研究与开发

6G网络任务卸载与细粒度切片资源调度联合优化算法

王晔[1]，王逸飞[2]，陈康[2]，朱晓荣[2]，童恩[3]，徐语菲[1]
（1. 江苏移动信息系统集成有限公司，江苏 南京 210013；
2. 南京邮电大学通信与信息工程学院，江苏 南京 210003；
3. 中国移动通信集团江苏有限公司，江苏 南京 210013）

摘　要：针对未来全域、全场景多样化的业务需求，6G网络需要提供场景化、个性化的服务能力。针对未来细粒度业务服务质量保障问题，提出了6G网络任务卸载与细粒度切片资源调度联合优化算法，联合考虑多MEC的计算卸载和网络切片的资源调度，在有限的资源内，最小化任务的执行时延和能耗成本，并采用异步训练的A3C强化学习算法进行求解。仿真结果表明，对比传统算法，该算法可以在满足用户业务需求的情况下降低计算成本，并且算法收敛速度快，可以实现快速决策。

关键词：6G网络；任务卸载；细粒度切片；多维资源；联合优化
中图分类号：TN92
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2024144

Joint optimization algorithm for 6G network task offloading and fine-grained slice resource scheduling

WANG Ye[1], WANG Yifei[2], CHEN Kang[2], ZHU Xiaorong[2], TONG En[3], XU Yufei[1]
1. Jiangsu Mobile Information System Integration Co., Ltd., Nanjing 210013, China
2. School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
3. China Mobile Communications Group Jiangsu Co., Ltd., Nanjing 210013, China

Abstract: In response to the diverse business needs of the future in all domains and scenarios, 6G networks need to provide scenario-based and personalized service capabilities. Aiming at the problem of quality assurance of fine-grained business services in the future, a joint optimization algorithm for 6G network task offloading and fine-grained slice resource scheduling was proposed, which jointly considered the calculation offloading of multiple MECs and the

收稿日期：2023−11−26；**修回日期：**2024−04−20
基金项目：国家自然科学基金资助项目（No.61871237， No.92067101）；江苏省高校“青蓝工程”基金资助项目；江苏省重点研发计划项目（No.BE2021013-3）；江苏省研究生科研与实践创新计划基金资助项目（No.KYCX21_0733）
Foundation Items: The National Natural Science Foundation of China (No.61871237, No.92067101), Program to Cultivate Middle-aged and Young Science Leaders of Universities of Jiangsu Province, The Key Research and Development Program of Jiangsu Province (No.BE2021013-3), Jiangsu Province Postgraduate Research and Practice Innovation Program (No.KYCX21_0733)

resource scheduling of network slices, and minimized the execution delay and energy consumption cost of the task within limited resources. Then the A3C reinforcement learning algorithm of asynchronous training was used to solve it. The simulation results show that, compared with the traditional algorithm, the proposed algorithm can reduce the computing cost while meeting the business needs of users. Additionally, the algorithm converges fast and can realize fast decision-making.

Key words: 6G network, task offloading, fine-grained slicing, multi-dimensional resource, joint optimization

0　引言

未来的6G网络将形成全域覆盖、全息通信、海量连接、泛在智能等场景，对比5G网络，服务将根据业务进行更细致的划分，将产生更多的指标和功能，例如比特率、流量容量、覆盖率、服务可用性和可持续性等，即便是同一指标也需要更细致的划分，提供差异化的服务，这些都需要网络提供更细致的定制化、场景化服务[1-2]。

6G网络将基于云化的基础设施，业务功能和网络功能都会向着原子化的方向发展。6G网络将呈现分布式的特点，可以根据实时业务需求和实时网络资源状况，自动部署资源池和弹性扩缩等，实现网络规模的自适应调整。可变的网络架构使得6G可以满足更加复杂的业务场景，将促进行业应用的快速发展，因此，对设备也提出了更高的需求。越来越多的时延敏感型任务出现，设备有限的能量和算力无法满足任务需求，卸载到核心云上将产生大量的传输时延，影响任务的执行[3]。移动边缘计算（mobile edge computing，MEC）技术通过在边缘侧部署边缘服务器，提供计算资源，使用户可以选择性地将任务卸载到一些网络的边缘节点上。MEC技术可以减轻设备执行任务的能耗和计算能力要求，同时减少传输时延，可以满足日益复杂的计算任务要求，提升应用的服务质量[3-7]（quality of service，QoS）。因此，MEC技术是未来网络发展不可或缺的关键技术之一。目前对MEC的研究主要包括任务卸载决策、时延优化和系统能效提升等。卸载决策问题会涉及卸载方式和卸载位置选择等方面，如本地执行、完全卸载和部分卸载等卸载方式，以及边缘云卸载和中心云卸载等卸载位置。对于MEC的计算卸载场景研究主要分为单MEC服务器卸载和多MEC服务器卸载。在单MEC服务器的卸载方面，文献[8]针对任务卸载和传输调度问题，综合考虑任务的等待状态、本地处理和传输单元的状态，采用马尔可夫决策的建模过程，提出了一个功率受限条件下的时延最小化的问题，并提出了一种有效的搜索算法解决问题，找到最优的任务调度策略。文献[9]将用户的任务进行了拆分，把拆分后可以并行运算的子任务分别送到MEC服务器或者本地进行运算，并使用时分多址和频分多址传输模式卸载数据，分别设计了卸载策略，通过分层的思想完成求解。在多MEC服务器的卸载方面，文献[10]针对网络计算任务分配不均问题，从负载均衡角度出发，通过Lyapunov技术开发了一种在线对等卸载，以最大化长期系统性。通过MEC间的协作，提升了MEC性能。文献[11]考虑了MEC网络中能效较低的问题，将能耗和时延成本共同考虑，综合考虑传输链路的选择以及路由优化等方面，提出了一种可以实现全局最优解的低复杂度资源分配算法。以上研究通过将卸载技术和实际场景结合，可以有效地减少决策时间，提升MEC性能，但没有考虑资源分配对卸载决策的影响。

在使用边缘计算技术时，需要根据当前资源计算本地和远程卸载的收益，判断是否业务需要上传至MEC，而用户卸载的情况也会影响到切片

资源的划分。因此，通常将卸载决策和资源分配进行联合考虑[12-15]，可以进一步提升MEC服务器的性能。由于卸载变量往往是离散变量，而资源分配却是连续变量，因此二者联合的优化问题往往是NP问题，主要求解方法包含子问题分解和强化学习等。对于单MEC服务器，文献[12]考虑用户的发射功率分配问题，提出了卸载决策和功率、计算等资源分配的联合优化问题，并将优化问题分解成请求卸载和资源分配两个子问题，采用多目标的优化算法完成求解。文献[13]在满足时延约束的同时，联合优化卸载决策、无线通信和计算资源分配。使用了考虑长期目标的强化学习方法求解问题。在多MEC服务器方面，文献[14]综合考虑了非静态的网络环境以及分布式任务，将计算卸载和资源分配联合优化，并建模为马尔可夫决策过程，提出了一种基于深度确定性的优势策略梯度的强化学习算法，可以降低系统的能耗。文献[15]采用了多智能体强化学习的方式，解决了任务卸载和资源的联合分配问题，可以减少云服务器的通信和计算资源竞争。

