

文章编号:1007-5321(2019)06-0064-06

DOI:10.13190/j.jbupt.2019-155

基于 DRL 的 MEC 任务卸载与资源调度算法

薛宁¹, 霍如^{1,2}, 曾诗钦³, 汪硕^{2,3}, 黄韬^{2,3}

(1. 北京工业大学 北京未来网络科技高精尖创新中心, 北京 100124; 2. 网络通信与安全紫金山实验室, 南京 211111;
3. 北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)

摘要: 为提高多接入边缘计算(MEC)任务卸载效率,提出了一个任务卸载和异构资源调度的联合优化模型. 考虑异构的通信资源和计算资源,联合最小化用户的设备能耗、任务执行时延和付费,并利用深度强化学习(DRL)算法对该模型求最优的任务卸载算法. 仿真结果表明,该优化算法比银行家算法的设备能耗、时延和付费的综合指标提升了27.6%.

关键词: 多接入边缘计算; 任务卸载; 异构资源调度; 深度强化学习

中图分类号: TP393

文献标志码: A

Tasks Offloading and Resource Scheduling Algorithm Based on Deep Reinforcement Learning in MEC

XUE Ning¹, HUO Ru^{1,2}, ZENG Shi-qing³, WANG Shuo^{2,3}, HUANG Tao^{2,3}

(1. Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing 100124, China;

2. Purple Mountain Laboratories, Nanjing 211111, China;

3. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In order to improve the task offloading efficiency in multi-access edge computing (MEC), a joint optimization model for task offloading and heterogeneous resource scheduling was proposed, considering the heterogeneous communication resources and computing resources, jointly minimizing the energy consumption of user equipment, task execution delay, and the payment. A deep reinforcement learning method is adopted in the model to obtain the optimal offloading algorithm. Simulations show that the proposed algorithm improves the comprehensive indexes of equipment energy consumption, delay, and payment by 27.6%, compared to the Banker's algorithm.

Key words: multi-access edge computing; task offloading; heterogeneous resource scheduling; deep reinforcement learning

多接入边缘计算(MEC, multi-access edge computing)将边缘计算的接入方式从电信移动网络进一步延伸至其他无线接入网络(如WiFi),并通过这项技术将网络控制、存储和服务推向网络边缘(例如基站和接入点)^[1],移动设备可以通过异构无线

网络接入 MEC 服务器,且接入侧都部署有计算资源,因此在对用户的计算任务进行卸载决策时需要综合考虑这些异构的通信和计算资源,优化资源调度,从而保证用户的使用体验^[2]. 近年来,一些研究工作对多用户 MEC 下的任务卸载策略及资源调度

收稿日期: 2019-07-11

基金项目: 国家自然科学基金项目(61902033); 未来网络操作系统发展战略研究(2019-XY-5)

作者简介: 薛宁(1994—),男,硕士生.

通信作者: 霍如(1988—),女,讲师, E-mail: huoru@bjut.edu.cn.

问题有许多研究成果. 一些研究提出一个基于博弈论的分布式任务卸载方案, 在优化能耗的同时使得用户间的博弈最终达到纳什均衡的状态, 但是需要每个用户频繁地从 MEC 服务器获取通信资源和计算资源的信息, 这将对移动设备形成很大的能耗负担^[3]. 另一些则考虑在多用户场景下利用分布式合作算法, 并结合无线电力、MEC 服务器资源分配和通信资源分配来减少用户的能耗和时延成本^[4-7]. 但是这些方式所造成的额外计算资源消耗和时延成本也很高.

在多用户场景下, MEC 边缘侧的资源状态变化对任务卸载决策有很大的影响, 可以通过将任务卸载决策导致的 MEC 边缘侧资源变化过程抽象为一个马尔可夫决策过程, 根据资源变化情况制定任务卸载策略, 任务卸载策略优化问题可以通过找到最佳的计算资源分配策略来解决, 但是该优化问题是非凸的, 并且 MEC 边缘侧资源的状态空间为无限大, 强化学习这种启发式算法对于解非凸优化问题具有快速、高效的特点, 同时利用深度神经网络可以很好地拟合资源环境的变化并学习强化学习作出的决策. 因此, 通过建立一个计算任务卸载和异构资源调度的联合优化模型, 利用深度强化学习 (DRL, deep reinforcement learning) 的方法最小化移动设备的能耗、任务执行时延及用户付费, 并综合考虑了 2 种异构的无线接入方式, 即移动网络和 WiFi, 这样可以为用户提供更合适的通信和计算资源, 同时该模型可充分利用神经网络的学习能力和泛化能力, 随着用户任务数量的增多, 算法生成的策略获得的收益将更高^[8].

