DOI:10.19652/j.cnki.femt.2305394

异构网络中基于MADDPG的协作边缘缓存策略研究*

宋端正[1] 李 晖[2] 诸锦涛[1] 王 昊[1]
(1.南京信息工程大学电子与信息工程学院 南京 210044;
2.无锡学院江苏省集成电路可靠性技术及检测系统工程研究中心 无锡 214105)

摘 要:由于大量用户和设备共存,移动网络经历了数据量和用户密度的巨大增长。在宏基站(macro base station,MBS)覆盖区域内部署小蜂窝基站(small basic station,SBS),并提前在SBS缓存热门内容,是下一代移动通信网络提供高速、低时延服务的有效手段。针对异构网络环境不稳定以及难以找到精确的数学模型进行优化的问题,提出一种基于传输时延最小的异构网络协作边缘缓存算法。首先以Markov移动预测模型为基础,考虑用户社交关系对于用户移动性的影响,给出了新的用户移动位置预测方法;其次,采用多智能体深度确定性策略梯度(multi-agent deep deterministic policy gradient,MADDPG)算法,通过用户关联、延迟控制和缓存设计来减少内容传输时延并提高缓存命中率。仿真结果表明,同传统DDPG和Greedy算法相比,MADDPG算法缓存命中率分别提高17.89%和42.71%,内容传输时延分别降低9.07%和12.86%,能够有效地解决异构网络中的资源分配和缓存设计问题。

关键词:异构网络;边缘缓存;资源分配;深度强化学习
中图分类号:TN929.5 **文献标识码**:A **国家标准学科分类代码**:510

Strategy of MADDPG-based collaborative edge caching in heterogeneous networks

Song Duanzheng[1] Li Hui[2] Zhu Jintao[1] Wang Hao[1]
(1. School of Electronic and Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China; 2. Jiangsu Province Engineering Research Center of Integrated Circuit Reliability Technology and Testing System, Wuxi University, Wuxi 214105, China)

Abstract: Mobile networks have experienced a tremendous growth in data volume and user density due to the coexistence of a large number of users and devices. Deploying small basic station (SBS) within the coverage area of macro base station (MBS) and caching popular content in SBS in advance is an effective means to provide high-speed and low-latency services in next-generation mobile communication networks. Aiming at the unstable heterogeneous network environment and the difficulty of finding an accurate mathematical model for optimisation, a collaborative edge caching algorithm for heterogeneous networks based on transmission delay minimization is proposed. Firstly, a new user mobile location prediction method is given based on Markovian mobile prediction model, considering the influence of user social relations on user mobility. Secondly, multi-agent deep deterministic policy gradient (MADDPG) algorithm is used to reduce the content delivery delay and improve the cache hit rate by user association, delay control and cache design. Simulation results show that compared with traditional DDPG and greedy algorithms, the cache hit rate is improved by 17.89% and 42.71%, and the content delivery delay is reduced by 9.07% and 12.86%, respectively, which can effectively solve the resource allocation and cache design problems in heterogeneous networks.

Keywords: heterogeneous networks; edge caching; resource allocation; deep reinforcement learning

收稿日期:2023-08-04
*基金项目:国家自然科学基金(61661018)、江苏省基础研究计划青年基金(BK20210064)、江苏省双创博士人才项目(JSSCBS20210863)、南京信息工程大学滨江学院科研启动项目(2021r006)资助

0 引 言

如今，第五代（5th generation，5G）网络的广泛商用已经成为现实，更好地支持增强移动宽带（enhanced mobile broadband，eMBB）业务、超可靠低时延（ultra-reliable and low latency communications，URLLC）关键应用和物联网（internet of things，IoT）背景下的大规模机器类型通信（massive machine type communication，mMTC）[1-2]。展望未来，第六代（6th generation，6G）网络预计将在 2030 年左右实现，届时，国际电信联盟（international telecommunication union，ITU）预计移动数据流量将超过每月 5 ZB，比 2010 年增长 670 倍[3]。与此同时，移动用户将增加 2 倍多，达到 171 亿，而 2010 年为 53.2 亿。

要想达到上述预期就需要采用新的无线网络设计方法和最近流行的机器学习（machine learning，ML）解决方案。基于 ML 的技术使通信网络能够利用各种移动应用中的丰富数据，并与它们的环境交互，以探索不同的动作，然后根据观察到的回报，它们适应并利用为其下一次冒险产生最高回报的动作。移动边缘计算（mobile edge computing，MEC）和缓存是推动向 6G 通信演进的一种技术，其中计算密集型任务发生在数据收集附近，热门内容离用户很近[4-5]。通过这种方式，避免了集中式的云计算，而回程和前端链接则从远程 Web 服务器获取不断的内容中解脱出来。此外，计算和通信延迟大大减少，便于提供低延迟应用程序。

同时，由于无线环境的动态性，涉及到大量的参数和约束，使得传统的非学习技术可能会失效，在线网络优化的复杂性令人望而却步。在这种情况下，ML 辅助的 MEC 和缓存可以利用过多的移动数据，并回答在何处、何时和缓存什么，以及哪些任务应该在边缘计算等问题[6-7]。由于在线 ML 辅助网络优化在 6G 演进中具有极其重要的意义，本文将重点研究异构网络中强化学习（reinforcement learning，RL）辅助边缘缓存技术。

人们对网络的资源分配进行了大量的研究。目前已有将信息中心网络（information centric network，ICN）技术应用于各种网络的有效内容分发方法[8-10]，但有效的内容缓存和分发方案仍值得研究。利用卫星广播的特性，文献[8]提出的方法测量用户的内容兴趣并更新通信节点上的缓存。然而，该方案只考虑了很少的影响内容传输过程的参数。文献[9]开发了一种匹配的基于游戏的内容放置策略。但其假定用户仅通过卫星服务，而不考虑与移动通信网络的合作。文献[10]设计了一个用于缓存文件传递的两层缓存模型。其特点是在地面上部署的站点分布稀疏，这不符合移动网络中的实际情况。所有的工作都试图探索可能的协同缓存和分配策略，然而，需要研究一种考虑不确定环境条件下的高效缓存和分配方案。

虽然很多文献都采用传统的方法对异构网络中的资源进行分配，但是在不稳定的环境下，传统的优化方法很难解决问题。一方面，异构网络的环境是不稳定的，用户对缓存文件的需求是不确定的。另一方面，在该场景的优化中引入了许多约束条件。有时很难找到精确的数学模型来解决这些优化问题。