6G网络将是超大规模的复杂动态网络，在用户需求、网络资源和流量负载等方面都是多维动态的，可以将大量智能设备动态地连接到提供最佳服务质量的网络，满足多样化的业务场景需求。同时，密集部署、区块链、人工智能和多节点协同等新技术的引入将增加网络的复杂性。由于网络向着密集复杂化的方向发展，对于用户设备有多个可选择接入的基站和切片，这意味着有多个MEC服务器可提供卸载服务。MEC会将计算资源分配到服务基站上部署的切片，因此，对于卸载的选择，除了MEC服务器的选取，还需进行切片维度的选取。

异步优势动作评价（asynchronous advantage actor-critic，A3C）算法是演员-评论家（actor-critic，AC）算法发展而来的一种强化学习方法。在AC框架中，有一个代理通过与环境交互来最大化累积回报，它包含两部分：演员和评论家。演员负责根据当前策略在给定状态下执行动作，而评论家则使用时间差分算法更新，以更好地评价和打分，并指导演员更新策略以获得更高的回报。A3C算法的特点在于它使用了多个并发工作的演员来异步训练神经网络，这样可以显著加快收敛速度。A3C有一个中央服务器存储网络参数，每个代理在达到最大动作索引或终止状态时计算梯度并将其发送到中央服务器。中央服务器更新全局参数并将其分发给各代理，确保它们共享相同的策略。这种方法使参数间的相关性降低，无须像传统的深度Q网络（deep Q-network，DQN）那样维护回放记忆，同时也大大缩短了训练时间。A3C算法因其基于策略的更新、基于步骤的更新以及异步训练的特点而优于许多现有的深度强化学习算法。基于策略的更新使A3C能在连续空间中进行搜索，处理具有巨大状态或动作空间的问题；基于步骤的更新提高了工作效率；多个并行工作的代理能有效加速训练并有效探索解空间。在通信领域，A3C算法已经被探索用于多种研发应用，包括资源分配、网络流量控制和计算卸载等。由于A3C算法能够高效处理大规模的离散动作空间，并且通过其多线程异步训练框架显著提高了收敛速度，它被认为是解决通信系统中复杂决策问题的有力工具。特别是在车辆网络和下一代无线网络中，基于A3C的学习算法被用于智能地分配通信和计算资源，以改善网络性能。

本文考虑多用户多MEC的细粒度切片的场景，用户选择最适合的MEC服务器和网络切片，将MEC的计算卸载和网络切片的资源调度联合考虑，在有限的资源内，最小化任务执行时延和能耗成本。考虑到切片维度增加的计算复杂度，本文提出采用异步训练的A3C强化学习算法，减少训练时间，完成问题求解。

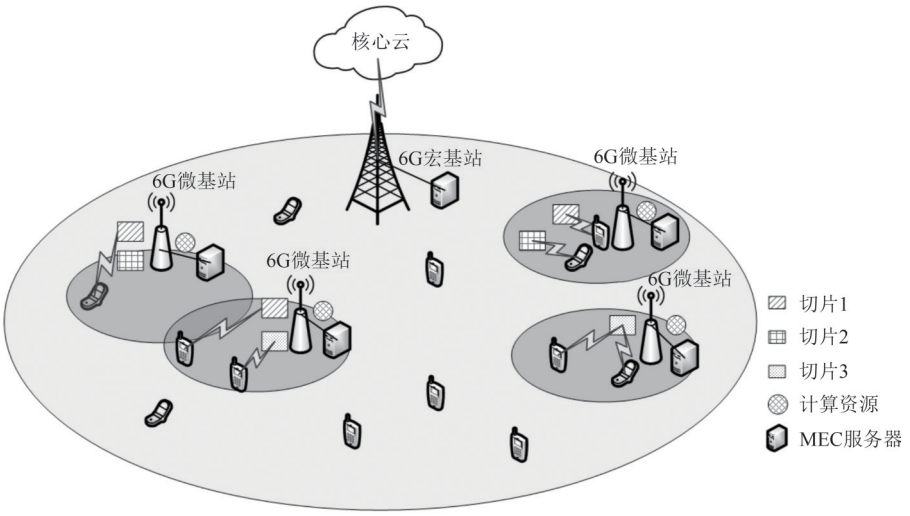

图1　网络架构

1　网络模型

本文考虑的网络架构如图1所示，网络由一个宏基站、多个微基站和用户设备组成，其中宏基站负责完成全局决策，不提供用户具体服务。相对地，每个微基站都配备了一个MEC单元，专门负责处理计算密集型的任务，这些微基站通过有线网络与核心网连接，确保数据传输的高速率和低时延。假设网络中有M个微基站，用$\boldsymbol{M}=\{1,2,3,\cdots,m,\cdots,M\}$表示，并部署了$N$个切片，用$\boldsymbol{N}=\{1,2,3,\cdots,n,\cdots,N\}$表示，每个切片分配了特定的最大时延阈值，用$\boldsymbol{T}^{\max}=\{T_1^{\max},T_2^{\max},\cdots,T_N^{\max}\}$表示。为了减少子信道间的干扰，系统采用正交频分复用技术，并将频带资源划分为B个子信道，由于业务之间的差异性，不同切片分得的频谱资源也有所不同，分别用$\{b_1,b_2,\cdots,b_N\}$表示。在微基站的覆盖范围内假设有K个用户设备，它们通过微基站接入网络，每个设备只能连接到一个网络切片，并且系统可以将多连接的设备等价于多个单连接设备。本文主要考虑低时延和高速率业务，基于业务指标，将设备分为大速率设备k_r和低时延设备k_u两大类。设备k产生的计算密集型任务表示为$\boldsymbol{Q}^k=(D^k,\Omega^k,T_{\max}^k)$，$D^k$表示任务的数据大小，$\Omega^k$表示执行任务需要的计算资源，$T_{\max}^k$表示任务的最大容忍时延，如果任务的执行时延超出最大容忍时延，则任务失败。只有满足最大时延要求的切片才能为设备k提供卸载服务。需要注意的是，低时延设备k_u对时延的要求更为严格，对应地，它们的任务参数相对较小。此外，本文仅考虑完全卸载模型，即任务只能在本地或者MEC执行，不考虑任务分割执行的情况。

