

基于强化学习的移动边缘计算任务卸载方法

陈刚, 王志坚, 徐胜超

(广州华商学院 数据科学学院, 广州 511300)

摘要: 最佳卸载策略直接影响移动计算任务卸载的时延与能耗, 因此提出基于强化学习方法的移动边缘计算任务卸载方法; 首先对移动设备的计算任务卸载形式展开具体分析, 并基于分析结果获取计算任务卸载能量消耗、发射功率、传输速率等相关参数数值, 以此建立移动边缘计算任务卸载模型; 最后基于建立的卸载模型结合 Q-Learning 算法对计算任务实施强化学习, 找出计算任务的最佳卸载策略, 从而实现移动边缘计算任务的实时卸载; 实验结果表明, 使用强化学习方法开展移动边缘计算任务卸载时, 卸载能耗低, 仅为 0.87 J, 时延小, 均不超过 6 s。

关键词: 强化学习方法; Q-Learning 算法; 移动边缘; 计算任务卸载; 卸载模型

Offload Method of Mobile Edge Computing Task Based on Reinforcement Learning

CHEN Gang, WANG Zhijian, XU Shengchao

(School of Data Science, Guangzhou Huashang College, Guangzhou 511300, China)

Abstract: Optimal offloading strategy directly affects the latency and energy consumption of mobile computing task offloading. Therefore, a mobile edge computing task offloading method based on reinforcement learning is proposed. Firstly, the specific analysis of task offloading form for mobile devices is carried out, and based on the analysis results, the parameters such as energy consumption, transmission power, and transmission rate for task offloading are obtained, thereby establishing a mobile edge computing task offloading model. Finally, based on the established offloading model and combined with the Q-Learning algorithm, reinforcement learning is applied to determine the optimal offloading strategy for computing tasks, thus achieving the real-time offloading of mobile edge computing tasks. Experimental results show that when using reinforcement learning for the mobile edge computing task offloading, the offloading energy consumption is low, only 0.87 J, and the latency is small, no more than 6 s.

Keywords: strengthening learning methods; Q-learning algorithm; mobile edge; calculation task unloading; unloading model

0 引言

近年来, 物联网技术^[1-2]的不断发展以及 5G 技术的出现, 成功开启了新型数据革命。然而, 由于物联网设备以及移动智能设备在计算领域的局限性, 应用 APP 的增加使设备本身无法满足应用 APP 的计算需求, 因此需要将设备上的一些计算任务卸载到具有足够计算能力的云服务器上实施计算处理, 从而促进了移动云计算的发展。通常来说, 云服务器多为集中形式, 计算数据量会随着时间的推移发生爆炸式增长。从而出现传播时延慢的情况, 由此有研究人员提出了移动边缘计算^[3], 移动边缘计算能够很好地缓解了网络计算数据量大导致的信道拥堵问题。其中移动边缘计算中的任务卸载作为边缘计算的关键环节, 对其开展有效的任务卸载方法研究, 就成为边缘计算领域急需面对的问题。

文献 [4] 提出基于遗传算法的可持续任务卸载决策, 该算法可以使卸载任务具有可持续性。但是该方法时效性较差, 具有延时性。文献 [5] 针对边缘计算任务卸载时信息延迟问题, 将模拟退火机制引入计算任务卸载中。以优化结果为基础, 改变用户卸载策略, 选取合适的服务器展开卸载执行, 实现移动边缘计算任务的高效卸载。但是, 该方法在进行任务卸载时存在卸载延时较高的问题。文献 [6] 基于移动边缘计算网络多任务卸载状态, 使用强化学习算法对其实施优化处理, 建立边缘计算任务卸载模型。最后通过求解模型, 实现移动边缘计算任务的有效卸载。该方法虽然能够完成边缘计算任务的卸载, 但是在实际应用过程中存在能耗较高的问题。文献 [7] 针对移动边缘计算任务卸载时低能耗、低时延的卸载需求, 提出一种最小化系统响应的任务卸载方案。通过设计的免疫优化算法, 获取最优卸载方案, 完成移动边缘计算时的任务卸载。但

收稿日期: 2023-05-29; 修回日期: 2023-07-12。

基金项目: 国家自然科学基金面上项目(61772221); 广州华商学院校内导师制科研项目资助(2023HSDS07)。

作者简介: 陈刚(1973-), 男, 硕士, 副教授。

王志坚(1970-), 男, 博士, 教授, 硕士生导师。

徐胜超(1980-), 男, 硕士, 副教授。

引用格式: 陈刚, 王志坚, 徐胜超. 基于强化学习的移动边缘计算任务卸载方法[J]. 计算机测量与控制, 2023, 31(10): 306-311, 316.