针对上述问题，引入深度强化学习（deep reinforcement learning，DRL）进行系统的资源分配和缓存设计。DRL 是求解不确定条件下优化问题的一种有效方法。文献[11]使用深度 Q 网络（deep Q-network，DQN）来实现用户访问。文献[12]提出了一种协作的多 Agent 深度强化学习框架来实现无线资源管理策略。文献[13]提出一种基于决斗双深 Q 网络（dueling double deep Q network，D3QN）的功率分配算法来优化系统的传输速率。文献[14]讨论了集成的地球卫星网络，并使用 DRL 来实现吞吐量和带宽等资源优化问题。文献[15]使用多目标 DRL 处理认知卫星场景。

DRL 也被用于许多缓存设计优化问题中。文献[16]在边缘缓存场景中使用了“actor-critic”框架。文献[17]使用双层 Q 网络实现了一种称为双重编码缓存的方案。文献[18]将基站和用户缓存联合优化问题分解为两个子问题，然后应用价值函数逼近 Q－学习和 DQN 来求解这两个子问题。文献[19]提出了一种基于 DRL 的算法，该算法可以优化用户关联、NOMA 的功率分配、无人机（unmanned aerial vehicle，UAV）的部署和无人机的缓存放置，共同使内容分发时延最小化。

以上工作中针对异构网络场景的资源分配和缓存设计问题都是通过传统的 DRL 来解决的。传统的 DRL 算法是单 Agent 算法，无法处理场景中多 Agent 的不稳定环境[20]。当智能体数量增加时，不稳定和动态的环境会降低优化性能。目前，利用多 Agent 强化学习技术对综合异构网络资源分配和缓存设计进行研究的还很少。

本文提出了一种支持缓存的通用异构综合网络框架，该框架由 MBS 和 SBS 共同服务于网络中的用户。以最小化内容传输时延和缓存命中率之差为目标，提出了一个资源分配与内容缓存联合优化问题。为了解决提出的问题，首先考虑用户的移动性，提出了一种基于 Markov 移动预测模型的算法，结合社会关系预测用户的位置。其次，采用 MADDPG 算法来优化资源分配和缓存设计。仿真结果表明，所提算法有良好的收敛性，能有效解决资源分配与缓存设计问题。

1 系统模型

1.1 网络模型

如图 1 所示，本文系统的网络是由一个 MBS、多个 SBS 和多个 UE 组成的异构网络。在该网络中，K 个基站共同为网络用户提供服务。设 B 表示 BS 的集合，其中 $B=\{B_1, B_2, \cdots, B_K\}$。$M$ 个用户集由 $U=\{U_1, U_2, \cdots, U_m\}$ 表示。

在每个时隙内，M 个用户只能与本系统中的一个基

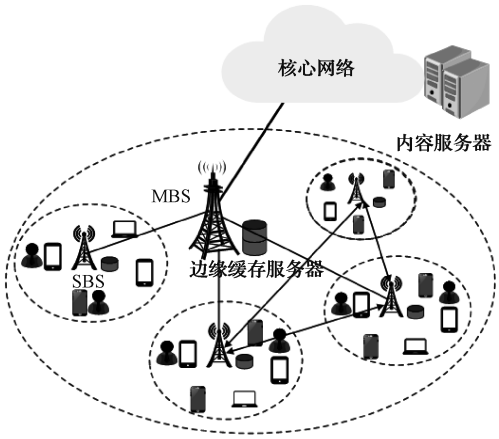

图1 系统模型

站进行关联。设 $a_m^k(t)$ 表示第 m 个用户与第 k 个 BS 的关联情况，当第 m 个用户与第 k 个 BS 关联时，$a_m^k(t)=1$，否则 $a_m^k(t)=0$。用户在一个时隙内只能与一个基站相关联。

在系统模型中，BS 在一个时隙 t 中服务的第 m 个用户的信干噪比（signal to interference plus noise ratio，SINR）可以表示为：

$$SINR_{Bm}(t)=\frac{a_m^k(t)\,|h_m^k(t)|^2 p_m(t)}{\sigma_{BI}(t)+\sigma_{BO}(t)+N_0} \tag{1}$$

式中：$|\cdot|^2$ 表示模的平方运算；$h_m^k(t)$ 是与第 k 个 BS 相关联的第 m 个用户之间的信道信息状态；$p_m(t)=a_m(t)\frac{p_{b\max}}{M_1}$，其中 $p_m(t)$ 为用户 m 的发射功率，$a_m(t)$ 为第 m 个用户的功率控制因子，$p_{b\max}$ 为基站最大发射功率，一个基站可以服务网络中的第 M_1 个用户；$\sigma_{BI}(t)$ 是来自同一基站的用户的干扰；$\sigma_{BO}(t)$ 是其他 BSs 中来自用户的干扰；N_0 是加性高斯白噪声（additive white Gaussian noise，AWGN）功率。相应的下行链路数据传输速率为：

$$R_{k,m}(t)=x_{k,m}(t)W_s\log_2(1+SINR_{Bm}(t)) \tag{2}$$

式中：$x_{k,m}(t)\in(0,1)$，表示 SBS_k 分配给关联 UEm 的带宽百分比；W_s 是 SBS 的带宽。

为了计算由同一簇内的用户引起的 $\sigma_{BI}(t)$，根据基站中的信道增益对用户进行排序 $|h_1^k(t)|\geqslant\cdots\geqslant|h_m^k(t)|\geqslant\cdots\geqslant|h_{M_1}^k(t)|$。

然后，根据信道增益的大小，$\sigma_{BI}(t)$ 是来自信道条件较好的用户的干扰。因此，来自同一簇的干扰可表示为：

$$\sigma_{BI}(t)=\sum_{i=1}^{m-1}a_i^k(t)\,|h_i^k(t)|^2 p_i(t) \tag{3}$$

而来自其他 BSS 服务的用户的干扰表示为：

$$\sigma_{BO}(t)=\sum_{q=1,q\neq k}^{K}\sum_{i=1}^{M_1}a_i^j(t)\,|h_i^k(t)|^2 p_i(t) \tag{4}$$

系统模型中的缓存设计描述如下：每个用户分别从文件库 $\boldsymbol{F}=\{1,2,3,\cdots,F\}$ 中请求文件。每个 BS 配置有缓存池来存储文件。因此，系统中有 K 个缓存池。BS 缓存池的大小设置为 $K_f<F$，每个 BS 从文件库 $\boldsymbol{F}$ 中选择 K_f 文件作为缓存文件的组合。每个 BS 可以存储 $K_f\times s$ 位文件。附属缓存池的大小设置为 $K_f<F$，每个附属缓存池从文件库 $\boldsymbol{F}$ 中选择 K_f 个文件作为缓存文件的组合。K_f 表示基站 K 缓存的文件数量，s 表示文件大小。