1.1　通信模型

在网络架构中，对于切片的部署需要从运营商租借信道和计算资源。将基站上的切片资源表示为$\boldsymbol{N}_{m,n}=(b_{m,n},F_{m,n})$，其中$b_{m,n}$表示切片$n$在基站$m$上占用的信道资源，$F_{m,n}$表示切片$n$在基站$m$上占用的计算资源。为确保切片之间的隔离性，假设在任意时刻，一个基站上不同切片所占用的信道和计算资源都是互斥的，即每个切片都使用独立的资源，不会出现资源共享的情况。

当任务需要卸载到MEC进行处理时，会涉及任务数据的上传过程。假设卸载过程中的场景是静态的，即设备的位置和信道条件在任务卸载期间不会发生显著变化。对于高速率业务，根据香农公式，连接基站m上切片n的设备k的传输速率可以表示为：

$$R_{m,n}^{k}=b_{m,n}^{k}\,\mathrm{lb}\,(1+\gamma) \tag{1}$$

$$\gamma=\frac{p^{k}g_{m}^{k}}{n_{0}b_{m,n}^{k}+\sum\limits_{m'\in M}\sum\limits_{n'\in N}\sum\limits_{k'\neq k}p^{k'}g_{m'}^{k'}\alpha_{m',n'}^{k'}} \tag{2}$$

其中，$b_{m,n}^{k}$表示设备k使用的信道资源，p^{k}表示设备k的发射功率，g_{m}^{k}表示设备k到基站m的信道增益，$\alpha_{m',n'}^{k'}$表示连接基站m上切片n的设备k和连接基站m'上切片n'的设备k'是否占用相同的频段资源，如果占用则$\alpha_{m',n'}^{k'}=1$，否则$\alpha_{m',n'}^{k'}=0$。此外，干扰除了高斯白噪声之外，还受其他设备的干扰。对于低时延业务，由于数据包较小，传输速率可以依据有限长块码理论来计算，此时连接基站m上切片n的设备k的传输速率可以表示为：

$$R_{m,n}^{k}\approx\frac{b_{m,n}^{k}}{\ln 2}\left[\ln(1+\gamma)-\sqrt{\frac{V}{L_{\mathrm{B}}}}f_{Q}^{-1}(\varepsilon_{c})\right] \tag{3}$$

其中，L_{B}为块长度，表示一个数据块中的符号数量，其大小可以通过$t\cdot b_{m,n}^{k}$计算得到，t表示数据块的传输时间，$f_{Q}^{-1}(\cdot)$表示Q函数的逆函数，ε_{c}表示解码错误概率，V表示信道色散系数，表达式为：

$$V=1-\frac{1}{(1+\gamma)^{2}} \tag{4}$$

1.2 任务卸载模型

本文考虑完全卸载的任务执行方式，即任务执行可以在本地进行，也可以在MEC处进行，但不能同时进行。对于卸载策略的选择，假设设备k选择基站m上的切片n提供服务，使用$\varphi_{m,n}^{k}$表示设备k的卸载策略，$\varphi_{m,n}^{k}=0$表示任务在本地执行，否则表示任务卸载到MEC处执行，可表示为：

$$\varphi_{m,n}^{k}=\begin{cases}0,\text{ 在本地执行}\\1,\text{ 卸载到MEC}\end{cases} \tag{5}$$

（1）本地计算模型

本地计算主要依赖设备的性能，即在本地设备上直接执行计算任务。假设设备k的计算能力为F_{loc}^{k}，其中计算能力指的是单位时间内CPU运行的周期数。设备k的每个CPU周期的能耗表示为$\sigma^{k}=\rho^{k}(F_{\mathrm{loc}}^{k})^{2}$，其中$\rho^{k}$表示设备一个CPU周期的能耗系数，这个值与设备芯片的架构有关，通常取10^{-27}，则任务处理的时延和能耗分别表示为：

$$T_{\mathrm{loc}}^{k}=\frac{\Omega^{k}}{F_{\mathrm{loc}}^{k}} \tag{6}$$

$$E_{\mathrm{loc}}^{k}=\sigma^{k}\Omega^{k} \tag{7}$$

（2）边缘计算模型

当决定通过卸载来执行任务时，需要花费额外的时延和能耗去上传任务的相关信息，由边缘云执行具体的计算操作。其中任务的完成过程主要包含3个部分。

1）用户设备需要分配一定的信道资源$b_{m,n}^{k}$，通过上传链路将任务数据上传到基站和MEC服务器。

2）MEC服务器完成计算密集型任务的执行，执行过程会消耗MEC的计算资源。计算资源的多少决定了计算时延的大小。

3）MEC服务器完成计算后，将计算的结果发回用户设备。

假设设备k选择基站m上的切片n提供服务，在数据上传过程中，需要根据业务的类型选择对应的传输速率$R_{m,n}^{k}$，其时延和能耗分别表示为：

$$T_{m,n}^{k}=\frac{D^{k}}{R_{m,n}^{k}} \tag{8}$$

$$E_{m,n}^{k}=\frac{p^{k}D^{k}}{R_{m,n}^{k}} \tag{9}$$

其中，p^{k}表示设备k的发射功率，即上传数据的传输功率，不会超过设备的最大功率$p_{\max}^{k}$。

对于计算过程，假设MEC服务器为用户分配的计算能力为$F_{m,n}^{k}$，则计算的时延表示为：

$$T_{\mathrm{mec}}^{k}=\frac{\Omega^{k}}{F_{m,n}^{k}} \tag{10}$$

此时，设备维持也会产生能耗，即：

$$E_{\mathrm{mec}}^{k}=p_{m}^{k}\frac{\Omega^{k}}{F_{m,n}^{k}} \tag{11}$$

p_m^k表示设备k的维持接口状态的功率。

对于基站m上的切片n，服务所有用户的计算资源和不会超过MEC服务器分配给切片的计算资源，即：

$$\sum_{k=1}^{K} F_{m,n}^k \leqslant F_{m,n} \tag{12}$$

对于回传过程，由于仅需要传输计算的结果，数据量一般较小，时延和能耗较低，因此不再考虑回程时延和能耗。

由于本地计算或者边缘计算可由$\varphi_{m,n}^k$表示，则任务卸载总体的时延和功耗可以表示为：

$$T_{\text{total}}^k = \varphi_{m,n}^k (T_{m,n}^k + T_{\text{mec}}^k) + (1-\varphi_{m,n}^k) T_{\text{loc}}^k \tag{13}$$

$$E_{\text{total}}^k = \varphi_{m,n}^k (E_{m,n}^k + E_{\text{mec}}^k) + (1-\varphi_{m,n}^k) E_{\text{loc}}^k \tag{14}$$