1 系统模型

多接入边缘计算系统场景如图 1 所示, 用户设备可以将本地任务卸载到 MEC 服务器上运行来降低设备消耗^[9], MEC 服务器提供了移动网络和 WiFi 2 种无线接入方式, 并且接入侧分别有 V^{cell} 和 V^{wifi} 个虚拟机来为用户服务^[10]. 集合表示为 $V^{\text{cell}} = \{V_n^{\text{cell}} : V_n^{\text{cell}} = 1, 2, \dots, n\}$, $V^{\text{wifi}} = \{V_m^{\text{wifi}} : V_m^{\text{wifi}} = n+1, n+2, \dots, m\}$. 假设 MEC 服务器周围用户均有向 MEC 发送任务卸载请求, MEC 可选择拒绝任务卸载请求或选择一种异构资源为用户提供服务, 并且在一个任务卸载期间内, 信道质量保持稳定.

1.1 通信模型

MEC 服务器可以为接入侧周围的用户提供服

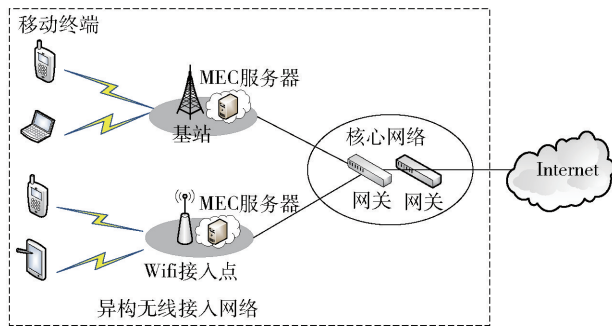


图1 多边缘计算系统场景

务, 这些有计算任务需要执行的用户用 $i (i = 1, 2, \dots, N)$ 表示, 且每个用户在当前时隙内只有 1 个任务需要执行, 用户 i 的任务表示为 X_i , 任务 X_i 的卸载决策表示为 a_i , 即任务是卸载到 MEC 服务器执行还是在本地执行. $a_i = 0$ 表示任务在本地设备执行, $a_i > 0$ 表示任务卸载到 MEC 服务器执行该任务. 任务决定卸载到 MEC 服务器执行时, 需要选择接入方式, 其中 $a_i \in V^{\text{cell}}$ 表示任务通过移动网络卸载到基站侧 MEC 服务器执行, $a_i \in V^{\text{wifi}}$ 表示任务通过 WiFi 卸载到 WiFi 接入点侧 MEC 服务器执行, 则用户卸载策略的集合为 $a_i = \{0\} \cup V^{\text{cell}} \cup V^{\text{wifi}}$. 卸载决策为 $a_i > 0$ 的用户传输任务的传输速率为

$$R_i(A) = \text{Clb} \left(1 + \frac{P_i H_i}{N_0 + \sum_{j \in I, j \neq i} P_j H_j} \right) \quad (1)$$

其中: P_i 为设备的传输功率, N_0 为信道内的高斯噪声功率, $C = \{C^{\text{cell}}, C^{\text{wifi}}\}$ 为接入网络提供的带宽 (接入移动网络时用 C^{cell} 表示, 接入 WiFi 时 C^{wifi} 用表示), H_i 为 MEC 服务器与用户 i 之间信道的信道增益参数, 从式 (1) 可以看出, 当多个用户使用同一信道时, 会使得数据的传输速率降低.

1.2 计算模型

假设 MEC 服务器接入侧周围的每个用户都有任务需要执行, 用户 i 的任务可表示为一个三元组 $X_i = \{D_{\text{in}}(i), Y(i), D_{\text{out}}(i)\}^{[11]}$, $D_{\text{in}}(i)$ 表示任务的输入数据量, 单位为 (kbit), $Y(i)$ 表示任务的负载量, 单位为 (cycle), $D_{\text{out}}(i)$ 表示任务的输出结果数据量, 单位为 kbit. 用户设备可通过应用调度图分析技术得到任务的三元组. 在获得任务三元组后便可建立多用户任务卸载的本地计算模型和异构资源计算模型.