是, 该方法需要经过多个步骤的计算, 导致该方法存在卸载时间较长的问题。文献 [8] 中提出了一种基于网格筛选的大规模密度峰值聚类算法, 使用网格化方法去除部分密度稀疏的点, 再使用密度峰值聚类算法的决策图方法选取聚类中心。该算法可以在保证聚类准确性的基础上, 有效降低计算复杂度, 并适用于数据的不均匀分布情况。文献 [9] 中提出了一种针对结构复杂数据集的模糊聚类有效性指标。该指标使用类内平方误差和和隶属度权值得到紧致性度量策略, 使用聚类中心距离最小值和各聚类中心到平均聚类中心的距离和, 衍生出分离性测算方法。通过合成得到新的有效性指标 $V_{-}(GSDC)$, 在 $V_{-}(GSDC)$ 的极值处对应的类别数可以自动得到最佳聚类数。

为解决上述移动边缘计算任务卸载过程中存在的问题, 本文提出基于强化学习方法的移动边缘计算任务卸载。本文的创新技术路线如下:

1) 针对移动设备的计算任务卸载形式进行详细分析, 并获取计算任务卸载相关参数值, 如能量消耗、发射功率和传输速率等。

2) 根据上述计算结果, 建立移动边缘计算任务卸载模型。

3) 基于建立的卸载模型并结合 Q-Learning 算法, 实施强化学习, 从而找出最佳的移动边缘计算任务卸载策略。

4) 实现移动边缘计算任务的实时卸载

本文使用的强化学习方法利用 Q-Learning 算法, 具有不依赖模型的特点, 以贪婪策略为目标策略, 通过迭代更新可以获得最优的学习结果。强化学习算法在移动边缘计算任务卸载中应用的创新点主要为: 该算法可通过学习和反馈调整决策策略, 并自适应地调整任务卸载方式, 使得系统能更好地响应网络状态、设备负载等因素, 具有较高的动态性、自适应性、可扩展性和精度提升效果。因此, 在实际应用中, 强化学习与移动边缘计算的结合可实现任务的智能调度和优化, 改善边缘计算的性能和用户体验。

1 移动边缘计算任务卸载模型构建

1.1 卸载方式

用户在享受移动边缘计算带来的便利时, 还需要面临计算任务卸载的决策问题。移动边缘计算过程的任务卸载决策主要是指用户在明确计算任务中时延、能耗各项指标后, 对计算任务是否卸载至边缘服务器展开决策, 该决策会直接影响用户的体验, 是整个边缘计算的关键。

从移动云计算整体架构来看, 计算任务在卸载时通常可以分成水平和垂直两个方向。水平方向一般将框架分成 3 层, 相同层设备和服务器可以看做移动边缘的计算中心。在水平方向上的卸载方式分成设备到设备之间通信以及服务器之间协同计算两种。而在垂直方向上, 则要直接划分 UE 层、移动云计算中心和云计算中心层三层架构。移动边缘垂直方向卸载主要是通过任务计算能力划分计算层级, 任务卸载在三个层级之间, 由低计算能力逐层向高计算能力层卸载, 过程如图 1 所示。

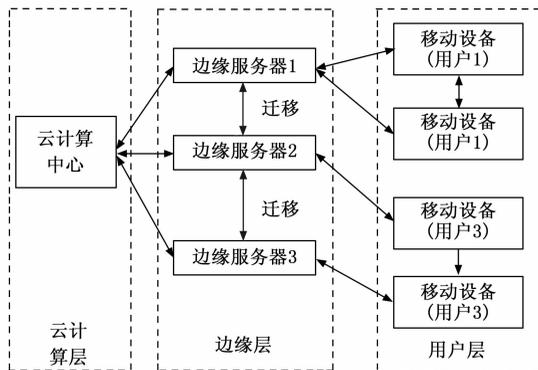


图 1 移动边缘计算任务卸载场景图

从图 1 可知, 用户在决定是否进行边缘计算任务卸载, 主要基于卸载决策, 其中包括任务卸载的任务量以及卸载边缘服务器的决策, 移动边缘计算任务卸载^[10-11]方式如下:

1) 本地计算: 由于移动边缘计算任务通常都是由用户自身 CPU 决定的, 不涉及任务卸载。过程中通常由于移动边缘计算资源的可用信道, 若资源计算结果未达到理想收益, 则不进行计算卸载。

2) 全部卸载: 用户资源计算任务由信道上传至移动边缘计算服务器, 通过服务器展开计算信道的计^[12-13]处理。这一过程中由于信道质量稳定, 计算资源充分, 因此计算任务可直接通过任务卸载形式获取计算收益。

3) 部分卸载: 该形式主要通过任务的两种计算类型来开展, 计算类型为上述的本地计算形式和卸载计算形式。卸载过程中主要针对可拆分任务, 利用本地计算和卸载计算结合的形式, 计算资源的收益, 根据计算结果将要求高的任务部分卸载至移动边缘计算服务器中, 将简单的任务放入用户本地计算中, 从而节省不必要的能耗。三种计算执行方式之间的关系如图 2 所示。

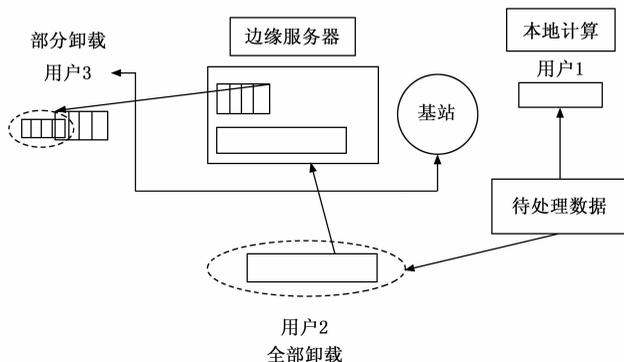


图 2 移动边缘计算执行方式关系图

1.2 构建任务卸载模型

根据对移动边缘计算卸载形式分析结果, 构建用户移动边缘计算卸载网络模型, 模型具体结构如图 3 所示。

基于图 3 可知, 移动边缘计算卸载模型是由 K 个移动设备以及一个具备高性能服务器的基站构建而成。在网络时隙中, 设备可将制定的卸载决策 TDMA (time division

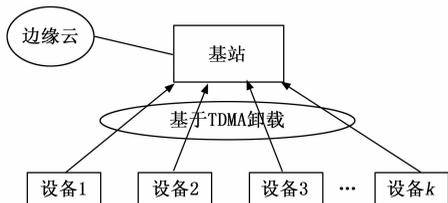


图 3 移动边缘计算卸载网络模型结构

multiple access) 卸载至边缘服务器。

在建立移动边缘计算任务卸载本地执行模型之前，需要进行能量消耗和执行时间的确定，以便准确建立移动边缘计算任务在本地计算设备上的执行模型。首先，对于本地执行，需要考虑移动设备的能量消耗。这包括移动设备在执行任务期间所消耗的能量，例如 CPU 运行、数据传输等过程中所需的功率。可以通过实验或者建立能耗模型来测量或预测移动设备在不同负载和操作条件下的能量消耗。其次，还需要确定移动边缘计算任务在本地执行时的执行时间。这涉及到移动设备处理任务所需的计算资源、存储资源以及网络传输延迟等因素。可以通过在移动设备上运行任务，并测量执行时间，或通过建立执行时间模型来评估任务在本地计算设备上的执行性能。通过准确测量能量消耗和执行时间，可以建立移动边缘计算任务在本地执行设备上的执行模型。这个模型将为后续的决策提供基础，帮助选择合适的卸载策略和资源分配方案，以最小化系统开销和提高任务执行效率。

设定 q_k 表示第 k 个设备任务上传发射功率，信道增益为 f_k ，以此获取任务传输速率^[14-15]以及任务卸载所需时间，结果如下式所示：

$$R_k = C \lg \left(1 + \frac{q_k f_k^2}{\sigma_0} \right) \quad (1)$$

$$\beta'_k = \frac{c_k \chi_k}{R_k} = \frac{c_k \chi_k}{C \lg \left(1 + \frac{q_k f_k^2}{\sigma_0} \right)} \quad (2)$$