2　问题形成与算法求解

2.1　问题形成

本文综合考虑了任务的时延和能耗成本，将任务的计算成本c^k表示为：

$$c^k = \lambda T_{\text{total}}^k + \mu E_{\text{total}}^k \tag{15}$$

$$\lambda + \mu = 1 \tag{16}$$

其中，λ、μ分别表示对时延、能耗指标的偏好程度，可根据任务的需求目标调整。当服务对时延较敏感时，增加λ比重，当设备能源不足时，增加μ比重，一般可取0.5。本文的研究目标是最小化设备的总任务成本：

$$\begin{aligned}
&\min \sum_{k=1}^{K} c^k \\
&\text{C1:} \sum_{k=1}^{M} \sum_{n=1}^{N} \varphi_{m,n}^k \leqslant 1,\ \varphi_{m,n}^k \in \{0,1\} \\
&\text{C2:} \sum_{k=1}^{K} b_{m,n}^k \leqslant b_{m,n} \\
&\text{C3:} \sum_{k=1}^{K} F_{m,n}^k \leqslant F_{m,n} \\
&\text{C4:} 0 \leqslant p^k \leqslant p_{\max}^k \\
&\text{C5:} F_{\text{loc}}^k, F_{m,n}^k > 0 \\
&\text{C6:} T^k \leqslant T_{\max}^k
\end{aligned} \tag{17}$$

其中，C1表示设备k的任务只能选择本地执行或卸载到唯一基站的唯一切片，C2表示基站m切片n服务所有设备消耗的频谱资源不能超过基站m给切片n分配的频谱资源，C3表示基站m切片n上所有设备消耗的计算资源不能超过基站m给切片n分配的计算资源，C4表示设备的发射功率不能超过设备的最大发射功率，C5表示设备k的任务计算能力大于0，C6表示设备执行任务的时延应满足任务的最大时延要求。

在考虑的场景中，任务卸载与资源分配是两个相互依赖的方面，其中任务卸载策略是整数变量，而资源分配是连续变量。目标是在满足多业务需求的前提下，最小化任务的计算成本。考虑到切片层面的资源分配，每个MEC服务器需要在各自服务的基站上进行切片资源的配置。用户在选择卸载任务时，不仅需要选择MEC服务器，还需要选择具体的切片，这一过程的复杂性会导致时延的增加，从而影响用户体验。利用强化学习方法，特别是A3C算法，可以有效减少决策时间。A3C算法适合于处理离散和连续的动作，通过本地网络的并行训练，可以减少训练相关性，加快训练速度。此外，当考虑切片维度后，总体的决策选择有了明显增加，因此考虑将MEC服务器当作智能体，在MEC服务器进行本地训练，通过不断交互来优化卸载和资源分配策略，并将这些智能体的学习结果用于全局网络的更新，实现卸载决策和资源分配的高效联合求解。

下面进行学习模型的设计。

（1）状态空间$\boldsymbol{S}$。综合考虑计算任务的特性以及MEC服务器在各切片上分配的计算资源，本文考虑的状态为t时刻任务的信息，即任务k的数据大小D^k，执行任务的计算资源Ω^k，任务的最大容忍时延$T_{\max}^k$，以及各MEC服务器可用的计算资源和无线资源，所以t时刻的状态$\boldsymbol{s}_t$可表示为：

$$\boldsymbol{s}_t=\{D^k,\Omega^k,T^k_{\max},F_{1,1},\cdots,F_{m,n},\cdots,F_{M,N},b_{1,1},\cdots,b_{m,n},\cdots,b_{M,N}\} \quad (18)$$

对于切片和基站的分配，并不是所有基站都包含所有切片，当基站m不包含切片n时，对应的$F_{m,n}$和$b_{m,n}$则为0。

（2）动作空间$\boldsymbol{A}$。智能体MEC服务器通过与环境交互做出卸载和资源分配结果，因此动作就是选择MEC和切片并分配资源来处理任务，可表示为：

$$\boldsymbol{A}=[a^k_{\text{loc}},\cdots,a^k_{m,n},\cdots,a^k_{M\times N},b^k,p^k,F^k] \quad (19)$$

当任务由本地执行时，设置a^k_{loc}为1，其余置0；若任务卸载到基站m上的切片n，则对应的$a^k_{m,n}$置1，其余置0。

（3）奖励值R。奖励值是从先前的动作获得利益的评估。由于本文的优化目标是最小化任务计算成本，但强化学习的目标是获得最大的回报，因此需要对优化目标进行处理，可以考虑对原优化目标取负数，即奖励值R表示为：

$$R=-c^k \quad (20)$$

2.2 基于A3C的任务卸载和资源分配算法

本文使用基于A3C的算法完成卸载决策，利用本地网络异步并行训练，基于A3C强化学习的任务卸载和资源分配算法框架如图2所示，每个智能体通过和环境交互，即Action网络完成卸载策略的选择，Critic网络完成对策略的评价；Action网络根据Critic网络的评价优化策略的选择。一轮训练结束后，本地网络将训练结果上传到全局网络，完成全局网络更新，并在下一轮训练前，将全局网络参数同步更新到本地。每个智能体各自训练，互不干扰，可以消除训练相关性，加快收敛。

A3C算法除了异步并行的优点外，在参数更新上也有提升。A3C算法既有值函数估计又有策略函数估计，因此需要更新计算的两个主要参数：值函数$V(s_t,\theta_v)$的参数θ_v和策略函数$\pi(a_t|s_t;\theta)$的参数θ。

对于更新值函数的参数θ_v时，使用n步累积的梯度下降法：

$$\mathrm{d}\theta_v\leftarrow\mathrm{d}\theta_v+\frac{\partial(R-V(s_i;\theta'_v))^2}{\partial\theta'_v} \quad (21)$$

其中，θ'_v是本地网络值函数的参数。

对于更新策略函数的参数θ时，使用梯度上升法，可以看作：

$$\mathrm{d}\theta\leftarrow\mathrm{d}\theta+\nabla_{\theta'}\,\mathrm{lb}\,\pi(a_t|s_t;\theta')(R-V(s_t;\theta'_v)) \quad (22)$$

其中，θ'是本地网络策略函数的参数。

关于算法具体流程如下。首先，对A3C的全局网络参数，包括全局Action网络参数θ和全局Critic网络参数θ_v，以及对各本地网络的Action、Critic网络参数θ'和θ'_v进行初始化。然后，在每次训练时，每个本地网络首先使用全局网络参数同步自己的参数，获得当前的系统状态s_t。同步后各本地网络同时且独立地与环境交互，根据策略$\pi(a_t|s_t;\theta')$，选择动作a_t，计算该策略下的时延和功耗，获得奖励$R(s_t,a_t)$并得到新的状态s_{t+1}。重复以上过程，直到达到最后一个任务或迭代次数的最大值，计算状态s_t的值函数$V(s_t,\theta'_v)$，并得到计算目标函数值。接下来，在并行的训练过程中，根据累积梯度函数，更新本地网络的参数θ'和θ'_v。之后将本地网络的更新参数发送到全局网络，以更新全局网络的参数θ和θ_v。不断重复上述训练过程，直至到达收敛状态，此时得到了预期累计收益最大的策略，为最优策略。