当用户请求到达时，系统首先在基站部署的本地缓存池中查找缓存文件。如果本地缓存池有用户需要的文件，则用户和本地 BS 之间的传输将发生。该文件从本地 BS 发送回用户，所消耗的功率为 $P_{m,r}(t)$。如果基站不能满足本地用户所需的缓存文件，用户将使用返回链路在核心网中查找所需的文件。此时消耗的功率为 $P_{c,r}(t)$，其中下标 m、c 和 r 分别表示用户、核心网和回程链路。

在系统中考虑缓存增益的方法有两种，一种是降低时延，另一种是降低功耗。这两个奖励取决于本地缓存设备是否满足用户的文件请求。

变量 $I_m(t)$ 可用于表示本地高速缓存设备是否满足第 m 个 BS 用户的文件请求。

$$I_m(t)=\begin{cases}1, & \text{满足要求}\\ 0, & \text{要求未得到满足}\end{cases} \tag{5}$$

假定文件的流行度分布遵循 ZipF 分布，受欢迎程度会影响缓存效果[21]。通常，受欢迎程度可以遵循一个广义的 ZipF 分布，产量估计在 0.56～0.83，有：

$$q_m=\frac{1/f^{\varepsilon}}{\sum_{f=1}^{F}1/f^{\varepsilon}},\quad \forall f \tag{6}$$

考虑到时间延迟的减少，缓存部署的回报为：

$$g_m(t)=I_m(t)\frac{count_m\times s}{T_m} \tag{7}$$

式中：T_m 是通过回程链路下载第 m 个用户请求的内容的时延；s 是文件的大小；$count_m$ 是第 m 个用户请求的内容的个数，这部分文件可以直接从本地缓存中获得。

Cache 命中率定义为系统中请求被满足的用户的比例，用来衡量 Cache 策略的性能。时隙 t 中的高速缓存命中率为：

$$Hit(t)=\sum_{m=1}^{M}I_m(t)/M \tag{8}$$

式中：M 为用户数量。

1.2 时延模型

在本文研究中假设每个用户每次只请求 1 个内容，该内容可以通过无线信道进行传输，并且，基站之间也能够采用协作缓存的方式缓存内容，假如当前基站没有所请求的内容，那么会首先到相邻的基站请求。假设用户请求的内容为 i，用 $x_{i,k}$ 表示基站 k 内容 i 的缓存状态，$x_{i,k}=0$ 表示内容 i 没有缓存在基站 k 上，$x_{i,k}=1$ 表示内容 i 缓存在基站 k 上，根据缓存状态分为如下两种情况。

1）内容 i 缓存在所请求的基站 k 上：在这种情况下，

用户可以直接从基站 k 获取内容 i，而无需请求距离很远的远端服务器，从而能够降低在回程链路上消耗的时延。因此，在这种情况下用户 u 从基站 k 请求内容 i 的传输时延为：

$$\tau_{1,u,i,k} = d_i / C_{u,k} \tag{9}$$

式中：d_i 表示内容 i 的大小；$C_{u,k}$ 表示用户 u 与基站 k 之间的信道容量。

2)内容 i 没有缓存在所请求的基站 k 上：在这种情况下，又可以分为如下两种情况。

(1)用户可以从所请求基站 k 的相邻基站获取内容 i，需要所请求基站 k 与相邻基站进行交互获取内容 i，因此，在这种情况下用户 u 从基站 k 请求内容 i 的传输时延为：

$$\tau_{2,u,i,k} = \frac{d_i}{C_{u,k}} + \frac{d_i}{C_{k',k}} \tag{10}$$

式中：$C_{k',k}$ 表示基站 k' 和基站 k 之间的信道容量。

(2)用户无法直接从所请求基站 k 的相邻基站得到内容 i，需要基站 k 和远端服务器利用回程链路实现信息交互获取内容 i，所以，在该模型中用户 u 从基站 k 请求内容 i 的传输时延为：

$$\tau_{3,u,i,k} = \frac{d_i}{C_{u,k}} + \frac{d_i}{C_{k',k}} + \frac{d_i}{r_i} \tag{11}$$

式中：r_i 表示分配给内容 i 的回传链路速率大小。

综合情况 1)和 2)可以发现，内容传输时延和缓存的状态有着非常紧密的联系，因此总传输时延为：

$$\tau_{u,i,k} = x_{i,k}\tau_{1,u,i,k} + (1-x_{i,k})\left[F\left(\sum_{k'\in K_k} x_{i,k'} \geqslant 1\right)\tau_{2,u,i,k} + F\left(\sum_{k'\in K_k} x_{i,k'} < 1\right)\tau_{3,u,i,k}\right] \tag{12}$$

式中：$x_{i,k}$ 表示基站 k 缓存内容 i 的状态，$x_{i,k}=1$ 表示基站 k 缓存内容 i，$x_{i,k}=0$ 表示基站 k 没有缓存内容 i；F 是指示函数，如果该函数中的式子成立，则该函数为 1，否则，该函数值为 0；K_k 表示基站 k 的相邻基站集合。

1.3 Markov 预测模型

假定在当前服务场景下具有 m 个不同位置，把位置 p 作为在 Markov 过程中的第 p 个状态 X_p，则在该场景下的状态空间可以定义为 $E=\{X_1, X_2, \cdots, X_p\}$ 以及在该场景下用户的移动模型可以定义为 $\{X, T\}$，其中，T 代表时间的序列。在此基础上，利用 Markov 预测方法，对每个节点的位置进行了预估，并提供了具体的建模和预测过程如下。

1)确定状态集合

对用户历史移动轨迹数据进行分析，将所有的历史访问地点集合用 L 表示。因为每个用户经过的所有地点不全都是重要的地点，只需要筛选出几个重要地点作为代表就可以了，所以挑选出访问频率最高的地点作为系统的状态集合 E，$E \subset L$。

2)分析得到一步转移概率矩阵

假设用户从位置 i 移动到位置 j 的总次数为 n 次，记为 $n_{i,j}$，那么在该模型中用户从位置 i 移动到位置 j 的概率为：

$$p_{i,j} = \frac{n_{u,i,j}}{\sum_{j\in E} n_{i,j}} \tag{13}$$

式中：$\sum_{j\in E} n_{i,j}$ 表示用户从位置 i 移动到所有其他位置的总次数。

同时，可以计算出转移概率矩阵，在该模型中用户的转移概率矩阵为：

$$\boldsymbol{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,m} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,m} \end{bmatrix} \tag{14}$$

