小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2023-0206
2024 年 10 月 第 10 期
Vol.45 No.10 2024

卫星物联网中联合资源分配的边缘计算卸载策略

杨桂松[1],陶 挺[1],何杏宇[1,2],杜 平[3]

[1](上海理工大学 光电信息与计算机工程学院,上海 200093)
[2](上海理工大学 出版印刷与艺术设计学院,上海 200093)
[3](上海卫星互联网研究院有限公司/上海市卫星互联网重点实验室,上海 201210)
E-mail:sherri_he@163.com

摘 要: 卫星物联网通过引入边缘计算技术将计算能力下沉至靠近用户的边缘服务器上,使用户的服务质量得到了提升.然而计算资源有限的边缘服务器在面对大量的突发卸载任务时可能会出现过载的情况,导致任务的处理时延增加.本文将上述问题转化为马尔可夫决策过程下的最优策略问题,提出了一种基于 D³PG(Dueling Double Deterministic Policy Gradients)的联合资源分配的边缘计算卸载算法.该算法利用 Double Q-learning 思想和 Dueling 架构重新设计了 DDPG(Deep Deterministic Policy Gradients)算法中的价值网络,以提高计算卸载决策的准确性.仿真结果表明,与传统的算法相比,该算法能有效降低系统的平均时延和能量消耗,提高任务的完成率.

关 键 词: 卫星物联网;边缘计算;计算卸载;资源分配;深度强化学习
中图分类号: TP393 **文献标识码:** A **文 章 编 号:** 1000-1220(2024)10-2544-07

Strategy of Joint Resource Allocation and Computing Edge Offloading in Satellite Internet of Things

YANG Guisong[1],TAO Ting[1],HE Xingyu[1,2],DU Ping[3]

[1](School of Optical-Electrical & Computer Engineering,University of Shanghai for Science & Technology,Shanghai 200093,China)
[2](College of Communication & Art Design,University of Shanghai for Science & Technology,Shanghai 200093,China)
[3](Shanghai Institute of Satellite Internet Engineering Co.,Ltd./Shanghai Satellite Network Research Institute,Shanghai 201210,China)

Abstract: By introducing edge computing technology,the satellite Internet of Things has sunk its computing power to edge servers close to users,and improved the service quality of users. However,the edge servers with limited computing resources may be loaded when they have to process a large number of burst offloading tasks,and resulting in increased task processing delay. In this paper,the above problem is transformed into the optimal policy problem under the Markov decision process,and a joint resource allocation edge computing unloading algorithm based on D³PG(Dueling Double Deterministic Policy Gradients) is proposed. This algorithm utilizes the Double Q-learning concept and Dueling architecture to redesign the value network in the Deep Deterministic Policy Gradients algorithm,in order to improve the accuracy of computational offloading decisions. The simulation results show that the algorithm can effectively reduce the average delay and energy consumption of the system,and improve the task completion rate.

Keywords: satellite internet of thing;edge computing;computing offloading;resource allocation;deep reinforcement learning

0 引 言

卫星物联网(Satellite Internet of Thing,SIoT)[1]融合了低轨卫星网络与物联网的特征,作为一种服务大规模物联网设备互通的下一代网络.卫星物联网弥补传统地面物联网的缺陷与不足[2],将会在未来的物联网发展过程中起到至关重要的作用,产生巨大的经济价值.与传统的地面通信网络相比较,低轨卫星网络具有可移动性强、灾难恢复能力强、覆盖范围广阔、受地理因素限制小等突出优点,为实现全球无缝的网络覆盖提供了可能,可以为终端用户提供无处不在的服务[3-6].

然而,在传统非地面网络的透传模式下,低轨卫星充当云计算中心与物联网终端之间的中继节点,当上传到云计算中心的数据量过大时,可能会导致过大的传输延迟,难以满足 SIoT 用户对于新兴业务服务质量的需求.通过将边缘计算的思想引入了卫星物联网中,可以解决 SIoT 用户对更高服务质量的需求.在靠近终端用户的地面站上部署边缘服务器,使得计算任务可以就近在边缘服务器上处理,形成了全新的低轨卫星边缘计算场景.该方式可以有效地减少星地间频繁的链路通信,显著地降低终端用户的服务响应时延,终端用户的需求得到更好的满足.

收稿日期:2023-07-25 收修改稿日期:2023-08-29 基金项目:南通市科技局社会民生计划项目(MS12021060)资助;敏捷智能计算四川省重点实验室开放式基金项目资助;浦东新区科技发展基金产学研专项项目(PKX2021-D10)资助. 作者简介:**杨桂松**,男,1982 年生,博士,副教授,CCF 会员,研究方向为物联网与普适计算等;**陶 挺**,男,1995 年生,硕士,研究方向为卫星物联网;**何杏宇**,女,1984 年生,博士,副教授,CCF 会员,研究方向为物联网、群智计算大数据分析;**杜 平**,男,1978 年生,博士,研究方向为卫星互联网、5G 移动通信、计算机网络等.

近年来,关于卫星网络中的计算卸载和资源分配的研究一直是一个热门话题,其中包括对计算卸载技术的研究,以及如何在卫星网络中有效利用计算资源,以达到更高的效率. Zhang 等人[7]讨论了在卫星网络中通过利用移动边缘计算技术,设计了一种基于协作的资源分配和计算卸载策略,显著地降低用户的感知时延和系统的能量消耗. Wang 等人[8]提出了一种双边缘的计算卸载策略,通过优化任务卸载延迟以及系统能耗,从而提升任务执行效率. Qiu 等人[9]提出一个软件定义的方法来管理和协调网络中的计算资源和通信资源,并采取深度 Q 学习来解决联合通信和计算资源分配问题,以此来降低系统的平均响应时延. He 等人[10]提出了一种基于加权方法和排队论的计算卸载策略,通过建立终端用户与卸载设备之间的关系模型,综合考虑各种因素评分来选择最合适的计算卸载节点,从而降低系统的响应时延. Yang 等人[11]针对卫星物联网设备数量快速增长和计算需求激增的问题,综合考虑了终端设备、地面基站和低轨卫星间的多重影响因素,提出了一种基于博弈论的资源分配和计算卸载策略,实现了最大化用户服务质量,降低了系统的响应时延. 现有的大多数研究都是联合资源分配和计算卸载的最小化系统能耗和时延问题,然而当边缘服务器在面对大量的突发性任务时,会使边缘服务的计算能力与负载能力不匹配,从而导致任务的处理时延增加.