在A3C算法的背景下，任务卸载和资源分配的接口与需求匹配主要涉及如何将多个任务有效地分配给不同的计算资源，同时确保任务之间的依赖和优先级得到妥善处理。接口和需求匹配的描述从以下几个方面进行。

（1）任务描述接口：描述各个任务，包括任务的类型、所需计算资源、执行时间、优先级以及与其他任务的依赖关系等。这有助于算法理解每个任务的具体需求，并据此做出决策。

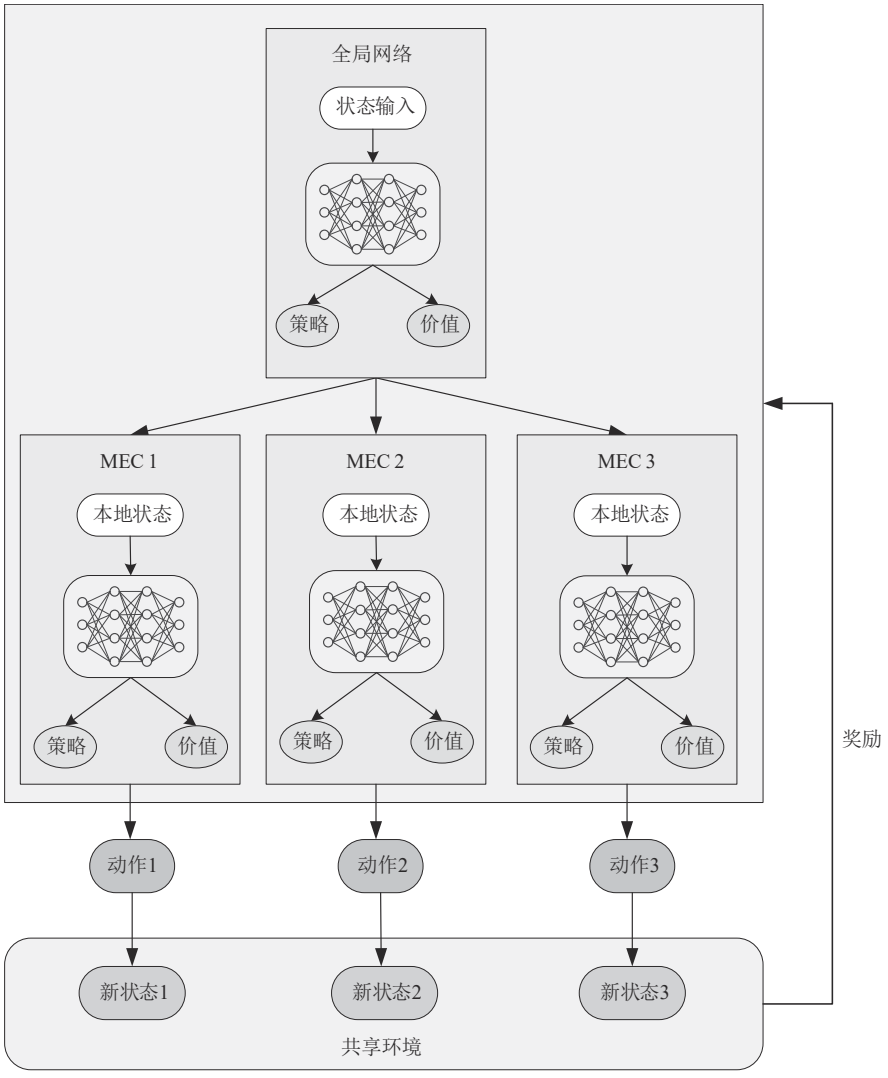

图2　基于A3C强化学习的任务卸载和资源分配算法框架

（2）资源描述接口：描述可用的计算资源，包括每种资源的类型、数量、性能指标（如CPU速度、内存大小等）、当前状态以及可能的访问限制等。这有助于算法了解当前可用的资源情况，以便进行有效的资源分配。

（3）卸载决策接口：算法需要一个接口来执行任务卸载决策，即决定哪些任务应当在本地执行，哪些任务应当卸载到云端或边缘计算节点执行。这个接口应当能够根据任务的特性、资源的可用性以及可能的网络条件等因素，动态地做出最优或近似最优的卸载决策。

（4）资源分配与调度接口：除了决定任务的卸载之外，算法还需要通过另一个接口来进行资源的分配和任务的调度。这包括决定每个任务应当使用哪些具体资源、在何时开始执行，以及如何处理任务之间的依赖和冲突等。

（5）性能反馈接口：为了优化决策过程，算法需要从系统中获取执行任务的性能反馈，包括任务执行时间、资源使用率、能耗等指标。这需要一个反馈接口，以便算法根据实际运行情况调整其策略。

（6）动态适应接口：鉴于计算环境和任务需

求可能会动态变化，算法还需要一个接口来监控这些变化，并相应地调整卸载和资源分配决策。这意味着系统需要为算法实时地提供环境和任务状态的更新信息。

基于A3C的任务卸载和资源分配算法流程如下。

算法1 基于A3C的任务卸载和资源分配算法

输入：$\boldsymbol{s}_t=\left\{D^k,\Omega^k,T^k_{\max},F_{1,1},\cdots,F_{m,n},\cdots,F_{M,N},b_{1,1},\cdots,b_{m,n},\cdots,b_{M,N}\right\}$

输出：$\boldsymbol{A}=[a^k_{\text{loc}},\cdots,a^k_{m,n},\cdots,a^k_{M\times N},b^k,p^k,F^k]$