式中， σ_0 为复杂高斯噪声的方差， C 为带宽， β'_k 为任务卸载所需时间， R_k 为传输速率， c_k 为卸载决策， χ_k 为卸载任务的大小。

基于上述计算结果，设定任务的发送时隙为 β_k ，当满足 $\beta'_k \leq \beta_k$ 时，针对用户第 k 个设备的卸载能耗，建立移动边缘计算任务卸载模型 Z_{xz} ：

$$Z_{xz} = \epsilon_{off,k} \times \beta'_k \quad (3)$$

式中， $\epsilon_{off,k}$ 为计算出的任务卸载能耗。

2 基于强化学习的移动边缘计算任务卸载

由于上述确定的移动边缘计算任务卸载多目标组合优化问题属于 NP 难问题，因此需要采用深度强化学习中的 Q-Learning 算法^[16-19]解决任务卸载以及资源分配问题。

强化学习算法主要是通过不断的试错获取问题的最优解，该算法通常由状态、动作、汇报以及智能体几个元素组成。强化学习算法通常分为有模型和无模型两种算法，

由于真实网络的场景环境较为复杂，无法有效建立算法模型，因此在移动边缘计算任务卸载过程中，可使用 Q-Learning 算法进行卸载任务强化学习，从而实现移动边缘计算任务的卸载。

2.1 基于 Q-Learning 算法的卸载计算

当面临马尔可夫决策过程 (MDP) 问题时，Q-Learning 是一种基于强化学习的算法，旨在通过不断与环境交互，学习最优策略。该算法利用一个 Q 值表格来表示在不同状态下采取不同动作的预期回报，并通过更新和选择最大 Q 值的方式，迭代地优化策略。Q-Learning 的核心是通过探索和利用的平衡，使智能体能够学习到长期回报最大化的策略，而无需事先了解环境模型。这种算法被广泛应用于游戏智能体训练、路径规划等领域。将用户看做一个智能体，设定系统状态 S 的卸载决策系数为 A ，资源分配系数描述成 H ，剩余以此获取系统状态值以及动作值，过程如下式所示：

$$\begin{cases} S = \{A, H, F\} \\ a = (a_i, a_j) \end{cases} \quad (4)$$

式中， F 为任务卸载时剩余计算资源系数， a 为系统动作值， a_i 为计算任务 T_i 的卸载方案， a_j 为计算任务 T_j 的卸载方案。由于在智能体系统中，卸载决策系数能够直接决定哪些计算任务需要卸载，哪些在本地计算执行。

基于上述计算结果可知，卸载决策在执行动作后，会得到奖励 $R(S, A)$ ，此时奖励函数会与目标功能关联在一起，因此可直接将目标优化问题看做最小化的系统成本开销问题，表述成 $r = \frac{\varphi_{local} - \varphi(s, a)}{\varphi_{local}}$ 形式，其中移动边缘计算任务在本地执行的系统成本开销用 φ_{local} ，当前状态下 JORC 算法的系统总开销表述成 $\varphi(s, a)$ 形式。

在 Q-Learning 算法中，设定智能体在 t 时刻的环境状态为 s ，基于随机选取的动作 a ，进入到 s 中获取响应的奖励值 r ，以此更新 Q 表格和算法当前状态，从而获取最佳的卸载策略，过程如下式所示：

$$Q(s_t, a_t) = Q(s_t, a_t) + \eta * (r_t + \epsilon * \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \quad (5)$$

式中， ϵ 为折扣因子， η 为学习率， $Q(s_t, a_t)$ 为获取的最佳卸载策略。

2.2 权重分配

基于确定的最佳卸载策略，可以构建一个层次结构模型来比较模型中不同准则层因素的重要性。这种层次结构模型将帮助明确各个因素之间的相对权重，从而对移动边缘计算任务进行更合理的决策。首先，需要确定准则层因素。这些因素可包括能源消耗、执行时间、任务负载、设备性能等与任务卸载和资源分配相关的因素。通过比较和分析这些因素，可以确定它们在决策过程中的相对重要性。接下来，根据比较结果，建立一个判断矩阵。判断矩阵是根据准则层因素之间的比较结果构建的，用于定量地表示它们的重要性。通过两两比较准则层因素，并给出其相对