3)分析得到 $l+1$ 时刻的绝对分布

假设初始时刻记为 l，用户处于状态 X_j 的概率记为 $p_j^{(l)}$，通过获取该时刻每个状态的概率便能得到用户的初始分布，在该模型中用户的初始分布为：

$$P(l) = \{p_1^{(l)}, p_2^{(l)}, \cdots, p_m^{(l)}\} \tag{15}$$

假设初始时刻为 l，初始状态取为 X_1，$m=5$，则可以得到初始分布 $P(l)=\{1,0,0,0,0\}$，则 $l+1$ 时刻的绝对分布为：

$$P(l+1) = P(l)\times P = \{p_1^{(l+1)}, p_2^{(l+1)}, p_3^{(l+1)}, p_4^{(l+1)}, p_5^{(l+1)}\} \tag{16}$$

取 $l+1$ 时刻绝对分布的最大值，最终可以得到 $t+1$ 时刻系统的状态 $X_j = \text{argmax}\{p_j^{(l+1)}\}$。

1.4 问题描述

本文研究了在异构网络服务场景下的边缘缓存放置策略，旨在满足每个基站缓存容量限制的前提下，选择合适的内容缓存在各个基站中，从而使得所有用户请求内容的总时延最小，并且保证每个小小区的缓存命中率最大。该问题被建模为最小化每个小小区中内容交付时延与缓存命中率之差，其数学表达式为：

$$\begin{aligned} &\min \sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K}\tau_{u,i,k} - Hit(t) \\ &\text{s.t.} \begin{cases} \sum_{i=1}^{I}(d_i \times x_{i,k}) \leqslant C_k (k = 1,2,\cdots,n) \\ x_{i,k} = 0 \text{ 或 } 1 \end{cases} \end{aligned} \tag{17}$$

在约束条件中，第 1 个约束是缓存空间限制，即每个基站的缓存总量不得超过其最大容量 $C_k(k=1,2,\cdots,n)$，其中 d_i 是内容 i 的大小，$x_{i,k}$ 是内容 i 在基站 k 的放置决策。第 2 个约束是关于内容放置的，即内容 i 在基站 k 只有两种可能，内容 i 在基站 k 只能完全缓存或完全不缓存，不允许部分缓存的情况，$x_{i,k}=0$ 表示基站 k 不缓存内容 i，$x_{i,k}=1$ 表示基站 k 缓存内容 i。

2 用户移动性预测

为了进一步提高用户移动性预测的精准度，本文将用户的社会关系引入到基于 Markov 预测的用户移动模型

中，利用用户的社会关系因素进一步优化对位置的预测，基于社会关系的用户移动性预测算法的流程如下。

步骤 1)收集用户历史数据。

步骤 2)获取基于 Markov 模型的预测结果。

步骤 3)划分用户社区。首先通过式(18)计算用户之间的相似度，然后采用 k-means 聚类算法将用户分组，使得同一社区内的用户具有较强的联系和较高的互动频率。相似度的计算公式如下：

$$usersim_{u,u'} = \frac{\sum_{i=1}^{I} 2\times\left(1-\frac{1}{1+\exp(-|req_{u,i}-req_{u',i}|)'}\right)}{I} \tag{18}$$

式中：$req_{u,i}$ 表示用户 u 历史请求内容 i 的次数。

步骤 4)构造社会关系矩阵。首先需要计算用户之间的接触概率，反映他们的社会联系强度。接触概率与接触时间有关，即用户之间的接触时间越长，说明他们之间的亲密关系越高。假设用户 u 和用户 u'在 T 的时间内接触的记录为 $J=\{(t_k^s,t_k^e)\}$，$k=1,2,\cdots,n$，其中 n 代表用户 u 和用户 u'在时间 T 内的接触次数，t_k^s 代表用户 u 和用户 u'在第 k 次接触的起始时间，t_k^e 代表用户 u 和用户 u'在第 k 次接触的结束时间，用户 u 和用户 u'在该模型中的接触概率为：

$$p(u,u')=\sum_{k=1}^{n}(t_k^e-t_k^S)/T \tag{19}$$

基于上述公式，可以计算出社会关系矩阵，该模型中用户之间的社会关系如下：

$$\boldsymbol{S}=\begin{bmatrix} 1 & \cdots & p(1,u) \\ \vdots & \ddots & \vdots \\ p(u,1) & \cdots & 1 \end{bmatrix} \tag{20}$$

步骤 5)获取预测的结果。假设用户 u 所处的社区为 C，并定义当前位置为 i 的该社区用户集合为 $Q=\{Q_{u1},\cdots,Q_{uj},\cdots,Q_{un}\}$，其中 $Q\subseteq C$。然后计算在给定当前位置为 i 的其他社区用户的情况下，用户 u 下一时刻到达位置 i 的条件概率，该模型中的条件概率为：

$$p_i(u\mid Q_{u_j})=p_i(u,Q_{u_j})/p_i(Q_{u_j}) \quad j=1,2,\cdots,n \tag{21}$$

式中：$p_i(Q_{u_j})$ 代表用户 u_j 在下一时刻还停留在位置 i 的概率，它可以通过之前获取的 Markov 预测模型得到；$p_i(u,Q_{u_j})$ 代表用户 u 和用户 j 在位置 i 碰巧遇到的概率。

$$p_i(u,Q_{u_j})=\frac{y_i(u,Q_{u_j})}{\sum_{i=1}^{m}y_i(u,Q_{u_j})} \quad j=1,2,\cdots,n \tag{22}$$

式中：$y_i(u,Q_{u_j})$ 代表用户 u 和用户 j 在位置 i 相遇的次数；$\sum_{i=1}^{m}y_i(u,Q_{u_j})$ 代表用户 u 和用户 j 在所有位置相遇的次数总和。则有：

$$\begin{aligned} p_i(u)&=\sum_{j=1}^{n}\gamma_j p_i(u\mid Q_{u_j}) \\ \gamma_j&=\frac{p(u,u_j)}{\sum_{j=1}^{n}p(u,u_j)} \end{aligned} \tag{23}$$