深度强化学习已经成功地应用于广泛的挑战性领域,如路径规划和机器人控制. 由于机器学习在函数逼近方面的突破,深度强化学习算法可以准确地识别状态之间的差异[12],准确地掌握问题的解空间,其函数近似器具有强大的拟合性,在解决卫星网络中计算卸载和资源分配问题上开始逐渐受到研究者的关注. Zhou 等人[13]研究了在天地一体化网络中面向延迟的物联网计算任务的卸载问题,提出了一种基于风险敏感的深度强化学习方法来解决任务的卸载与资源的分配. Sthapit 等人[14]研究了卫星网络计算任务卸载过程中产生的危险与威胁,提出了一种基于深度强化学习的计算卸载风险感知算法,以应对卫星网络系统中的安全问题. Chen 等人[15]研究了联合管理卸载路径选择和资源分配来卸载计算密集型和延迟敏感的任务,采用了一种深度强化学习方法来做出最佳决策. Jiang 等人[16]研究了多层卫星网络中的资源管理问题,提出了一种基于 Q 学习的长期最优资源分配算法,该算法可以根据当前状态和过去的状态来调整资源分配方案,以优化系统的长期效用.

本文研究了卫星物联网中计算卸载和资源分配问题,考虑到边缘服务器在面对大量的突发型任务时,边缘服务器出现计算能力与负载能力不匹配的情况,提出了一种基于 D^3PG 的联合资源分配的边缘计算卸载算法. 该算法利用 Double Q-learning 思想和 Dueling 架构重新设计了 DDPG 算法中的价值网络,以提高计算卸载决策的准确性. 在实现最小化系统的平均时延和能耗的同时,保证了训练过程的稳定.

1　系统模型

1.1　任务模型

卫星边缘计算系统模型由 IoT 终端设备、地面站、信关站、边缘服务器、低轨卫星网络和云计算中心组成. IoT 终端设备具有一定的计算资源,可以对计算要求较低的任务进行处理;边缘服务器部署在靠近终端设备的地面站中,终端设备产生的计算任务可以就近在边缘服务器上处理,大大减少的卫星回传延迟,缓解整个低轨卫星网络的流量压力;低轨卫星网络作为 IoT 终端设备与云计算中心之间的中继节点,不具备处理任务的能力,当边缘服务器不能满足任务的计算需求时,可以通过低轨卫星网络将任务转发到云计算中心执行;云计算中心拥有大量的计算资源,可以为任务提供最优质的计算服务,但由于距离 IoT 终端设备较远,会产生大量的传输延迟. 信关站配备了强大的定向天线,能够支持大量的用户群,如蜂窝网络或区域 IP 网络,可以通过星地链路与低轨卫星进行通信. 地面站构建了区域的 IP 网络或蜂窝网络,为区域内的 IoT 终端设备提供接入网络服务. 如图 1 所示.

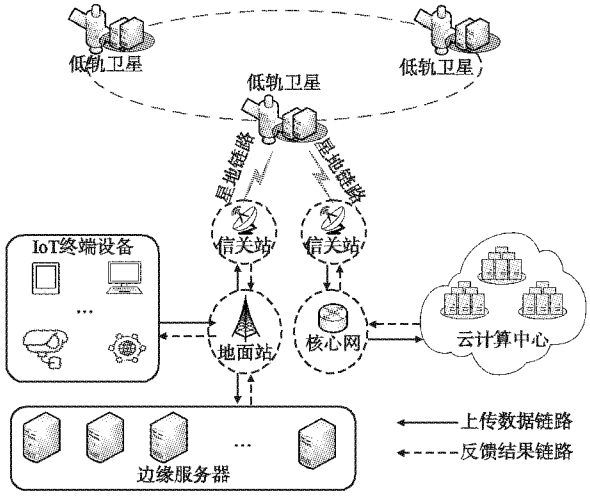

图 1　卫星边缘计算系统模型

Fig. 1　Satellite edge computing system model

为了便于表达和分析,定义 IoT 终端设备的集合为 $U=\{U_i \mid 1 \leqslant i \leqslant K\}$,所有终端用户产生任务的集合为 $T=\{T_{i,j} \mid 1 \leqslant i \leqslant K, 1 \leqslant j \leqslant N\}$. 假设 IoT 终端设备 U_i 产生了一个数据大小为 $\varphi_{i,j}$ 的计算任务 $T_{i,j}$,首先,终端设备获得计算任务的需求信息;然后,终端设备依据计算任务的需求和卫星物联网中计算资源的使用情况,结合卸载算法将计算任务分配到指定的节点进行处理;最后计算任务在指定的节点完成计算后将结果返回到终端设备. 因此,在整个任务卸载过程中,系统的平均时延和能耗是评价卸载策略好坏的重要性能指标.

一般的,本文采用一个元组 $(\varphi_{i,j},\rho_{i,j})$ 表示每个终端设备产生的计算任务 $T_{i,j}$. 这里 $\varphi_{i,j}$ 表示终端设备 U_i 产生的计算任务 $T_{i,j}$ 的数据大小;$\rho_{i,j}$ 表示终端设备 U_i 产生的计算任务 $T_{i,j}$ 的最大可容忍时延.