设置全局网络的共享参数、θ和θ_v，全局最大迭代次数$T_{\max}$，全局共享计数器$T=0$

设置本地网络参数θ'和θ'_v

初始化本地计数器$t\leftarrow 1$

初始化边缘计数器的资源和实验参数

repeat

重置全局网络参数：$\mathrm{d}\theta\leftarrow 0$，$\mathrm{d}\theta'\leftarrow 0$

使用全局网络参数来同步更新本地网络参数：$\theta'=\theta$，$\theta_v=\theta'_v$

令$t_{\text{start}}=t$

得到初始状态$\boldsymbol{s}_t=\left\{D^k,\Omega^k,T^k_{\max},F_{1,1},\cdots,F_{m,n},\cdots,F_{M,N},b_{1,1},\cdots,b_{m,n},\cdots,b_{M,N}\right\}$

repeat

基于策略选择$\pi(a_t|s_t;\theta')$，选择动作a_t，并计算该策略下的时延和功耗

通过计算的时延和功耗，得到奖励值r_t并获得新的状态s_{t+1}

计算状态s_t的值函数$V(s_t,\theta_v)$

令$T\leftarrow T+1$，$t\leftarrow t+1$

until s_t为终止状态，或者$t-t_{\text{start}}=t_{\max}$

如果s_t为终止状态，$R=0$

否则$R=V(s_t,\theta_v)$

for $i\in\{t-1,\cdots,t_{\text{start}}\}$ do

计算每个时刻的函数值$R\leftarrow r_i+\gamma R$

计算本地参数θ'的累积梯度：

$\mathrm{d}\theta\leftarrow d\theta+\nabla_{\theta'}\text{lb}\,\pi(a_t|s_t;\theta')(R-V(s_t;\theta'_v))$

计算本地参数θ'_v的累积梯度：$\mathrm{d}\theta_v\leftarrow\mathrm{d}\theta_v+\frac{\partial(R-V(s_i;\theta'_v))^2}{\partial\theta'_v}$

end

分别使用本地累积梯度$\mathrm{d}\theta$和$\mathrm{d}\theta_v$异步更新全局参数θ和θ_v

until $T>T_{\max}$

由上述算法的步骤可以得出，A3C算法的时间复杂度取决于动作网络的时间复杂度$O(T_a)$、评价网络的时间复杂度$O(T_c)$、基站和切片的个数M和N、本地网络的个数M和迭代次数$T_{\max}$。总体的A3C算法的时间复杂度为$O(\frac{(T_a(M+N)+T_c)T_{\max}}{M})$，是传统AC算法的$\frac{1}{M}$，所以，A3C算法可以减少训练时间。

2.3 基于匹配理论的任务卸载和资源分配联合优化算法

为比较算法性能，本文将任务拆分成资源分配和任务卸载两个子问题，分别进行求解。第一阶段，根据用户情况完成资源分配；第二阶段，当资源分配确定后，使用匹配算法，在可卸载的基站、切片里选择任务计算成本最小的接入。

对于资源分配主要包括：无线频谱资源、计算资源以及用户功率。其中，用户功率假设都使用最大传输功率。

（1）无线频谱资源

考虑根据任务量分配基站的频谱资源，对于基站m切片n服务的$K_{m,n}$个用户，可表示为：

$$b^k_{m,n}=\frac{D^k}{\sum_{k'=1}^{K_{m,n}}D^{k'}}\cdot b_{m,n} \tag{23}$$

（2）计算资源

关于计算资源的子问题可以表示为：

$$\begin{aligned}&\min \sum_{k=1}^{K} c^k \\ &\text{C1: } \sum_{k=1}^{K} F_{m,n}^k \leqslant F_{m,n} \\ &\text{C2: } F_{m,n}^k > 0\end{aligned} \tag{24}$$

计算目标任务计算成本的二阶导数，可以得到：

$$\frac{\partial^2 c^k}{\partial {F_{m,n}^k}^2} = \frac{2\Omega^k (p_m^k + (1-p_m^k)\lambda)}{{F_{m,n}^k}^3} \geqslant 0 \tag{25}$$

所以计算资源子问题是凸优化问题，可以通过拉格朗日乘子法进行求解。引入拉格朗日乘子ξ形成新的公式：

$$L(c^k, \xi) = c^k + \xi\left(\sum_{k=1}^{K} F_{m,n}^k - F_{m,n}\right) \tag{26}$$

借助KKT条件可得：

$$\frac{\partial L(c^k, \xi)}{\partial F_{m,n}^k} = 0 \tag{27}$$

$$\sum_{k=1}^{K} F_{m,n}^k - F_{m,n} = 0 \tag{28}$$

最终可以得到计算资源的最优分配为：

$${F_{m,n}^k}^* = \frac{\sqrt{\Omega^k (p_m^k + (1-p_m^k)\lambda)}}{\sum_{k'=1}^{K} \sqrt{\Omega^{k'} (p_m^{k'} + (1-p_m^{k'})\lambda)}} F_{m,n} \tag{29}$$

通过以上得到的资源分配方式进行资源分配，用户将从可卸载的基站、切片里选择任务计算成本最小的接入。这里考虑使用Gale-Shapley匹配算法，用户的选择偏好表示为任务计算成本，成本越低，优先级越高。MEC的偏好列表则为任务的最大容忍时延，时延越小，优先级越高。任务卸载的一体化匹配过程如图3所示。

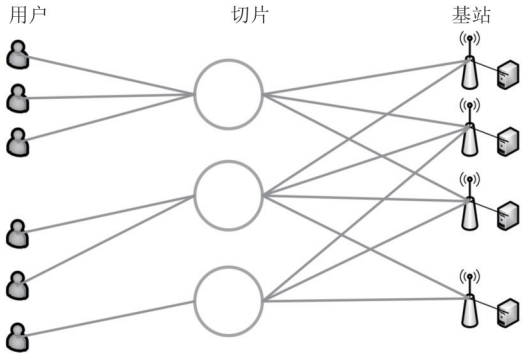

图3　任务卸载的一体化匹配过程

具体的Gale-Shapley算法包括3个阶段。第一阶段，匹配请求者和匹配接收者感知彼此的信息，如用户可以得到切片的保障速率和基站的位置及负载情况，切片和基站可以得到用户的需求速率。第二阶段为初始化阶段，其中所有匹配请求者和匹配接收者需要初始化其偏好列表，然后所有未匹配的匹配请求者放入集合UNMATCH中。第三阶段为核心匹配阶段，主要分为两步：第一步，在集合UNMATCH中的所有匹配请求者向匹配接收者发送匹配请求；第二步，匹配接收者根据优胜劣汰的原则来确定匹配对象，当匹配请求者没有可选择的匹配接收者时，该匹配请求者从集合UNMATCH中移除。重复执行第三阶段的两个步骤直到集合UNMATCH为空。下面为算法的具体过程。