重要性的比较判断, 可以得到一个方阵形式的判断矩阵。然后, 需要进行一致性检验来验证判断矩阵的合理性。一致性检验是为了确保判断矩阵的逻辑一致性, 即判断矩阵中的比较判断是否符合一定的数学规律。通过计算一致性指标 (例如一致性比例 CR), 可以评估判断矩阵的一致性, 以确保判断矩阵的可靠性和合理性。最终, 通过计算一致性检验结果和矩阵运算, 可以得到最终的移动边缘计算任务权值向量。这个权值向量代表了各个准则层因素对于最佳卸载策略的相对重要性。基于这些权值, 可以进行任务卸载和资源分配的决策, 选择最优的移动边缘计算策略, 以达到系统开销的最小化和任务执行效果的最大化。

首先设定层次模型^[20-22]层数为 3 层, 第一层负责任务缓存价值获取, 第二层为计算任务的流行度、任务数据大小以及计算量的获取层; 第三层为任务层。

建立辨识矩阵时, 设定矩阵中第 i 个元素与第 j 个元素之间重要程度为 $b_{i,j}$, 任务的流行度、任务数据大小以及计算量都受任务缓存价值支配, 建立额定辨识矩阵表述成 $\mathbf{B} = b_{i,j}$ 形式。

基于上述建立的辨识矩阵, 利用评价指标算法确定矩阵的不一致程度值以及随机一致性比值概率, 以此完成最终计算任务权重向量的计算, 过程如下式所示:

$$\lambda = w_1 x_i + w_2 y_i + w_3 Q(s_i, a_i) \quad (6)$$

式中, \mathbf{w} 表示与矩阵最大特征值对应的特征向量, λ 为获取的权重向量, w_1, w_2, w_3 分别为任务在度量时流行度、任务 CPU 周期大小以及数据量权值, Q 为表格, x_i 为 T_i 时刻横坐标值, y_i 为 T_i 时刻纵坐标值。

当用户发出移动边缘计算任务的卸载请求时, 移动边缘计算服务器会首先检测是否已将该任务的计算结果缓存起来。如果存在已缓存的计算结果, 服务器将直接返回缓存结果, 并根据预设算法计算卸载策略值。该策略值决定是否将移动边缘计算任务卸载到边缘设备上执行, 以降低系统的总开销成本。然而, 如果计算结果未被缓存, 服务器将继续使用 Q-Learning 算法来决定最佳的卸载策略和资源分配策略, 以最小化系统的开销。Q-Learning 算法通过不断的学习和迭代, 基于环境与动作的奖励情况, 逐步优化策略, 并探索可能的卸载策略和资源分配方案。通过联合优化计算移动边缘计算任务的卸载、资源分配以及任务缓存策略值, 系统可以得到最优的开销成本。这种方法利用缓存和智能决策相结合, 针对不同的任务和环境条件^[23-25], 实现了高效的任务卸载和资源利用, 提高了系统性能和用户体验。

3 仿真实验与性能分析

3.1 仿真实验环境

为了验证本文基于强化学习的移动边缘计算任务卸载方法的整体有效性, 需要对此方法测试。

分别采用基于强化学习方法的移动边缘计算任务卸载 (本文方法)、基于遗传算法的可持续任务卸载决策方法 (文献 [4] 方法)、基于免疫优化的边缘计算任务卸载 (文

献 [7] 方法) 进行对比测试。

测试过程中, 基于计算机仿真平台建立一个虚拟大型的集中式移动边缘计算网络, 仿真环境参数如表 1 所示。

表 1 仿真环境参数

运行环境	配置	参数
硬件环境	CPU	Intel(R) Core(TM) i5-9400
	频率	2.90 GHz
	RAM	16.0 GB
软件环境	操作系统	Windows 10
	版本	18362.1082 专业版
	位数	64 bit
	模拟软件语言	APDL
	仿真软件	Matlab 7.0

设定网络中移动边缘计算服务器位于基站附近, 小企业服务器随机分散在网络中, 覆盖范围为 35 m, 其余测试参数如表 2 所示。

表 2 集中式移动边缘计算网络参数值

参数内容	取值
信道带宽	0.2 MHz
用户最大发送功率	23 dBm
用户计算能力	0.1~1 GHz/周期
计算任务大小	400~1200 kB
客户最大容忍时延	1~4 s
移动边缘计算能力	4 GHz/周期
背景噪声功率	-100 dBm