1.2.1 本地计算模型

当任务 X_i 在本地执行时, 用 f_i^l 表示用户 i 的移动设备 CPU 频率, 单位为 (GHz), T_i^l 表示任务 X_i 在

本地的执行时间,单位为(ms),表示为

$$T_i^l = Y(i)/f_i^l \quad (2)$$

任务 X_i 在本地执行时, η_i 表示移动设备的计算能耗系数, E_i^l 表示任务 X_i 在本地的执行的移动设备能耗,单位为(J),表示为

$$E_i^l = \eta_i Y(i) \quad (3)$$

根据式(2)和式(3)可得出任务 X_i 在本地执行时的消耗为

$$W_i^l = \alpha_i^t T_i^l + \alpha_i^e E_i^l \quad (4)$$

s. t. $\alpha_i^t, \alpha_i^e \in [0, 1]$

α_i^t 表示时延的权值, α_i^e 表示能耗的权值,可以针对每个用户的需求来动态调整时延和能耗在总消耗中的比重,当用户任务对时延敏感时,可以调高 α_i^t 的值,当用户任务对能耗敏感时,则可以调高 α_i^e 的值。

1.2.2 异构资源计算模型

当任务 X_i 卸载到 MEC 服务器执行时,有 2 种异构资源可以提供服务:一种是通过移动网络接入 MEC 服务器端的虚拟机执行任务;另一种是通过 WiFi 网络接入 MEC 服务器端的虚拟机执行任务, $f^s = \{f^{\text{cell}}, f^{\text{wifi}}\}$ 表示 MEC 服务器 CPU 执行频率(移动网络侧的 MEC 服务器 CPU 执行频率用 f^{cell} 表示, WiFi 网络侧的 MEC 服务器 CPU 执行频率用 f^{wifi} 表示),单位为 GHz, T_i^s 表示任务 X_i 在 MEC 服务器的执行时间,单位为 ms。

$$T_i^s = Y(i)/f^s + (D_{\text{in}}(i) + D_{\text{out}}(i))/R(a) \quad (5)$$

当任务 X_i 卸载到 MEC 服务器执行时, P_i 表示移动设备的传输功率,移动设备的传输能耗为

$$E_i^s = P_i (D_{\text{in}}(i) + D_{\text{out}}(i))/R(a) \quad (6)$$

当 MEC 服务器端的虚拟机数量不足时,上传的任务将进入等待队列, T_i^{line} 表示任务 X_i 在 MEC 服务器的排队时间,表示为任务 X_i 之前执行的任务执行时间总和,单位为(ms), T_j^s 表示队列中在任务 X_i 之前执行的任务执行时间。

$$T_i^{\text{line}} = \sum_{j \in \text{queue}, j=a_i} T_j^s \quad (7)$$

当任务 X_i 卸载到 MEC 服务器执行时, P_{idle} 表示移动设备的传输功率,移动设备的空闲能耗为

$$E_i^s = P_i (D_{\text{in}}(i) + D_{\text{out}}(i))/R(a) \quad (8)$$

当任务 X_i 卸载到 MEC 服务器执行时,需要向选择的服务器付费, K_i 为用户 i 所选择异构资源的价格, $K_i \in \{l_1, l_2\}$, l_1 和 l_2 分别为移动网络接入侧资源和 WiFi 网络接入侧资源的价格。

$$K_i = \begin{cases} l_1, & a_i \in V^{\text{cell}} \\ l_2, & a_i \in V^{\text{wifi}} \end{cases} \quad (9)$$