1.2　通信模型

本文采用无线电磁波进行星地间的通信,根据香农公式,信关站到 LEO 卫星的上行传输速率 R_{up} 可表示为:

$$R_{\mathrm{up}}=\omega \log_2\left(1+\frac{P_{up}G_{up}}{\sigma^2}\right) \tag{1}$$

其中,G_{up} 表示信关站的发射天线增益;P_{up} 表示信关站的上行发射功率;ω 表示无线信道的带宽;σ^2 表示噪声功率.

同理,LEO 卫星到信关站的下行传输速率 R_{down} 可表示为:

$$R_{down}=\omega\log_2(1+\frac{P_{down}G_{down}}{\sigma^2}) \tag{2}$$

其中,P_{down} 表示 LEO 卫星的下行发射功率;G_{down} 表述 LEO 卫星的发射天线增益.

1.3 计算模型

1.3.1 本地计算

由于 IoT 终端设备具有一定的计算资源,可以处理对于一些计算要求较低的计算任务. 因此,计算任务 $T_{i,j}$ 在终端设备 U_i 上处理,产生的时延可表示为:

$$d_{local}^{i,j}=\frac{\varphi_{i,j}\gamma}{f_u^i} \tag{3}$$

其中,f_u^i 为终端设备 U_i 上剩余的计算资源.

计算任务 $T_{i,j}$ 在终端设备 U_i 上处理,需要消耗的能量可表示为:

$$e_{lcoal}^{i,j}=\kappa(f_u^i)^2\varphi_{i,j}\gamma \tag{4}$$

其中,κ 表示能量因子,其数值取决于采用的 CPU 芯片架构. 结合式(3)和式(4),可以得到本地计算处理的成本函数:

$$D_{local}^{i,j}=\alpha d_{local}^{i,j}+\beta e_{local}^{i,j} \tag{5}$$

其中,α 表示时延占成本函数的比重,β 表示能耗占成本函数的比重,且 $\alpha+\beta=1$.

1.3.2 边缘服务器

如果边缘服务器上的计算资源不能满足计算任务的需求,新到达的计算任务在计算队列中等待被处理,要么将计算任务卸载至云计算中心处理. 特别注意地,可以不用考虑终端设备到边缘服务器的传播时延. 因此,计算任务 $T_{i,j}$ 卸载到边缘服务器上处理,整个过程中产生的时延可表示为:

$$d_{\text{edge}}^{i,j}=d_{wait}^{up}+\frac{\varphi_{i,j}}{R_e}+\frac{\varphi_{i,j}\gamma}{f_s^i} \tag{6}$$

其中,d_{wait}^{up} 表示计算任务在边缘服务器上等待被执行的排队时延;R_e 表示从终端设备到边缘服务器的传输速率;$\varphi_{i,j}/R_e$ 表示计算任务从终端设备到边缘服务器的传输时延;f_s^i 表示计算任务在边缘服务器上处理完消耗的计算资源;$\varphi_{i,j}\gamma/f_s^i$ 表示计算任务在边缘服务器上处理完的计算时延.

计算任务 $T_{i,j}$ 卸载到边缘服务器上处理,整个过程中的能耗可表示为:

$$e_{edge}^{i,j}=\kappa(f_s^{i,j})^2\varphi_{i,j}\gamma+\frac{\varphi_{i,j}}{R_e}\cdot P_{i,s} \tag{7}$$

其中,$\kappa(f_s^{i,j})^2\varphi_{i,j}\gamma$ 表示计算任务在边缘服务器上执行的能耗;$P_{i,s}$ 表示终端设备的传输功率;$(P_{i,s}\varphi_{i,j})/R_e$ 表示计算任务从终端设备到边缘服务器的传输能耗. 结合式(6)和式(7)选择边缘服务器卸载的成本函数为:

$$D_{edge}^{i,j}=\alpha d_{edge}^{i,j}+\beta e_{edge}^{i,j} \tag{8}$$

1.3.3 云计算中心

若边缘服务器不能满足任务的计算需求,则计算任务可通过低轨卫星中继卸载至云计算中心进行处理. 假设云计算中心拥有无限的计算资源,任何计算任务到达云计算中心都能立即被处理,不存在排队等待情况,并且分配给每个计算任务的计算资源恒定. 因此,根据式(1)和式(2)得,计算任务 $T_{i,j}$ 卸载到云计算中心处理,整个过程中产生的时延可表示为:

$$d_{cloud}^{i,j}=\frac{\varphi_{i,j}\gamma}{f_o}+\frac{\varphi_{i,j}}{R_{\text{up}}}+\frac{\varphi_{i,j}}{R_{down}} \tag{9}$$

其中,$\varphi_{i,j}\gamma/f_o$ 表示计算任务在云计算中心处理的计算时延,f_o 表示云计算中心分配给计算任务的计算资源;$\varphi_{i,j}/R_{up}$ 表示计算任务从信关站到低轨卫星的上行传输时延;$\varphi_{i,j}/R_{down}$ 表示计算任务从低轨卫星到信关站的下行传输时延.