开展移动边缘计算任务卸载方法有效性检测时, 设定卸载实验方案有 3 种, 如表 3 所示。

表 3 实验方案

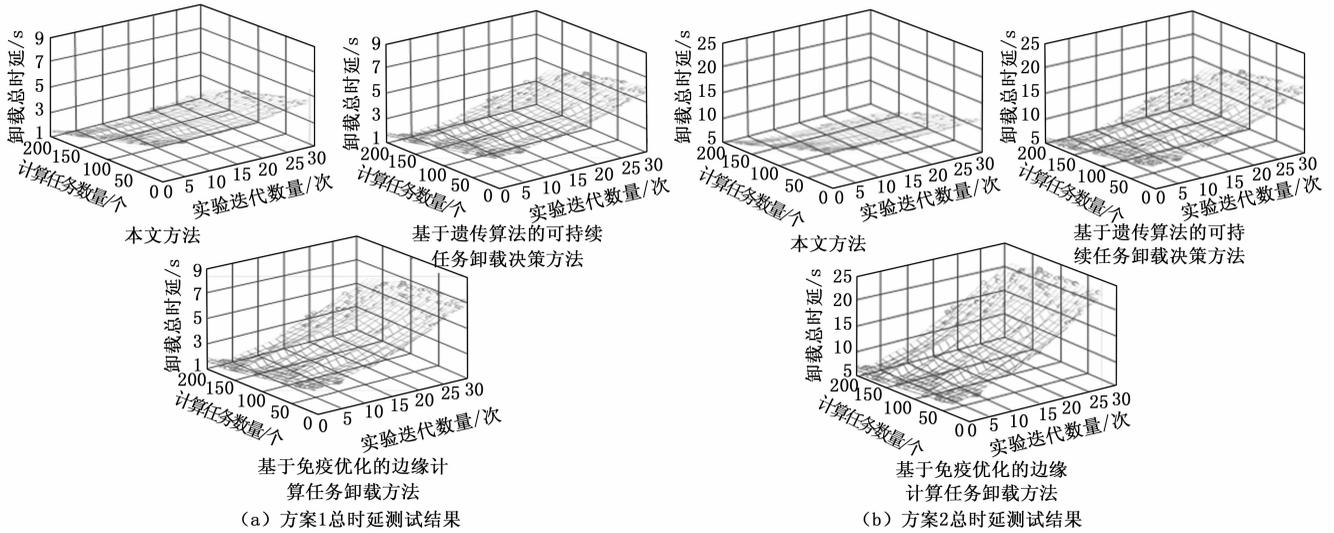
方案名称	内容
方案 1	时延权重 0.3 的卸载决策
方案 2	计算任务全部本地执行
方案 3	计算任务全部卸载

3.2 仿真结果与性能分析

3.2.1 时延测试

选取 200 个计算任务作为卸载任务目标, 基于上述设定的卸载方案, 分别采用基于强化学习方法的移动边缘计算任务卸载 (本文方法)、基于遗传算法的可持续任务卸载决策方法 (文献 [4] 方法)、基于免疫优化的边缘计算任务卸载 (文献 [7] 方法) 开展移动边缘计算任务卸载, 检测上述 3 种方法在移动边缘计算任务卸载时的总时延变化情况, 结果如图 4 所示。

分析图 4 可知, 移动边缘计算任务的数量越多, 任务卸载计算时所测试出的总时延就越高。整体来看, 当移动边缘计算任务都在本地执行时所测试出的总时延较小, 计算任务若全部卸载, 则总时延较大。本文方法在移动边缘