综上所述,任务在 MEC 服务器执行时,任务 X_i 的能耗、时延、付费加权和为

$$W_i^s = \alpha_i^t (T_i^s + T_i^{\text{line}}) + \alpha_i^e E_i^s + \beta_i K_i \quad (10)$$

s. t. $\beta_i \in [0, 1]$

其中 β_i 表示用户对付费的敏感度。

2 MEC 任务卸载与异构资源调度

在这一节中,将利用深度强化学习的方法对计算任务卸载和无线资源调度进行联合优化,达到最小化移动设备的能耗、执行时延及用户付费的目的。

2.1 深度强化学习概述

强化学习(reinforcement learning)属于一种无监督的机器学习方法,在与环境交互中进行学习,通过策略迭代学习最大化累计回报来获得最优策略。强化学习中智能体(agent)与环境交互的过程可借助马尔可夫决策过程可形式化表述为 $\{S, A, P_a(s, s'), \gamma, R\}$, S 表示状态空间, A 表示动作空间, $P_a(s, s')$ 表示在动作 a 下状态的转移概率, $0 < \gamma < 1$ 表示折扣因子,表示 agent 的远视程度, R 表示即时回报函数。

agent 从 t_1 时刻的状态 s_t 采取动作 a_t 状态转移到 s_{t+1} 并获得即时奖励 r_t , 价值函数 $Q^\pi(s, a)$ 定义了策略 π 在状态 s 下作出动作 a 所获得的期望回报,表示为

$$Q^\pi(s, a) = E[r_t | s_t = s, a_t = a, \pi] \quad (11)$$

当策略 π 对于所有状态都能使期望回报最大时,策略 π 为最优策略,最优策略的价值函数为

$$Q^*(s, a) = \max_{\pi} E[r_t | s_t = s, a_t = a, \pi] \quad (12)$$

当状态和动作空间数量很大时,使用 Q 表存储状态动作对的概率是一种很低效的做法,因此使用深度神经网络拟合 Q 表来产生动作,使得相近的状态可以得到相近的动作,其模型的优化目标函数为

$$H(\theta) = E\{[r + \gamma \max_{a'} Q(s', a' | \theta) - Q(s, a | \theta)]^2\} \quad (13)$$

其中: γ 为折扣因子, θ 为神经网络模型的权重参数, θ 值的更新使用随机梯度下降算法。

在多用户 MEC 场景中,基于深度强化学习的 MEC 计算任务卸载与异构资源调度算法(DRL-COHRs)框架如图 2 所示。

2.2 状态空间

利用深度强化学习算法对系统环境进行建模的

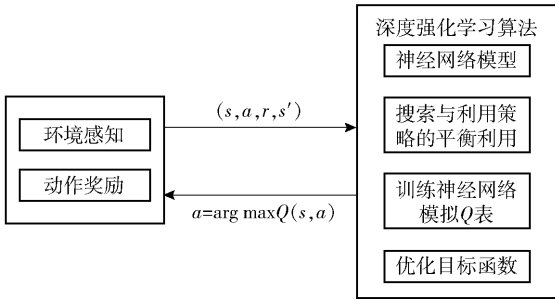


图2 深度强化学习算法框架

第1步是定义环境模型的状态空间,在定义中要同时考虑 MEC 服务器侧的资源使用状况和用户的接入方式. 环境模型的状态空间由移动网络带宽、WiFi 带宽和接入侧虚拟机数量组成,状态空间定义为

$$S = \{s_1^{\text{cell}}, s_2^{\text{cell}}, \dots, s_n^{\text{cell}}, s_{n+1}^{\text{wifi}}, \dots, s_m^{\text{wifi}}\} \quad (14)$$

s_i^{cell} 代表移动网络接入侧的虚拟机任务队列状态,该虚拟机的 CPU 频率为 f^{cell} , s_i^{wifi} 代表 WiFi 网络接入侧的虚拟机状态,该虚拟机 CPU 频率为 f^{wifi} .

2.3 决策时刻和动作空间

在 MEC 的场景下,卸载决策的时刻为当有新用户向 MEC 服务器发出卸载请求的时刻. 此时,卸载决策的动作空间可表示为

$$a = \{0, 1, 2, \dots, n, \dots, m\} \quad (15)$$

其中: $a_i = 0$ 表示拒绝卸载请求,任务在本地执行; $a_i > 0$ 表示接受卸载请求; $0 < a_i \leq n$ 表示任务通过移动网络卸载到 MEC 服务器; $n < a_i \leq m$ 表示任务通过 WiFi 卸载到 MEC 服务器,当动作为卸载到 MEC 服务器执行时,则 MEC 服务器会提供一定的带宽资源和计算资源来执行任务.