计算任务 $T_{i,j}$ 卸载到云计算中心处理,整个过程中的能耗可表示为:

$$e_{cloud}^{i,j}=\kappa(f_o)^2\varphi_{i,j}\gamma+\frac{\varphi_{i,j}}{R_{\text{up}}}P_{up}+\frac{\varphi_{i,j}}{R_{down}}P_{down} \tag{10}$$

其中,$\kappa(f_o)^2\varphi_{i,j}\gamma$ 表示计算任务在云计算中心处理的能耗,f_o 表示云计算中心分配给计算任务的恒定计算资源;$(P_{up}\varphi_{i,j})/R_{up}$ 表示计算任务从信关站到低轨卫星的上行传输能耗;$(P_{down}\varphi_{i,j})/R_{down}$ 表示计算任务从低轨卫星到信关站的下行传输能耗. 结合式(1),式(2),式(9)和式(10)选择云计算中心卸载的成本函数为:

$$D_{cloud}^{i,j}=\alpha d_{cloud}^{i,j}+\beta e_{cloud}^{i,j} \tag{11}$$

2 问题描述

对于终端设备产生的计算任务,要么本地执行,要么卸载至边缘服务器上执行,要么卸载至云计算中心执行. 因此本文引入一个卸载决策 $X_{i,j}$ 表示计算任务的卸载情况:

$$X_{i,j}=\{x_{i,j},y_{i,j},z_{i,j}\} \tag{12}$$

$$x_{i,j}\in\{0,1\},y_{i,j}\in\{0,1\},z_{i,j}\in\{0,1\} \tag{13}$$

$$x_{i,j}+y_{i,j}+z_{i,j}=1 \tag{14}$$

其中,$x_{i,j}$ 表示终端设备产生的计算任务是否在本地执行;$y_{i,j}$ 表示终端设备产生的计算任务是否卸载到边缘服务器上执行;$z_{i,j}$ 表示终端设备产生的计算任务是否卸载到云计算中心执行. 由系统模型得,完成所有任务耗费的系统平均成本可表示为:

$$D=\frac{1}{|T|}\sum_{U_i}^{U}\sum_{T_{i,j}}^{T}(x_{i,j}D_{local}^{i,j}+y_{i,j}D_{edge}^{i,j}+z_{i,j}D_{cloud}^{i,j}) \tag{15}$$

本文的目的是找到使整个系统的平均成本最小的卸载决策和资源分配策略,终端设备根据任务信息和系统的资源情况决定计算任务的卸载方式,则针对优化问题 $P1$ 的目标函数可以表述如下:

$$P1:\min_{X,P}D \tag{16}$$

$$s.t.\ C1:(12)(13)(14),\forall T_{i,j}\in T \tag{17}$$

$$C2:x_{i,j}d_{local}^{i,j}+y_{i,j}d_{edge}^{i,j}+z_{i,j}d_{cloud}^{i,j}\leqslant\rho_{i,j} \tag{18}$$

$$C3:\sum_{U_i}^{U}\sum_{T_{i,j}}^{T}f_s^{i,j}\leqslant f_s \tag{19}$$

$$C4:f_s^{i,j}>0,\forall T_{i,j}\in T \tag{20}$$

其中,C_1 是卸载约束,表示计算任务可以本地执行、边缘服务器或者云计算中心上执行;C_2 是任务时延容忍约束,表示任务从产生到执行完成的时间间隔要在给定的 $\rho_{i,j}$ 范围之内;C_3 和 C_4 计算资源约束,C_3 表示边缘服务器上计算任务占用的计算能力不得超过边缘服务器拥有的最大计算能力.

3 基于 D³PG 强化学习算法的解决方案

上述问题 $P1$ 是一个任务数量未知的整数非线性优化问

题,一般的方法难以在有限的时间内给出最优解. 考虑边缘服务器计算资源的动态变化和终端设备产生任务的不确定性,本文采取一个基于强化学习的自适应卸载决策去处理这样的问题. 因此,本文首先将这个问题重新表示为系统中最优的马尔可夫决策过程求解问题,然后提出了一种基于 D^{3}PG 的计算卸载算法,并给出了详细的算法设计方案.

3.1　问题建模

1)状态空间:智能体根据当计算任务的状态信息和卫星物联网环境中计算资源情况,为终端设备产生的计算任务分配合适的计算资源和选择最佳的卸载节点. 假设 t 时隙的状态 $s_t \in S$(S 为状态空间)定义为:

$$s_t = \{\boldsymbol{u}(t), \boldsymbol{v}(t), \boldsymbol{\tau}(t)\} \tag{21}$$

其中,$\boldsymbol{u}(t) = \{u_1(t), u_2(t), \cdots, u_{K+1}(t)\}$,表示 t 时隙终端设备 U_i 和边缘服务器的可用计算资源;$v(t) = \{v_1(t), v_2(t), \cdots, v_K(t)\}$,表示 t 时隙终端设备 U_i 产生计算任务的数据量,$\boldsymbol{\tau}(t) = \{\tau_1(t), \tau_2(t), \cdots, \tau_K(t)\}$ 表示 t 时隙终端设备 U_i 产生计算任务的执行时间限制.

2)动作空间:为了使任务的长期平均成本最小化,并尽可能利用边缘服务器的资源,智能体必须根据每个时隙 t 环境的状态 s_t 做出适当的行动决策. 本文不仅考虑离散的任务卸载决策,还考虑了连续的资源分配决策,因此,动作空间 a_t 可以定义为:

$$a_t = \{\boldsymbol{H}(t), \boldsymbol{B}(t)\} \tag{22}$$

其中,$\boldsymbol{H}(t) = \{H_1(t), H_2(t), \cdots, H_K(t)\}$ 表示终端设备产生计算任务的卸载决策,$H_i(t) = 2$ 表示计算任务在云计算中心上处理,$H_i(t) = 1$ 表示计算任务在边缘服务器上处理,$H_i(t) = 0$ 表示计算任务在终端设备上处理;$\boldsymbol{B}(t) = \{B_1(t), B_2(t), \cdots, B_K(t)\}$ 表示为每个终端设备产生计算任务分配的计算资源,$B_i(t) = 0$ 表示没有计算任务产生,不需要分配计算资源.