(a) 方案1总时延测试结果

(b) 方案2总时延测试结果

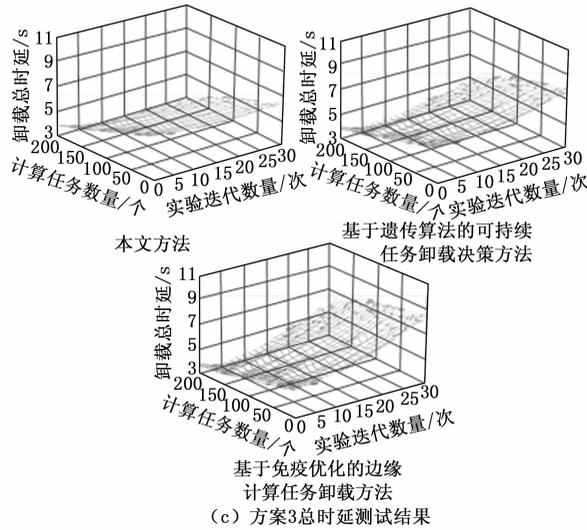


图 4 不同方法任务卸载总时延测试结果

计算任务卸载时，测试出的总时延要优于其他方法的测试结果。其中，基于遗传算法的可持续任务卸载决策方法直接根据优化结果制定卸载策略，所以该方法在完成任务卸载时所用时延较大；基于免疫优化的边缘计算任务卸载方法在设计免疫算法时，算法本身误差较大，所以该方法在任务卸载过程中，测试出的总时延较不理想。而本文方法则使用 Q-Learning 算法对卸载任务展开强化学习，在不断的试错过程中，获取计算任务的卸载策略，所以该方法在任务卸载时，测试出的总时延较为理想。

3.2.2 能耗及时间测试

采用本文方法，文献 [4] 方法以及文献 [7] 方法开展移动边缘计算任务卸载时，对上述 3 种方法的卸载能耗以及卸载时间展开测试，以此验证上述 3 种卸载方法的卸载性能，测试结果如表 4 所示。

分析表 4 可知，在开展移动边缘计算任务卸载时，随着待卸载任务数量的增加，3 种方法测试出的卸载时间以及卸载能耗均出现不同程度的上升趋势。但是整体来看，本文

表 4 不同方法的卸载能耗、时间测试结果

卸载任务数量/个	测试内容	不同方法测试结果		
		本文方法	文献[4]方法	文献[7]方法
50 个	卸载时间/ms	20	36	29
	卸载能耗/J	0.87	1.15	2.43
100 个	卸载时间/ms	35	52	55
	卸载能耗/J	1.38	2.91	4.66
150 个	卸载时间/ms	59	71	80
	卸载能耗/J	3.94	5.83	8.22
200 个	卸载时间/ms	78	97	102
	卸载能耗/J	6.73	9.05	10.07

方法在测试过程中检测出的卸载时间以及任务卸载能耗都是 3 种方法中最好的。由此可证明，本文方法在移动边缘计算任务卸载时，卸载方法具备有效性。本文方法具有较低卸载能耗与较短卸载时间的原因在于，本文方法对任务卸载的相关参数进行精准的计算，并构建任务卸载模型。并且结合深度学习中的 Q-Learning 算法对计算任务进行强

化训练, 从而获得最佳的任务卸载策略。

4 结束语

基于强化学习的移动边缘计算任务卸载方法能够有效地优化数据处理和计算任务的分配, 提高了移动边缘计算系统的性能和效率。然而, 该方法也存在一些局限性。首先, 由于强化学习方法需要大量的训练数据和计算资源, 因此其计算成本相对较高。其次, 当计算任务量较大时, 强化学习可能会产生比较大的决策空间, 导致计算开销增加并降低系统的性能。此外, 基于强化学习的任务卸载方法也比较难以处理多目标和多约束的情况。为了解决这些问题, 未来可以从以下几个方面进行改进。首先, 可以采用深度强化学习等技术提高算法的学习效率和性能, 并通过分布式计算等方式降低计算成本。其次, 可以使用遗传算法等进化算法来处理多目标和多约束问题, 提高决策效率和优化性能。此外, 还可以通过引入传感器网络、边缘计算设备和物联网等技术, 构建更加智能、高效和可靠的移动边缘计算系统, 为实现更广泛的应用场景打下基础^[26-27]。

参考文献:

[1] CUI E, YANG D, WANG H, et al. Learning-based deep neural network inference task offloading in multi-device and multi-server collaborative edge computing [J]. *Transactions on Emerging Telecommunications Technologies*, 2022, 33 (7): 4485-4505.

[2] ENVELOPE M A A B, ENVELOPE M H M A P, ENVELOPE M K H C. Task offloading using GPU-based particle swarm optimization for high-performance vehicular edge computing [J]. *Journal of King Saud University-Computer and Information Sciences*, 2022, 23 (2): 156-161.