算法2　Gale-Shapley算法

初始化

每个匹配请求者和匹配接收者分别得到自己的偏好列表

将所有未匹配的匹配请求者都放入UNMATCH集合中

while UNMATCH $\neq \varnothing$

for $i \in$ UNMATCH

匹配请求者向列表中未发送过请求的偏好度最高的匹配接收者发出匹配请求

for $j \in$ 匹配接收者

if 可提供匹配请求者速率<匹配请求者速率

将该匹配请求者从偏好列表中移除

end if

end for

for $j \in$ 匹配接收者

if 匹配接收者还未达到最大承载数，则匹配接收者按照自己的偏好序列与匹配请求者临时配对，并将请求者从UNMATCH集合中移除

else if 匹配接收者已达到最大承载数，则匹配接收者按照自己的偏好序列比较当前临时配对对象和当前请求者，选择速率更高的请求者组成临时配对，没有配对者进入UNMATCH集合，将成功配对者从UNMATCH集合中移除

else 匹配接收者未收到匹配请求，则进入下一轮

end if

end for

end for

end while

Gale-Shapley算法与传统单向匹配的匈牙利算法不同，它联合考虑了匹配双方的偏好序列，得到的匹配结果是符合双方期望的最优值。本文为了得到最高的用户速率和，需要提高网络资源的利用率，所以得到的是用户和网络双方匹配的最优结果，Gale-Shapley算法符合求解目标。

3 仿真与分析

本文将从算法收敛情况、系统的任务执行总成本等方面来验证基于A3C联合优化算法的性能。可以通过对比以下几种方案，验证算法的有效性。

（1）任务全部本地执行：所有的任务都由设备本地执行，不会卸载到MEC处执行。

（2）任务全部MEC执行：所有的任务都卸载到MEC处执行，本地不做任务计算。

（3）任务随机卸载执行：任务随机地卸载到MEC处执行，其他由本地完成计算。

（4）基于匹配的任务卸载和资源分配联合优化算法：基于匹配理论，选择任务总成本最小的方式执行。

（5）基于A3C的任务卸载和资源分配联合优化算法：基于A3C强化学习算法，选择任务总成本最小的方式执行。

3.1 仿真参数设置

本文仿真分析使用Python3.6 + Tensorflow环境，利用gym的框架编写仿真环境，并利用MATLAB完成数据处理及仿真。在仿真环境中，使用1个宏基站和4个微基站分布在600 m×600 m的范围内。宏基站负责完成全局决策，不提供用户服务。每个微基站部署不同的细粒度切片，可以提供不同级别的服务。用户大多随机分布在密集覆盖区域，仿真网络参数见表1。

表1 仿真网络参数

网络参数	参数值
仿真区域/m^2	600×600
微基站/MEC数目M	4
切片数目N	6
用户数	60
用户最大发射功率p^k_{max}/W	0.5
基站无线信道带宽/MHz	20
噪声功率/(dBm·Hz^{-1})	−174
任务数据大小D^k/KB	[200, 2 000]
任务计算资源Ω^k/G cycle	[0.2, 2]
任务最大容忍时延T^k_{max}/s	[1, 2]
用户本地设备计算资源F^k_{loc}/GHz	1
MEC服务器计算资源F_{mec}/GHz	[40, 60]
用户设备能耗系数ρ^k	10^{-27}
时延权重系数λ	0.5
折扣因子γ	0.99
本地线程数	4
Actor网络更新步长	0.01
Critic网络更新步长	0.001

3.2 仿真结果分析

基于A3C的任务卸载和资源分配联合优化算法的收敛情况如图4所示。可以看出使用的A3C在2 000多次训练后，就基本得到收敛结果。说明算法可以减少任务卸载和资源分配的决策时间，收敛程度良好，具有较好的可用性。

不同方案低时延用户数对任务计算成本的影

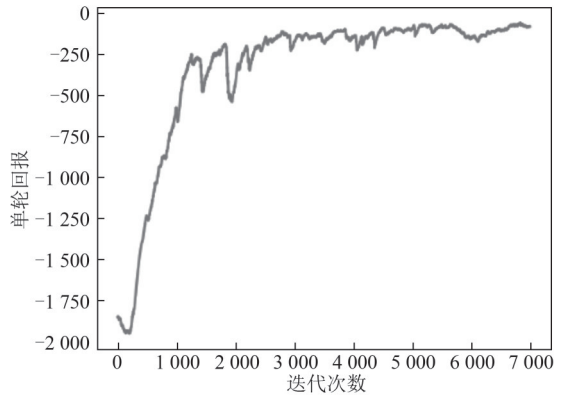

图4　基于A3C的任务卸载和资源分配联合优化算法的收敛情况

响如图5所示。由于低时延用户对时延要求较高，

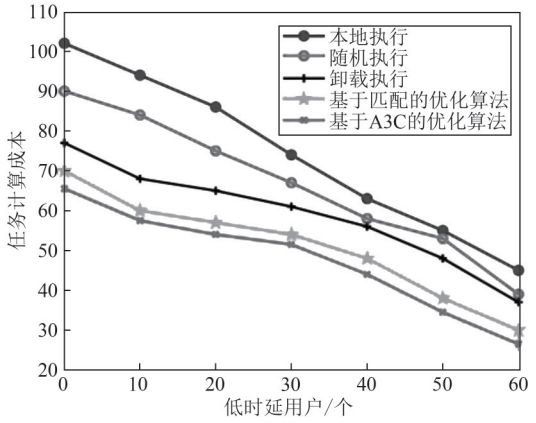

图5　不同方案低时延用户数对任务计算成本的影响

时延较低，而任务成本主要依赖于时延和能耗，因此任务成本会降低，所以5种算法的任务计算成本都随着低时延用户数目的增加而减少。本文提出的基于A3C、匹配的优化算法可以从本地和卸载计算中选择计算成本最低的方式进行任务计算，因此计算成本较低。卸载执行可以利用MEC的计算资源减少任务计算成本，但没有比较本地执行的情况，因此计算成本略高。随机执行采用随机的方式，本地执行没有利用MEC的计算资源，因此计算成本最高。

不同方案切片种类数对任务计算成本的影响如图6所示。切片种类越多，对资源的划分越细，从而提升对资源的利用效率，更能发挥MEC的作用。所以使用MEC进行计算的算法任务计算

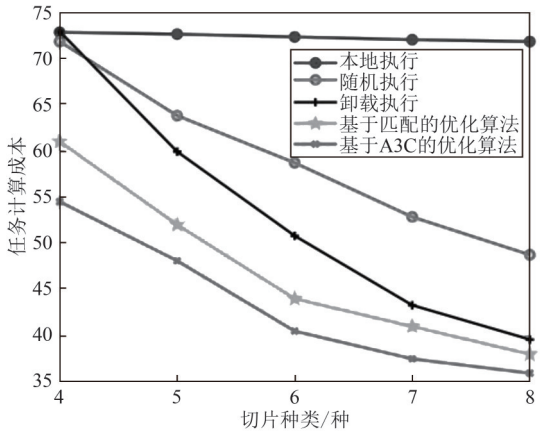

图6　不同方案切片种类数对任务计算成本的影响

成本都会随着切片种类增加而降低。本地执行仅使用本地设备计算，与切片种类无关，因此任务计算成本基本不变。

不同方案时延权重因子对任务计算成本的影响如图7所示。时延权重因子为0到1的小数，表示时延占任务计算成本的比重。值越大，时延占的比重越大，能耗比重越小。可以通过调整比重，实现不同的业务需求。由于本节参数设计中，时延成本比能耗更高，因此任务计算成本随时延权重因子的增大而增加。