式中：γ_j 代表每一个条件概率的权值，且 $\sum_{i=1}^{n}y_i=1$。

步骤 6)终止。

用户移动性预测算法伪代码如算法 1 所示。

算法 1 基于社会关系的用户移动性预测算法

输入：用户历史移动数据，用户历史请求数据，状态空间大小 m 以及当前时刻 l；

输出：预测的位置数组 *predictedPosition*

1. 初始化一个二维数组 *UserHistoryMove*，用于存储用户历史移动数据，其中 u 为用户个数。
2. 调用 Markov 预测函数
3. 计算一步转移概率存放在 *transProbablityMatrix* 二维数组中。
4. 调用聚类算法函数 Cluster (*UserHistoryRequest*)，返回一个一维数组 *Community*，表示每个用户所属的社区编号。
5. 初始化一个二维数组 *socialMatrix*，用于存储社会关系矩阵。
6. **for** $i=0,1,\cdots,u-1$ **do**
7. **for** $j=i+1,i+2,\cdots,u-1$ **do**
8. 调用接触概率函数 ContactProbability(i, j)，返回一个数值 s，表示用户 i 和 j 之间的接触概率。
9. socialMatrix[i][j] = socialMatrix[j][i] = s；
10. **end for**
11. **end for**
12. 初始化一个二维数组 *NextPosition*，用于存储预测的结果。
13. **for** $j=0,1,\cdots,u-1$ **do**
14. **for** $i=0,1,\cdots,m-1$ **do**
15. 计算用户 j 到达位置 i 的概率 $p_i(j)$；
16. *NextPosition*[j][i] = $p_i(j)$；
17. **end for**
18. **end for**
19. **return** *predictedPosition*

3 基于多 Agent DRL 的异构网络资源分配和缓存设计

为了最大限度地提高目标函数，并将 MADDPG 框架引入到综合异构通信网中。优化过程包括两个部分，用户关联和延迟控制，然后是缓存设计；给出了一种基于 MADDPG 的算法来解决这两个问题。

3.1 RL 模型

RL 不需要一个数据集在每个回合从环境中接收奖励信息，学习然后更新模型参数。RL 中的 Agents 可以与环境交互并观察行为的奖赏，然后学习如何改变自己的行为以获得更高的奖赏。代理以试错的方式不断取得进展。

在这种异构通信网络场景中，环境中存在着多个 Agent。当 Agent 数量不断增加时，传统的单 Agent 强化学习会遇到一个不稳定的动态环境，这会导致 Agent 过度适应竞争者的强策略。本文提出了 MADDPG 算法来处理复杂的多 Agent 场景，能够更好地适应多 Agent 之间的交互，获得更好的优化效果。

异构网络的延迟优化问题可以用 Markov 决策过程(Markov decision process，MDP)来建模。MDP 由状态空间 S、动作空间 A、奖励空间和转移概率空间构成。作为一个 Agent，每个用户可以观察环境并得到观察结果，然后从动作空间中选择并执行动作。然后，它将在执行动作后获得奖励。本文算法中的 Agent、动作、状态和奖励如下定义。

1)Agent，网络中的每个 BS 都被视为一个 Agent。

2)状态，将网络中缓存的内容视为当前状态 $S(t)$。为了聚焦具有高流行度的内容，基于预测内容的内容流行度，将状态空间 $S(t)$的内容按降序排序，从而可以将当前状态表示为 $S(t)=(S_1,S_2,\cdots,S_C)$，其中 S_i 是最流行的内容。

3)动作，在内容放置阶段，智能体需要决定是否以及在哪缓存 UE 请求的内容；在内容交付阶段，智能体需要决定如何分配带宽。动作的定义为：

$$a_k(t)=\{\boldsymbol{y}_k(t),\boldsymbol{x}_k(t)\}=\{y_{k,1}(t),\cdots,y_{k,J_k}(t),x_{k,1}(t),\cdots,x_{k,J_k}(t)\} \tag{24}$$

式中：$\boldsymbol{x}_k(t)$ 表示智能体决定是否、以及在哪缓存 UE 请求内容；$\boldsymbol{y}_k(t)$ 表示智能体决策带宽分配。

4)奖励，奖励函数是用来评估智能体的决策效果，它与问题建模的优化目标相关。每个 SBS 作为一个智能体，通过与环境交互来训练网络模型，以达到最大的奖励值。然而，本文的优化目标是一个最小化问题，所以奖励函数与优化目标是负相关的。因此，奖励函数定义为：

$$r_k(t)=\begin{cases}\omega_t\times\dfrac{c(t)-\tau_{u,i,k}}{c(t)}+\omega_h\times Hit(t), & \\ \quad \tau_{u,i,k}<c(t) & \\ 0, \quad \text{其他} & \end{cases} \tag{25}$$

式中：$c(t)$表示从核心网到小小区 k 中所有 UE 的内容交付总时延。

$$c(t)=\sum_{j\in J_k}\left(\frac{L}{\frac{W_s}{J_k}\log_2(1+SINR_{Bm}(t))}+T_{mbs}+T_{core}\right) \tag{26}$$

其中，$\dfrac{c(t)-\tau_{u,i,k}}{c(t)}$ 是归一化小小区 k 中所有 UE 的内容交付时延。此处奖励函数的归一化可以加快训练速度，提高训练效果。ω_t 和 ω_h 是内容交付时延和缓存命中率的权重，$\omega_t+\omega_h=1$。

MADDPG 算法可以获取其他 Agent 执行的动作，减少不稳定性。转移概率为：

$$P(s'\mid s,a_1,\cdots,a_N,\chi_1,\cdots,\chi_K)=P(s'\mid s,a_1,\cdots,a_N)=P(s'\mid s,a_1,\cdots,a_N,\chi'_1,\cdots,\chi'_K) \tag{27}$$

由式(27)中的状态转移概率可知，当代理的策略被动态地改变和更新时，环境仍然是稳定的。其中，$a_i,\forall i\in[1,K]$是代理的动作，s 是状态。网络中有 K 个代理，系统中所有 K 个代理的策略值设置为 $\chi=\{\chi_1,\cdots,\chi_K\}$。每个代理的策略都有其对应的参数值 $\overline{w}=\{\overline{w}_1,\cdots,\overline{w}_K\}$)。

提出了 MADDPG 算法，用于解决异构通信网络的优化问题。在 MADDPG 算法中，每个 Agent 通过优化策略来获得最大的收益。目标函数的梯度可以通过如下方程求解：

$$\nabla_{\omega_i}J(\chi_i)=E_{x,a\sim D}[\nabla_{\omega_i}\chi_i(a_i\mid o_i)\nabla_{a_i}Q_i^{\chi}(x,a_1,\cdots,a_N)] \tag{28}$$