3)奖励函数:本文的优化目标是为最小化任务的长期平均成本,长期平均成本 R 表示为:

$$R = \sum_{t=1}^{|T|} r_t, \ \forall t \in \tau \tag{23}$$

其中 r_t 表示执行动作 a_t 后获得的及时奖励,可用来评价智能体所做卸载决策的好坏程度,其具体的定义可由式(24)和式(25)给出:

$$D_t = (D_{local}^{i,j})_t - (x_{i,j} \cdot D_{local}^{i,j} + y_{i,j} \cdot D_{edge}^{i,j} + z_{i,j} \cdot D_{cloud}^{i,j})_t \tag{24}$$

$$r_t = \begin{cases} \sum_{i \in N} D_i, & st(17) \sim (20) \\ -\mu, & otherwise \end{cases} \tag{25}$$

当正智能体的卸载决策满足公式(17)~公式(20)的约束条件时,环境则会根据式(24)给定智能体卸载决策的奖励进行奖励;否则,则卫星物联网环境会给予智能体一个惩罚值 μ,以此告诫智能体选择当前动作是不合适的.

3.2　算法设计

近年来,许多基于价值的深度强化学习算法被应用于边缘环境,如 DQN[17]、Double DQN[18,19],这些方法所面临的挑战是如何处理连续的动作空间. 目前,一般的方法是量化动作空间,但过于粗糙的离散化会导致大量行为信息的丢失,而过细的离散化会导致维度的快速增加. 由于卫星边缘计算卸载过程涉及连续控制,本文通过改进 DDPG 方法,提出了一种基于 D^{3}PG 联合资源分配的边缘计算卸载算法.

DDPG 算法[20]采用 Actor-Critic 架构[21]进行训练,Actor 通过确定性策略梯度方法学习最优策略,Critic 通过价值函数 Q(s,a)对 Actor 选择的动作进行评估. Actor 中包含一个在线策略网络 $\pi_\phi(s)$和一个目标策略网络 $\pi_{\phi'}(s)$,在线策略网络负责将输入状态 s_t 映射到动作 a_t;Critic 中包含一个在线 Q 网络 $Q_\theta(s,a)$和一个目标 Q 网络 $Q_{\theta'}(s,a)$,在线 Q 网络以状态 s_t 和动作 a_t 为输入,输出 Q 网络的期望奖励值 $Q(s_t,a_t)$. 然而,Actor-Critic 框架中对 Q 值的预测准确性,会对算法学习的稳定性产生至关重要的影响.

本文利用 Dueling 网络架构[22]可以解决 DDPG 计算卸载问题中算法学习稳定性的问题. Dueling 网络架构中价值网络和优势网络分别学习价值函数 $V(s)$和动作优势函数 $\Lambda(s,a)$,通过将 DQN 中全连接层改成两条流,其中一条输出关于状态的价值另外一条输出关于动作的优势函数值,最终合并为 Q 函数. Dueling DQN 的 Q 函数是由状态的价值函数 $V(s)$加上每个动作的优势函数 $\Lambda(s,a)$得到,表示公式如下:

$$Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta},\zeta,\eta) = V(\boldsymbol{s};\boldsymbol{\theta};\eta) + \Lambda(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta},\zeta) \tag{26}$$

通过引入 Dueling 网络架构,虽然提高基于 DDPG 卸载算法的稳定性,但是依然不可避免算法的过估计问题,这是由于 Q 网络和目标网络过于相似造成. Double Q-learning[23]的思想能够很好的解决上述问题,它通过使用两个单独的 Q 值评估器去消除过估计问题. 具体来说,每个 Q 值评估器都可以用来相互更新,通过相对方的估计值选择的动作,然后进行无偏估计. 该思想通过设置一对策略网络($\pi_{\phi_1}, \pi_{\phi_2}$)和一对价值网络($Q_{\theta_1}, Q_{\theta_2}$),然后 π_{ϕ_1}和 π_{ϕ_2}分别针对 Q_{θ_1}和 Q_{θ_2}进行了优化:

$$y_1 = r + \gamma Q_{\theta_2'}(s', \pi_{\phi_1}(s')) \tag{27}$$

$$y_2 = r + \gamma Q_{\theta_1'}(s', \pi_{\phi_2}(s')) \tag{28}$$

然而,两个价值网络使用相同的经验回放池去学习经验,并且它们并不是完全独立的. 因此,Double Q-learning 并不能完全消除过高估计. 在上述的基础上,通过从两个估计中选择较小的一个来给出目标网络参数更新,可以解决算法的过估计.

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi_i}(s')) \tag{29}$$

通过上述一系列的改进,相比较于传统的 DDPG 算法,在 Critic 中,通过引入了 Dueling 网络架构和 Double Q-learning 思想来修改 Q 值的估计方式,并且对 Critic 中的 Dueling 网络进行了重新设计,使其能够适应连续的动作空间. 如图 2 所示,给出了 D^{3}PG 算法的网络结构图. 因此,提出了一种基于 D^{3}PG 联合资源分配的边缘计算卸载算法去求解联合资源分配的边缘计算卸载问题,可以实现比原 DDPG 更稳定、更快的训练过程.

为了更好的描述本文提出的基于 D^{3}PG 联合资源分配的边缘计算卸载算法,这里给出算法的伪代码,如算法 1 所示.