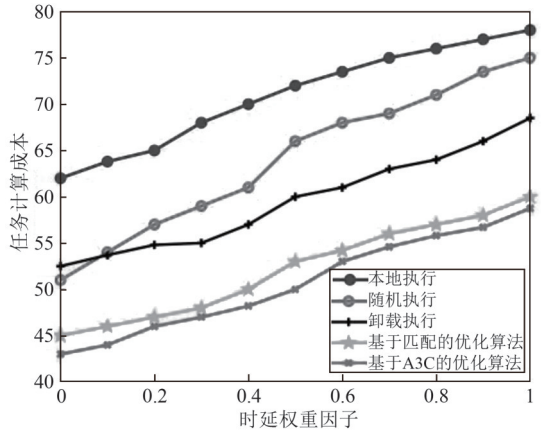

图7　不同方案时延权重因子对任务计算成本的影响

不同方案MEC计算能力对任务计算成本的影响如图8所示。当MEC计算能力低时，无法满足用户需求，计算成本会偏高。随着MEC计算能力的提升，可以提供更多的计算资源，减少任

务计算成本。本地执行仅使用本地设备计算，与MEC计算能力无关，因此任务计算成本基本不变。

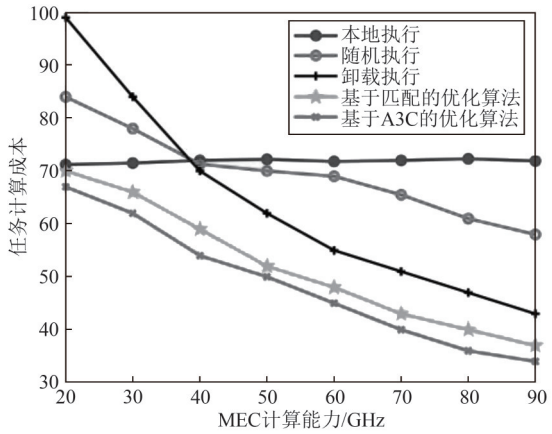

图8　不同方案MEC计算能力对任务计算成本的影响

4　结束语

本文介绍了基于A3C的任务卸载和资源分配联合算法，针对多用户多MEC的细粒度切片场景，考虑任务执行时延和功耗，并联合任务卸载和资源分配，建立最小化任务计算成本问题。由于引入切片维度，会增加任务决策时间，利用A3C强化学习算法，将MEC作为智能体，以异步并行的方式减少训练时间，完成快速决策。最后，通过仿真验证了算法可以在满足用户需求的前提下，降低任务计算成本。然而本文对于任务本身的考虑较为简单，仅考虑了完全卸载和本地卸载的简单情况，对于部分卸载的任务考虑较少。需要进一步对任务的划分比例、依赖关系等方面进行考虑。

参考文献：

[1] HASSEBO A. The road to 6G, vision, drivers, trends, and challenges[C]//Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway: IEEE Press, 2022: 1112-1116.

[2] KAMBLE P, SHAIKH A N. 6G wireless networks: vision, requirements, applications and challenges[C]//Proceedings of the 2022 5th International Conference on Advances in Science and Technology (ICAST). Piscataway: IEEE Press, 2022: 577-581.

[3] JASSIM MOHAMMED R, AL-SHAMMARI S W. Performance study of mobile edge computing[C]//Proceedings of the 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT). Piscataway: IEEE Press, 2022: 1-6.

[4] ZHANG Y, LI W, SEAH W K G. Admission control with latency considerations for 5G mobile edge computing[C]//Proceedings of the 2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). Piscataway: IEEE Press, 2023: 167-174.

[5] ARUN V, AZHAGIRI M. Design of long-term evolution based mobile edge computing systems to improve 5G systems[C]//Proceedings of the 2023 2nd International Conference on Edge Computing and Applications (ICECAA). Piscataway: IEEE Press, 2023: 160-165.

[6] GULERIA C, DAS K, SAHU A. A survey on mobile edge computing: efficient energy management system[C]//Proceedings of the 2021 Innovations in Energy Management and Renewable Resources (52042). Piscataway: IEEE Press, 2021: 1-4.

[7] MASOUDI M, CAVDAR C. Device vs edge computing for mobile services: delay-aware decision making to minimize power consumption[J]. IEEE Transactions on Mobile Computing, 2021, 20(12): 3324-3337.

[8] LIU J, MAO Y Y, ZHANG J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems[C]//Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT). Piscataway: IEEE Press, 2016: 1451-1455.

[9] ZHANG P, YANG J, FAN R F. Energy-efficient mobile edge computation offloading with multiple base stations[C]//Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). Piscataway: IEEE Press, 2019: 255-259.

[10] CHEN L X, ZHOU S, XU J. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1619-1632.

[11] WU B W, ZENG J, GE L, et al. A game-theoretical approach for energy-efficient resource allocation in MEC network[C]//Proceedings of the ICC 2019 - 2019 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2019: 1-6.

[12] HU S H, LI G H. Dynamic request scheduling optimization in mobile edge computing for IoT applications[J]. IEEE Internet

of Things Journal, 2020, 7(2): 1426-1437.
[13] YANG Y J, CHEN X, CHEN Y, et al. Green-oriented offloading and resource allocation by reinforcement learning in MEC [C]//Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT). Piscataway: IEEE Press, 2019: 378-382.
[14] LIU Y, YANG C, JIANG L, et al. Intelligent edge computing for IoT-based energy management in smart cities[J]. IEEE Network, 2019, 33(2): 111-117.
[15] ZHANG J, XIA W W, YAN F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing[J]. IEEE Access, 2018(6): 19324-19337.

[作者简介]

王晔（1987– ），男，博士，江苏移动信息系统集成有限公司技术总监、工程师，主要研究方向为移动蜂窝技术、车路协同、算力网络、工业互联网。

王逸飞（2000– ），男，南京邮电大学硕士生，主要研究方向为网络切片、资源调度等。

陈康（1998– ），男，南京邮电大学硕士生，主要研究方向为无线通信、接入网切片技术等。

朱晓荣（1977– ），女，南京邮电大学教授、博士生导师，主要研究方向为 5G/6G 通信系统、物联网、区块链等。

童恩（1971– ），男，博士，中国移动通信集团江苏有限公司首席专家、正高级工程师，主要研究方向为移动蜂窝技术、物联网、车联网、大数据等。

徐语菲（1997– ），女，现就职于江苏移动信息系统集成有限公司，主要研究方向为工业互联网。