式中：x、a 分别为 K 个智能体的状态空间和动作空间；D 为回放存储器；χ_i 为代理的策略值设置；Q_i^x 为算法中评估动作的 Q 函数。

演员网络和批评家网络在 MADDPG 算法中扮演不同的角色。演员网络将根据策略选择动作。根据策略值选择动作 A_1 或 A_2，动作空间是连续的。批评家网络评估将要执行的行动。评估动作的方法是更新 Q 函数，如式(28)所示，计算动作的 Q 函数是 $Q_i^x(x,a_1,a_2,\cdots,a_n)$。在 MADDPG 算法中，演员和批评家的网络更新采用了不同的方法。参与者网络通过式(26)中的梯度下降更新用于选择行动的策略网络[22]。批评家网络更新评价演员网络选择的动作的 Q 函数，以最小化以下损失函数 $L(\omega_i)$。

$$y=r_i+rQ_i^{u'}(x',a'_1,\cdots,a'_N)\mid_{a'_j=u'_j(o_j)} \tag{29}$$

3.2 算法描述

算法 2 系统 Cache 设计问题的 MADDPG 算法，用于异构网络资源分配。

算法 2 异构网络资源分配问题的 MADDPG 算法

1. 输入：深度神经网络的参数和重放记忆；
2. **for** episode = 1 to Ep **do**
3. 初始化异构网络的观测，包括用户关联和延迟控制；
4. **for** agent = 1 to N **do**
5. **for** step = 1 to S_t **do**
6. 每个 BS 得到观测状态 $S(t)=(S_1,S_2,\cdots,S_C)$；
7. 每个 BS 从 $a(t)=\{a_1(t),\cdots,a_K(t)\}$中选择用户关联和延迟控制；
8. 每个 BS 通过式(22)观测奖励；
9. **end for**
10. **end for**

11. 从重播内存中随机取样一批；
12. 每个代理更新演员网络和批评家网络；
13. 更新目标网络的参数；
14. **end for**
15. 输出：训练后的深度神经网络参数，用户关联和延迟控制。

MADDPG 算法流程如图 2 所示。在决策时刻 t，每个 Actor 在观察全局状态 s_t 后，采取行动 $a_{t,u}$。然后，每个 Actor 在实行动作 $a_t=(a_{t,1},a_{t,2},\cdots,a_{t,u})$后获得个人奖励 $r_u(s_t,a_t)$。在处理所有 Actor 的单个奖励后，获得全局奖励 $r(s_t,a_t)$，并将其与 s_t、a_t 以及新的全局状态 s_{t+1} 一起存储在 Replay memory 中。然后，Critic 从 Replay memory 中提取一批样本来评估每个 Actor 的策略。Actor 根据 Critic 的评价更新他们的策略。

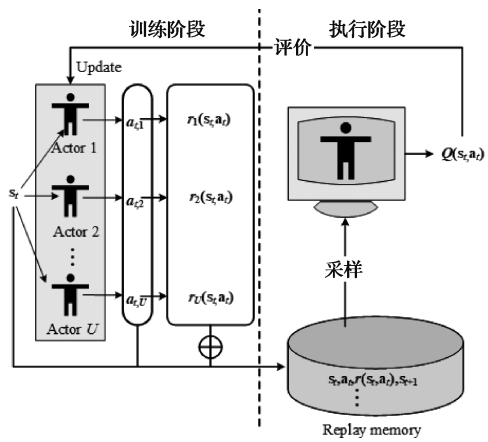

图 2 MADDPG 算法流程

4 实验结果分析

4.1 实验设置

本文所提深度强化学习算法通过 Python 平台的 Pytorch 框架实现，仿真结果表明了 MADDPG 框架的收敛性能，并与传统的深度强化学习算法进行了比较。

MBS 覆盖范围内设置 3 个小小区，每个小小区的半径设置为 150 m，每个小小区中的 UE 随机分布在其覆盖区域内。用户与基站间的信道增益遵循自由空间路径损耗模型，表示为：

$$h_{u,n}(t)=A_d\left(\frac{3\times10^8}{4\pi f_c d_{u,n}(t)}\right)^2 \tag{30}$$

其中，天线增益 $A_d=4.11$，载波频率为 $f_c=900$ MHz，$d_{u,n}(t)$为用户设备与基站间的距离。

假设 UE 周期性生成内容请求服务，且遵循 Zipf 分布。缓存相关仿真环境如下，文件库的大小 F 为 400，内容大小统一为 1 Mb，BS 缓存设备的大小 N_f 为 3。用户所需的文件数 $count_m$ 设置为 1，文件内容大小 s 设置为 2 bit。为便于参考，表 1 为其他仿真参数。

表 1 仿真参数

参数	值
W_s	10 MHz
P_s	1 W
N_0	−174 dBm/Hz
K	10
C_m	200 Mb
C_s	100 Mb
T_{sbs}	10 ms
T_{mbs}	10 ms

优化器为 AdamOptimizer，激活函数为 ReLU。神经网络的学习速率为 $alr=clr=0.001$。折扣因子为 0.95。批大小设置为 10。实验的总回合数设定为 1 000。在每一回合里，代理需要完成 100 个步骤。

4.2 仿真结果与分析

4 种不同算法的收敛过程如图 3 所示。从图 3 可以看出，在 40 次迭代后，所有算法都达到了收敛，只有微小的波动。可以发现，当回合数较少时，奖励值变化较大。这是因为系统在开始阶段还在学习和优化网络，所以一些探索性的行为会导致性能下降。随着回合数的增长，奖励值逐渐趋于稳定和平滑。这表明此时系统已经逐渐找到了最优策略，神经网络的训练也接近完毕。从图 3 还可以看出，基于 MADDPG 算法的模型比基于 DDPG 算法的模型奖励值更高。而且，这两种强化学习模型的性能都明显优于 Greedy 和 Random 算法。

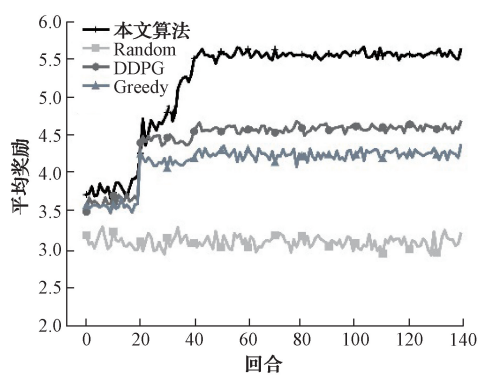

图 3 算法收敛性

在 BS 的缓存容量固定为 100 的情况下，本文算法每回合的缓存命中率和内容传输延迟变化如图 4 所示。随着回合数的增长，前 10 回合内缓存命中率逐步提高，内容传输延迟逐步降低。这是因为本地 BS 和相邻 BS 在前 10 回合内逐步缓存了更适合的热门内容。另外，可以看到缓存命中率和内容传输延迟在第 10 回合左右达到了稳定。这是因为本地 BS 在大约 10 回合内就能够学习到最优的协同缓存策略。