算法 1. 基于 D^{3}PG 联合资源分配的边缘计算卸载算法

输入:边缘服务器、终端设备和卫星的状态信息,计算任务的状态信息

输出:计算任务卸载策略

1. 随机初始化 Double Dueling 价值网络 Q_{θ_1}的参数 θ_1 和 Q_{θ_2}的参数 θ_2;
2. 随机初始化策略网络 π_ϕ 的参数 ϕ;
3. 初始化目标网络参数 $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$;
4. 初始化经验复用池 $D = \varnothing$;

5. **for each episode**:
6. 获取环境初始的系统全局状态 S_0;
7. **for each step**:
8. 根据在线策略网络 $\pi_\phi(s_t)$选择动作 a_t;
9. 执行动作 a_t,从环境中获得奖励 r_t 和下一个状态 s_{t+1};
10. 将(s_t,a_t,r_t,s_{t+1})存储经验复用池 D 中;
11. 从 D 中随机采样含有 N 条经验数据(s_t,a_t,r_t,s_{t+1})的小批量;
12. 获得目标动作 $a'_{i+1}=\pi_{\phi'}(s_{i+1})$;
13. 根据目标动作计算:$Q'_{i+1}=\min_{i=1,2}Q_{\theta'_i}(s_{i+1},a'_{i+1})$;
14. 得到目标 $y_i=r_i+\gamma Q'_{i+1}$;
15. 最小化损失函数更新 Double Dueling 价值网络:
$\theta_i\leftarrow\arg\min_{\theta_i}\frac{1}{N}\sum(y_i-Q_{\theta_i}(s,a))^2$;
16. 根据确定性策略梯度更策略网络:
$\nabla_\phi J(\phi)=\frac{1}{N}\sum\nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\phi(s)}\nabla_\phi\pi_\phi(s)$;
17. 更新 Double Dueling 价值目标网络和策略目标网络:
$\theta_i\leftarrow\tau\theta_i+(1-\tau)\theta'_i$
$\phi'\leftarrow\tau\phi+(1-\tau)\phi'$
18. **end for**
19. **end for**

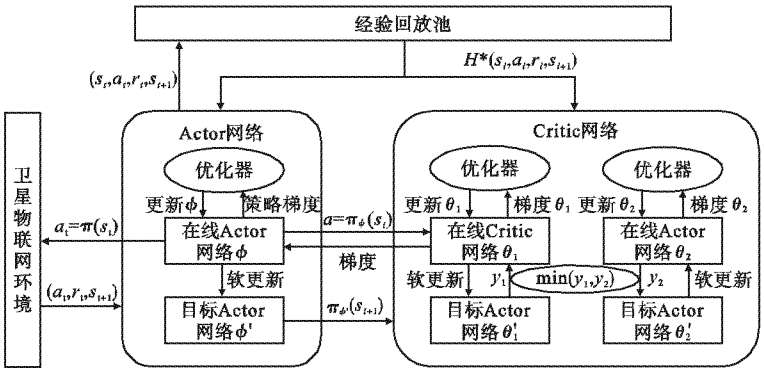

图 2 D^3PG 网络结构图

Fig. 2 Network structure diagram of D^3PG algorithm

算法训练的具体流程为:首先,终端设备通过获取卫星物联网边缘环境中的边缘服务器、终端设备和卫星的状态信息以及计算任务的状态信息;其次,智能体根据获得的状态信息 s_t,并将状态信息作为在线策略网络 π_ϕ 的输入,得到最佳的任务卸载决策和资源分配决策 a_t;接着,卫星物联网边缘环境对相应的动作 a_t 进行响应,得到响应的奖励 r_t 和下一个状态 s_{t+1},并将状态 s_t、动作 a_t、奖励 r_t 以及下一个状态 s_{t+1} 作为一条经验数据存放到经验复用池中;然后,智能体从经验复用池中随机的采样数据,用于训练构建的网络模型;最后,通过不断地训练策略网络和价值网络,得到最佳的卸载策略. 根据算法 1 的伪代码分析,算法的时间复杂度受到训练周期 episode 和时间步 step 的影响,当 episode 和 step 的非常大时,算法的时间复杂度可以表示为:$T(epsiode,step)=O(n_{step}n_{epsiode})$,其中 n_{step} 表示每个 episode 中 step 的数量,$n_{epsiode}$ 表示 episode 的数量.

4 仿真与结果分析

4.1 仿真环境与参数设置

本文假设在卫星物联网和边缘计算结合的场景中,系统由多颗低轨卫星节点、1 个边缘服务器、4 个终端设备和一个云计算中心组成. 对于每个终端设备产的任务,任务的数据大小在 100kb 到 400kb 之间,任务的计算负载 γ 大小为 25cycles/bit,终端设备的计算能力为 500MHz;边缘服务器的计算能力在 5GHz 到 25GHz 之间,云计算中心给每个计算任务分配的计算能力为 2GHz,系统的总上传带宽和下载带宽为 2MHz,传输功率为 30mW,实验中的默认参数见表 1.

对于所提出的 D^3PG 算法,使用全连接神经网络来训练模型,D^3PG 和 DDPG 的神经网络由 PyTorch 框架实现,神经网络两个隐藏层的神经元数量分别设置为 200 和 300. 每次训练采用数据大小为 64 的小批处理,并使用大小为 2000 的经验复用池,目标网络软更新的 τ 设置为 0.02.

表 1 实验参数

Table 1 Experimental parameters

名 称	值
终端设备的计算能力 f_u^i	500MHz
边缘节点的计算能力 f_s^i	5GHz-25GHz
云计算中心分配的计算能力 f_c^n	2GHz
频道的带宽 w	2MHz
传输功率 P_B	30mW
任务数据大小 $\varphi_{i,j}$	100kb-400kb
计算负载 γ	25cycles/bit

4.2 算法收敛性分析

为了评估超参数对所提算法性能的影响,通过设置不同的学习率对算法收敛性进行评估. 在仿真实验中,学习率分别被设定为 $1e^{-3}$、$1e^{-4}$和 $1e^{-5}$. 从图 3 可以看出,学习率为 $1e^{-5}$ 时,曲线在 100 个 episode 后收敛到最优值,并在达到收敛后保持稳定状态. 然而,当学习率为 $1e^{-3}$或 $1e^{-4}$时,需要花费大于 50 个 episode 才能达到收敛状态. 由此可以得出结论,曲线的最优值与学习率的大小不成正比,当算法中优化器的学习率太小时,算法需要更多的训练回合来达到收敛状态,当学习率过大时,曲线不一定能收敛到一个较好的值,甚至会导致训练不稳定.