2.4 即时回报

即时回报是在某一系统状态下做出某一动作后环境给予的反馈值,表示在该状态下采取特定动作的好坏程度. 设定 $A = [a_1, a_2, \dots, a_N]$ 为所有用户的卸载策略的决策向量,目标是作出合理的决策向量最小化所有用户设备能耗、时延和付费:

$$\min_{A = [a_1, \dots, a_N]} \sum_{a_i=0} \alpha_i^l T_i^l + \alpha_i^e E_i^l + \sum_{a_j \neq 0} \alpha_j^l (T_j^s + T_j^{\text{line}}) + \alpha_j^e E_j^s + \beta_j K_j \quad (16)$$

其中:用户对于能耗、时延和付费设定的权重因子分别为 α_i^l 、 α_i^e 和 β_i ,结合式(4)和式(9)奖励函数建模为

$$r(S, a_i) = \begin{cases} -W_i^l, & a_i = 0 \\ -W_i^s, & a_i \neq 0 \end{cases} \quad (17)$$

强化学习算法的目标是最大化累计折扣回报的期望,即最小化所有用户设备能耗、时延和付费的加权和,表示为 $\max E \left[\sum_{i=0}^N \gamma^i r_i \right]$, γ^i 为折扣因子,表示对每个任务卸载决策的即时回报的权重,设置为1,表示每个任务的即时回报对总回报的贡献权值是一样的.

2.5 基于深度强化学习的 MEC 计算任务卸载与异构资源调度方法

基于深度强化学习的 MEC 计算任务卸载与异构资源调度方法(DRL-COHRs)的训练过程如算法1所示: S 为状态空间, a 为动作空间, θ 为神经网络的参数, $\text{rand}()$ 为随机生成探索新策略的概率, ε 为接受探索新策略的阈值. $\text{rand}(a)$ 在动作空间 a 中随机选择一个动作, $\text{env}(a_i)$ 表示环境对执行动作 a_i 后返回的即时奖励和新的状态.

算法1 基于深度强化学习的 MEC 计算任务卸载与异构资源调度方法实现

- 1 初始化深度神经网络 Q , 权重参数 θ
- 2 初始化深度神经网络 Q' , 权重参数 $\theta' = \theta$
- 3 初始化样本存储池 D
- 4 for episode = 1, M do
- 5 初始化深度神经网络输入 S_1 , 即当前接入侧的虚拟机任务队列状态, 并通过深度神经网络输出一组卸载策略 A
- 6 for $t = 1, N$ do
- 7 if $\text{rand}() > \varepsilon$ then
- 8 $a_t = \text{rand}(a)$
- 9 else
- 10 $a_t = \arg \max_a Q(S_t, a | \theta)$
- 11 $S_{t+1}, r_t \leftarrow \text{env}(a_t)$ // 执行动作 a 后, 返回环境所给的奖励
- 12 存储 (S_t, a_t, r_t, S_{t+1}) 到 D 中
- 13 随机采样 D 中 minibatch 个样本 (S_j, a_j, r_j, S_{j+1})
- 14 $y_j = \begin{cases} r_j, & j = N \\ r_j + \sum_{i=j+1}^N \gamma^{i-j} \max_{a'} Q'(S_i, a'_i | \theta'), & j < N \end{cases}$
- 15 以 $(y_j - Q(S_j, a_j | \theta))^2$ 作为损失函数通过 SGD 更新 θ
- 16 每隔 C 步更新 Q 网络参数为当前网络参数
- 17 end for

18 end for

3 仿真结果与分析

通过仿真对 DRL-COHS 算法与分布式协作算法(DCCO,distributed cooperative computation offloading)^[6]、银行家算法(Bank)、优化控制算法(OACP,optimal admission control policy)^[7]进行了实验对比,仿真参数如表 1 所示.

表 1 实验参数表

参数	分布
$D_{in}(i)/\text{kbit}$	$\text{Normal}([5,10])$
$Y(i)/\text{cycle}$	$\text{Normal}([150,300])$
$D_{out}(i)/\text{kbit}$	$\text{Normal}([2.5,5])$
η_i	$\text{Normal}([0.5,0.5])$
N_0/dBm	-120
P_i/dBm	$\text{Normal}([150,300])$
f_i^l/GHz	$\text{Normal}([15,30])$
$C^{\text{cell}}/(\text{Mbit}\cdot\text{s}^{-1})$	100
$C^{\text{wifi}}/(\text{Mbit}\cdot\text{s}^{-1})$	500
V^{cell}	10
V^{wifi}	10
$l_1/\text{元}$	30
$l_2/\text{元}$	40