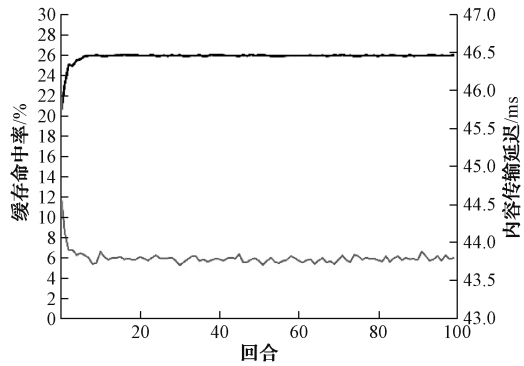

图 4　算法缓存命中率和传输时延收敛性

各算法在不同缓存容量下的缓存命中率比较如图 5 所示。从图 5 可以看出，随着缓存容量的增加，所有方案的缓存命中率都有所提高。这是因为本地 BS 能够用更大的空间缓存更多的内容，所以用户的请求内容更容易从本地 BS 中获取。另外，Random 方案的缓存命中率最低，因为该方案只是随机地选择内容，没有考虑到内容的流行度。另外，本文算法和 DDPG 算法的性能都明显优于 Random 和 Greedy 算法。这是因为 Random 和 Greedy 算法不能通过学习来预测哪些内容应该被缓存，而本文算法和 DDPG 则能够根据历史请求数据来决定缓存策略。从图 5 还得出，基于 MADDPG 的本文算法在缓存命中率方面，分别平均比 DDPG 和 Greedy 算法提高 17.89%和 42.71%。

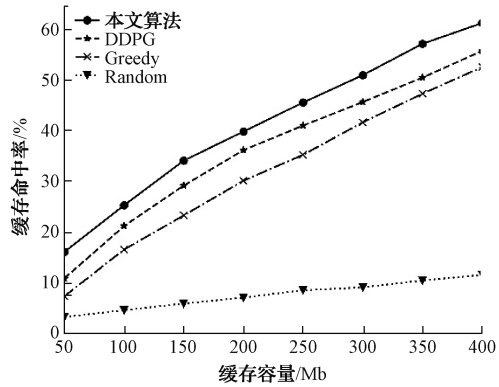

图 5　不同算法缓存命中率

每个 BS 在不同缓存容量下的内容传输延迟比较如图 6 所示。从图 6 可以看出，随着缓存容量的提高，所有方案的内容传输延迟都有所降低。这是因为每个 BS 能够用更大的空间缓存更多的内容，所以每个载体从本地 BS 和相邻 BS 获取内容的概率更高，从而降低了内容传输延迟。另外，本文算法的内容传输延迟比其他方案更低，分别平均比 DDPG 和 Greedy 算法降低 9.07%和 12.86%。这是因为本文算法的缓存命中率更高，所以更多的用户可以直接从本地 BS 中获得内容，从而减少了内容传输的时间。

在每个 BS 的缓存容量为 100 的情况下，不同选择用户数对 MADDPG 方案的缓存命中率和内容传输延迟的影响如图 7 所示。从图 7 可以看出，随着选择用户数的增多，缓存命中率也随之提高。这是因为选择更多的用户参与训练可以提供更丰富的数据和计算资源，从而提高 MADDPG 方案的预测精度。另外，内容传输延迟也随着选择用户数的增多而降低，这是因为缓存命中率的提高使得更多的用户可以直接从本地 BS 中获得内容，从而减少了内容传输的时间。

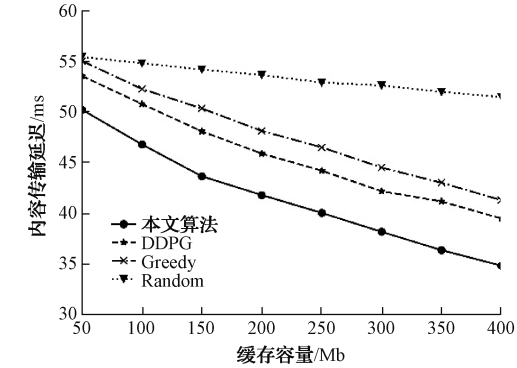

图 6　不同算法内容传输时延

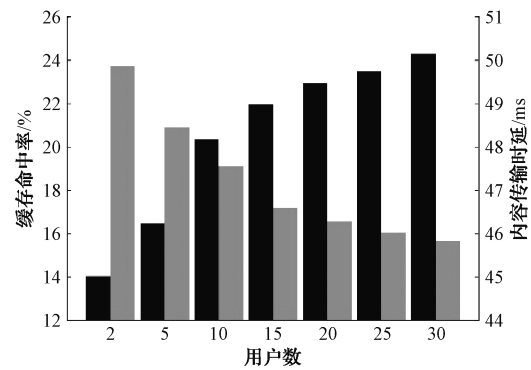

图 7　不同用户数影响

在每个 BS 的不同缓存容量下，本文算法和本文算法(无移动预测)的缓存命中率对比如图 8 所示。从图 8 可以看出，本文算法的缓存命中率高于本文算法(无移动预测)。这是因为本文算法能够根据预测的热门内容来决定最佳的协同缓存，从而能够在本地 BS 中缓存更适合的热门内容。

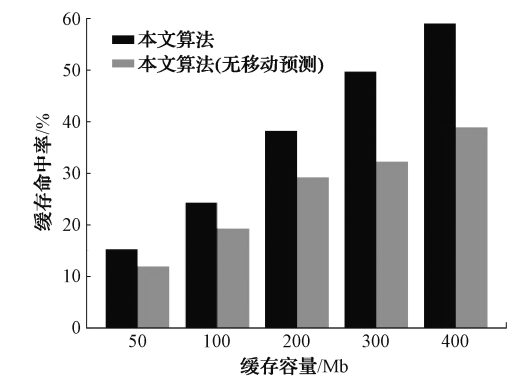

图 8　无移动预测对缓存命中率的影响

在每个 BS 的不同缓存容量下，本文算法和本文算法(无移动预测)的内容传输延迟对比如图 9 所示。从图 9 可以看出，本文算法的内容传输延迟低于本文算法(无移动预测)。这是因为本文算法的缓存命中率高于本文算法(无移动预测)，所以更多的用户可以直接从本地 BS 中获得内容。

