8):7808-7822.

### 作者简介:

庄颖瑜 女, 2000 年生于广西钦州, 2022 年获学士学位, 现为硕士研究生, 主要研究方向为海洋信息环境感知、空天海地一体化、动态资源管理等。

潘春雨 女, 1989 年生于河南新野, 2019 年获博士学位, 现为副教授, 主要研究方向为基于人工智能的移动通信网络、计算与资源管理技术、无人机通信技术。

李学华 女, 1977 年生于湖北荆州, 2008 年获博士学位, 现为教授, 主要研究方向为无线通信关键技术、物联网技术、智能边缘技术等。