[3] WANG Z, GUO C, LIU J, et al. Accurate and privacy-preserving task allocation for edge computing assisted mobile crowdsensing [J]. *IEEE Transactions on Computational Social Systems*, 2021, 9 (1): 120-133.

[4] CHAKRABORTY S, MAZUMDAR K. Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing [J]. *Journal of King Saud University-Computer and Information Sciences*, 2022, 34 (4): 1552-1568.

[5] 张德干, 李 霞, 张 捷, 等. 基于模拟退火机制的车辆用户移动边缘计算任务卸载新方法 [J]. *电子与信息学报*, 2022, 44 (9): 3220-3230.

[6] 邝祝芳, 陈清林, 李林峰, 等. 基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法 [J]. *计算机学报*, 2022, 45 (4): 812-824.

[7] 朱思峰, 孙恩林, 柴争义. 移动边缘计算场景下基于免疫优化的任务卸载 [J]. *西安电子科技大学学报*, 2022, 49 (1): 152-160.

[8] 徐 晓, 丁世伟, 孙统风, 等. 基于网格筛选的大规模密度峰值聚类算法 [J]. *计算机研究与发展*, 2018, 55 (11): 2419-2429.

[9] 唐益明, 丰刚永, 任福继, 等. 面向结构复杂数据集的模糊聚类有效性指标 [J]. *电子测量与仪器学报*, 2018, 32 (4): 119-12.

[10] LU G, LIU M, WANG Z, et al. Research on key technologies of edge computing in discrete manufacturing industry in the era of big data [J]. *Journal of Physics: Conference Series*, 2021, 1952 (4): 422-428.

[11] KIM W S. Edge computing server deployment technique for cloud VR-based multi-user metaverse content [J]. *Korea Multimedia Society*, 2021, 41 (8): 56-62.

[12] GU M, LYU J, LI Z, et al. Research on fast recognition of vulnerable traffic participants in intelligent connected vehicles on edge computing [J]. *Journal of Circuits, Systems and Computers*, 2022, 36 (13): 96-103.

[13] WANG B, LV B, SONG Y. A hybrid genetic algorithm with integer coding for task offloading in edge-cloud cooperative computing [J]. *IAENG International Journal of Computer Science*, 2022, 48 (23): 49-56.

[14] TRUONG V T, HA D B, VAN V N, et al. On the system performance of mobile edge computing in an uplink NOMA WSN with a multiantenna access point over Nakagami-m Fading [J]. *IEEE/CAA Journal of Automatica Sinica*, 2021, 48 (23): 49-56.

[15] KS A, RPY A, KCJ B. Joint MAP channel estimation and data detection for OFDM in the presence of phase noise from free running and phase-locked loop oscillator-science direct [J]. *Digital Communication and Network*, 2021, 7 (1): 55-61.

[16] LIU J, PAN B, ZHANG X, et al. Mobile E-commerce information system based on industry cluster under edge computing [J]. *Mobile Information Systems*, 2021, 11 (6): 1-11.

[17] HAMERLY R, SLUDDS A, BANDYOPADHYAY S, et al. Edge computing with optical neural networks via WDM weight broadcasting [C] // *Emerging Topics in Artificial Intelligence (ETAI)*, 2021.

[18] LIU G, CHEN X, ZHOU R, et al. Social learning discrete particle swarm optimization based two-stage X-routing for IC design under intelligent edge computing architecture [J]. *Applied Soft Computing*, 2021 (104): 1-16.

[19] TRITSCHLER N, DUGENSKE A, KURFESS T. An automated edge computing-based condition health monitoring system: with an application on rolling element bearings [J]. *Journal of Manufacturing Science and Engineering*, 2021, 143 (7): 1-10.

[20] ZHOU W, JIE L I, LIU Z, et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning [J]. *Chinese Journal of Aeronautics*, 2022, 35 (7): 100-112.

[21] FIRDAUS M, RAHMADIK A S, RHEE K H. Decentralized trusted data sharing management on internet of vehicle edge computing (IoVEC) networks using consortium blockchain [J]. *Sensors*, 2021, 21 (2): 241-246.

[22] XIONG K, WEI C, ZHOU P. Integrated autonomous optical navigation using Q-Learning extended Kalman filter [J]. *Aircraft Engineering and Aerospace Technology*, 2022, 94 (6): 848-861.

(下转第 316 页)