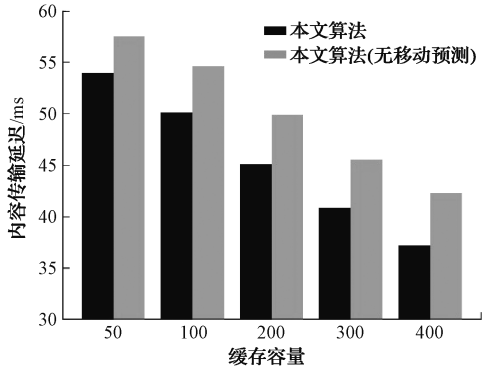

图 9　无移动预测对传输延迟的影响

5　结　论

本文利用移动性的计算、缓存、通信和控制来智能地缓存边缘网络中的内容。首先，值得注意的是，在使用 Markov 预测模型的情况下，考虑了一种基于社会关系的用户移动性预测算法。其次，提出了一种基于多 Agent 深度强化学习的异构网络资源分配和缓存设计方案。目标是最小化总传输时延并提高缓存命中率。采用 MADDPG 算法来实现用户关联、延迟控制和缓存设计，以降低系统的总传输时延。仿真结果表明，所提出的框架在解决问题方面具有良好的有效性和潜力。与传统的单智能体深度强化学习算法 DDPG 等基准算法相比，具有更好的优化性能。异构网络架构中 BS 的灵活部署存在局限性。下一步的研究中，不局限于异构网络，可以拓宽网络场景，如无人机或车载网络等，采用 MADDPG 算法研究基于 MEC 的多层异构网络环境下的最优资源分配问题。

参 考 文 献

[1] ALI R, ZIKRIA Y B, BASHIR A K, et al. URLLC for 5G and beyond: Require-ments, enabling incumbent technologies and network intelligence [J]. IEEE Access, 2021(9): 67064-67095.

[2] AL-RUBAYE S, RODRIGUEZ J, FRAGONARA L Z, et al. Unleash narrowband technologies for industrial internet of things services [J]. IEEE Network, 2019, 33(4): 16-22.

[3] 郑冰原，孙彦赞，吴雅婷，等. 基于深度强化学习的超密集网络资源分配[J]. 电子测量技术，2020，43(9)：133-138.

[4] 王昊，李晖，宋端正，等. 面向云-雾计算系统中的遗传算法任务调度研究[J]. 电子测量与仪器学报，2023，37(8)：40-51.

[5] LIU D, CHEN B, YANG C, et al. Caching at the wireless edge: Design aspects, challenges, and future directions [J]. IEEE Communications Magazine, 2016, 54 (9): 22-28.

[6] ZHANG C, PATRAS P, HADDADI H. Deep learning in mobile and wireless networking: A survey[J]. IEEE Communications Surveys & Tutorials, 2019, 21 (3): 2224-2287.

[7] LUO F L. Deep multi-agent reinforcement learning for cooperative edge caching [J]. IEEE Machine Learning for Future Wireless Communications, 2020, DOI: 10.1002/9781119562306.ch21.

[8] GALLUCCIO L, MORABITO G, PALAZZO S. Caching in information-centric satellite networks[C]. 2012 IEEE International Conference on Communications (ICC), 2012: 3306-3310.

[9] LIU S, HU X, CUI G, et al. Distributed caching based on matching game in LEO satellite constellation networks[J]. IEEE Communications Letters, 2017, 22(2): 300-303.

[10] WU H, LI J, LU H, et al. A two-layer caching model for content delivery services in satellite-terrestrial networks [C]. 2016 IEEE Global Communications Conference (GLOBECOM), 2016: 1-6.

[11] CAO Y, LIEN S Y, LIANG Y C. Deep reinforcement learning for multi-user access control in non-terrestrial networks[J]. IEEE Transactions on Communications, 2021, 69(3): 1605-1619.

[12] LIAO X, HU X, LIU Z, et al. Distributed Intelligence: A verification for multi-agent DRL based multibeam satellite resource allocation [J]. IEEE Communications Letters, 2020, 24(12): 2785-2789.

[13] 刘子怡，李君，李正权. 多用户蜂窝网络中基于深度强化学习的功率分配[J]. 国外电子测量，2023，42(3)：30-35.

[14] FERREIRA P V R, PAFFENROTH R, WYGLINSKI A M, et al. Reinforcement learning for satellite communications: From LEO to deep space operations[J]. IEEE Communications Magazine, 2019, 57(5): 70-75.

[15] FERREIRA P V R, PAFFENROTH R, WYGLINSKI A M, et al. Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles [J]. IEEE Journal on Selected Areas in Communications, 2018, 36(5): 1030-1041.

[16] ZHONG C, GURSOY M C, VELIPASALAR S.

Deep reinforcement learning-based edge caching in wireless networks [J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(1):48-61.

[17] ZHANG Z, CHEN H, HUA M, et al. Double coded caching in ultra dense networks: Caching and multicast scheduling via deep reinforcement learning[J]. IEEE Transactions on Communications, 2020, 68(2):1071-1086.

[18] QIAN Y, WANG R, WU J, et al. Reinforcement learning-based optimal computing and caching in mobile edge network[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(10):2343-2355.

[19] ZHANG T, WANG Z, LIU Y, et al. Joint resource, deployment and caching optimization for AR applications in dynamic UAV NOMA networks[J]. IEEE Transactions on Wireless Communications, 2021, 21(5):3409-3422.

[20] GAO Z, YANG L, DAI Y. Large-scale computation offloading using a multi-agent reinforcement learning in heterogeneous multi-access edge computing[J]. IEEE Transactions on Mobile Computing, 2022, 22(6):3425-3443.

[21] LIU X, ZHANG H, LONG K, et al. Energy efficient user association, resource allocation and caching deployment in fog radio access networks[J]. IEEE Transactions on Vehicular Technology, 2022, 71(2):1846-1856.

[22] ARULKUMARAN K, DEISENROTH M P, BRUNDAGE M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

作 者 简 介

宋端正，硕士研究生，主要研究方向为强化学习、无线通信网络等。

E-mail:20211249620@nuist.edu.cn

李晖(通信作者)，博士，教授，主要研究方向为空间信息网络、异构网络优化、AI 在 ICT 网络中应用等。

E-mail:hitlihui1112@163.com

诸锦涛，硕士研究生，主要研究方向为无线通信技术、网络优化等。

E-mail:1042807641@qq.com

王昊，硕士研究生，主要研究方向为无线通信系统、网络融合等。

E-mail:863447666@qq.com