4.3 对比实验

通过将本文提出的方法与其它方法进行对比,证明本文所提方法的优越性. 对比的其它算法如下:a) DDPG;b) DQN;

c)贪婪方法;d)随机方法.

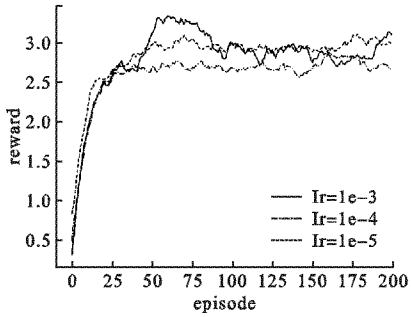

图 3　学习率对算法收敛性的影响

Fig. 3　Influence of learning rate on convergence

1)为验证任务数据量大小对系统平均延迟、任务完成率和能量消耗影响,本文通过设置了任务数据量从100kb增长到400kb,增量为50kb. 图4(a)表明,随着计算任务数据量的增加,所有算法的平均时延都在增加. 这是因为任务处理需要系统中的各种资源,随着任务数据量的增加,终端设备之间对资源的竞争会不断加剧,在这种情况下,为保证计算任务在约束的时间内能够被完成,分配给每个任务的资源会相应的增加,从而增加了任务处理的平均延迟. 图4(b)显示了随着计算任务数据量的增加,所有算法的能量消耗也都在增加,系统的能量消耗与任务数据量成正比的. 图4(c)表明随着计算任务数据量的增加,所有算法的任务完成率都在不断地下降. 这是由于任务数据量的增加,计算任务将会占用大量的计算资源,然而新到达的计算任务得不到计算资源分配,出现等排队现象产生等待延迟,导致不能在规定的时间内完成任务.

由图4可知,当任务数据量大小设置为250kb时,D^3PG的性能优于DDPG,平均延迟缩短了11.56%,能量耗降低了13.97%,任务完成率提高了8.05%. 同时,D^3PG的性能优于DQN,服务延迟缩短了29.05%,能耗降低了208.71%,任务成功率提高了25.54%. 显然,D^3PG在平均延迟、能量消耗和任务完成率方面优于其他算法.

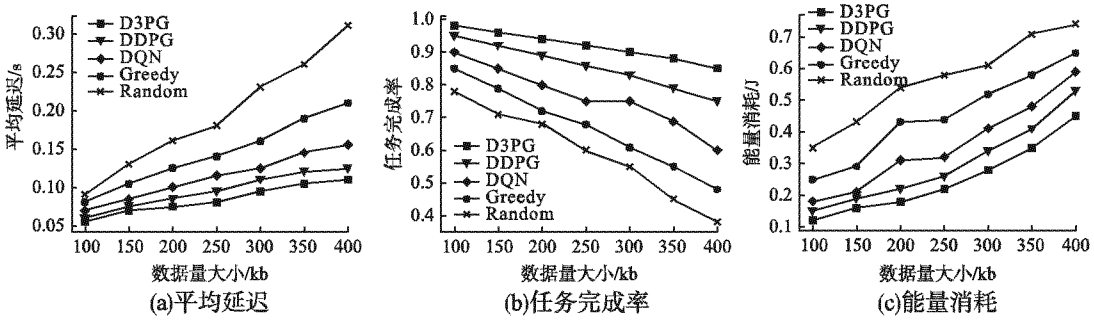

图 4　任务数据量大小对系统平均延迟、任务完成率和能耗的影响

Fig. 4　Impact of task data size on system average delay, success rate and energy consumption

2)为验证边缘服务器的计算能力对系统平均延迟、任务完成率和能量消耗影响,本文通过设置了边缘服务器的计算能力从5GHz增长到25GHz,增量为2.5GHz.

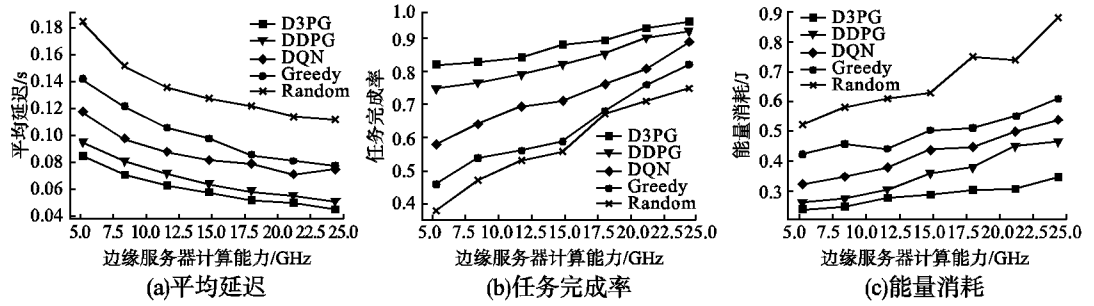

图 5　边缘服务器计算能力对系统平均延迟、任务完成率和能量消耗的影响

Fig. 5　Impact of computing capacity on system average delay, success rate and energy consumption

图5(a)表明随着边缘服务器计算能力的增加,整个系统的平均延迟都在不断地降低. 其原因是随着边缘服务器计算能力的不断增强,可以处理更多的计算任务,更多的计算任务会选择卸载到边缘服务器上处理,这样会大大缩短了计算延迟,降低系统的平均时延. 图5(b)显示了随着边缘服务器计算能力的增加,所有算法的能量消耗也都在增加. 这主要是因为边缘服务器计算能力的增强,更多的计算任务将会被执行,对于能量的消耗也将会大大的增加. 图5(c)表明随着边缘服务器计算能力的增加,所有算法的任务完成率也在不断地上升. 其原因是边缘服务器计算能力的增加,加快了处理任务的效率,更多处于等待状态的任务能够立即被处理,大大提升了任务的完成率.

5 总 结

本文研究了卫星物联网中联合资源分配的边缘计算卸载问题,为了提高终端用户的服务质量,考虑了系统平均延迟、能量消耗和任务完成率等因素.针对计算资源有限的边缘服务器在面对大量的突发卸载任务时,出现计算能力与负载能力不匹配的情况,提出了一种基于 D^3PG 的联合资源分配的边缘计算卸载算法,通过利用 Double Q-learning 思想和 Dueling 架构重新设计了 DDPG 算法中的价值网络,提高计算卸载决策的准确性.仿真实验结果表明,与其他强化学习算法相比,D^3PG 在降低系统平均延迟和能量消耗、提高任务完成率等方面具有更好的性能.

References:

[1] Kim T, Kwak J, Choi J P. Satellite edge computing architectures and network slice scheduling for IoT support[J]. IEEE Internet of Things Journal,2021,9(16):14938-14951.

[2] GOU L,ZUO Y P,WAN Y Y,et al. Summary of IoT for low earth orbit satellites[J]. Informatization Research,2022,48(5):1-9.

[3] Zong B Q,Fan C,Wang X Y,et al. 6G technologies:key drivers, core requirements, system architectures, and enabling technologies [J]. IEEE Vehicular Technology Magazine,2019,14(3):18-27.

[4] Xie R C,Tang Q,Wang Q,et al. Satellite-terrestrial integrated edge computing networks:architecture, challenges, and open issues[J]. IEEE Network,2020,34(3):224-231.

[5] YU X,CHEN Y B,LIU H,et al. Strategy of joint resource allocation and computation offloading in LEO satellite edge computing scenario[J]. Journal of Nanjing University of Posts and Telecommunication(Natural Science Edition),2021,41(6):1-9.

[6] Wei J,Cao S. Application of edge intelligent computing in satellite Internet of Things[C]//IEEE International Conference on Smart Internet of Things(SmartIoT),2019:85-91.

[7] Zhang Z,Zhang W,Tseng F H. Satellite mobile edge computing:improving QoS of high-speed satellite terrestrial networks using edge computing techniques[J]. IEEE Network,2019,33(1):70-76.

[8] Wang Y,Zhang J,Zhang X,et al. A computation offloading strategy in satellite terrestrial networks with double edge computing [C]//IEEE International Conference on Communication Systems (ICCS),2018:450-455.

[9] Qiu C,Yao H,Yu F R,et al. Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks[J]. IEEE Transactions on Vehicular Technology,2019,68(6):5871-5883.

[10] He M,Zhong L,Tan H,et al. A novel edge computing server selection strategy of LEO constellation broadband network[C]//IEEE World Congress on Services(SERVICES),2020:275-280.

[11] Wang Y,Yang J,Guo X,et al. A game-theoretic approach to computation offloading in satellite edge computing[J]. IEEE Access,2019,8:12510-12520.

[12] Liu Q,Shi L,Sun L,et al. Path planning for UAV-mounted mobile edge computing with deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology,2020,69(5):5723-5728.

[13] Zhou C,Wu W,He H,et al. Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN[J]. IEEE Transactions on Wireless Communications,2020,20(2):911-925.

[14] Sthapit S,Lakshminarayana S,He L,et al. Reinforcement learning for security-aware computation offloading in satellite networks[J]. IEEE Internet of Things Journal,2021,9(14):12351-12363.

[15] Chen T,Liu J,Ye Q,et al. Learning-based computation offloading for iort through Ka/Q-Band satellite terrestrial integrated networks [J]. IEEE Internet of Things Journal,2021,9(14):12056-12070.

[16] Jiang C,Zhu X. Reinforcement learning based capacity management in multi-layer satellite networks[J]. IEEE Transactions on Wireless Communications,2020,19(7):4685-4699.

[17] Lv L,Zhang S,Ding D,et al. Path planning via an improved DQN-based learning policy[J]. IEEE Access,2019,7:67319-67330.

[18] Van Hasselt H,Guez A,Silver D. Deep reinforcement learning with double q-learning[C]//AAAI Conference on Artificial Intelligence,2016,30(1).

[19] Sewak M,Sewak M. Deep Q network(DQN),double DQN,and dueling DQN:a step towards general artificial intelligence[J]. Deep Reinforcement Learning:Frontiers of Artificial Intelligence, 2019:95-108.

[20] Hou Y,Liu L,Wei Q,et al. A novel DDPG method with prioritized experience replay[C]//IEEE International Conference on Systems,Man,and Cybernetics(SMC),2017:316-321.

[21] Peters J,Schaal S. Natural actor-critic[J]. Neurocomputing,2008, 71(7-9):1180-1190.

[22] Wang Z,Schaul T,Hessel M,et al. Dueling network architectures for deep reinforcement learning[C]//International Conference on Machine Learning(PMLR),2016:1995-2003.

[23] Chen X,Wang C,Zhou Z,et al. Randomized ensembled double q-learning:learning fast without a model[J]. arXiv preprint arXiv: 2101.05982,2021.

附中文参考文献:

[2] 苟 亮,左云鹏,万扬洋,等.低轨卫星物联网综述[J].信息化研究,2022,48(5):1-9.

[5] 余 翔,陈宇博,刘 晗,等.一种低轨卫星边缘计算场景下联合资源分配的计算卸载策略[J].南京邮电大学学报(自然科学版),2021,41(6):1-9.