Q 表的拟合函数为包含 2 个隐藏层的全连接神经网络,输入层的维度为 7,2 个隐藏层的神经元个数分别为 256 和 128,输出层的维度为 m ,使用 Relu 函数作为激活函数,并使用 L2 正则化来防止模型过拟合. 训练时采用策略迭代的方式进行训练,仿真实验中动作的选择策略为 ε -greedy 策略. 在训练过程中, ε 初始值为 1,每迭代 50 次 ε 减小 0.1,降低至 0.1 后保持不变,一直以 10% 的概率进行探索.

深度强化学习中拟合 Q 表的神经网络模型是后续仿真实验的基础,经过训练才能更好地拟合环境给出的回报及最优策略. 根据表 1 中任务参数生成 1 500 个任务,分为 10 个批次,其中 8 个批次作为训练集训练模型,2 个批次作为测试集来验证模型的性能. 同时在训练集上分别训练能耗、时延和费用的权重依次为 $\{0.1,0.5,0.9\}$ 的 DRL-COHS 算法模型. 图 3 为训练过程,可以看出,在训练初期,由于动作的随机选择概率 ε 比较大,agent 处于探索

学习阶段,总回报值较低,随着迭代次数的增加,agent 由探索学习状态转为利用经验状态,算法快速收敛,总回报值趋于稳定.

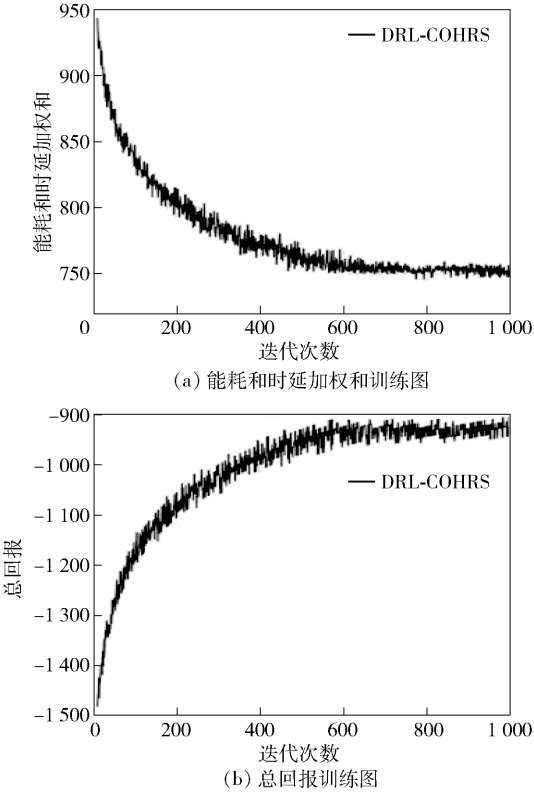
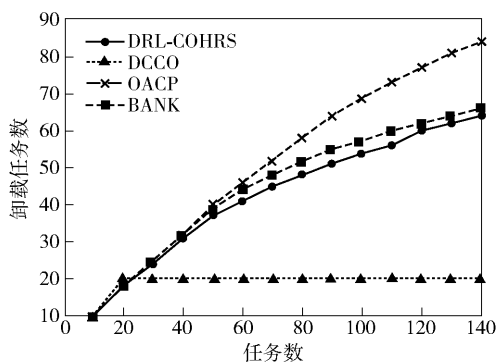


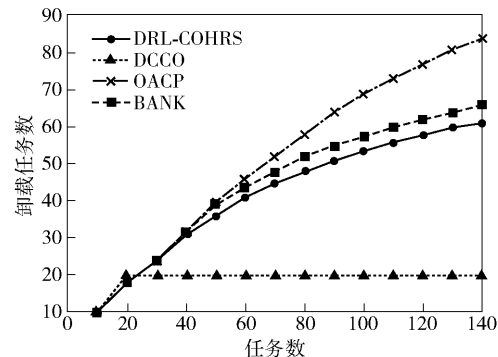
图 3 能耗和时延加权、总回报随迭代次数变化

图 4 所示分别为训练集和测试集中不同算法卸载任务数,用户选择的能耗、时延和费用的权重从集合 $\{0.1,0.5,0.9\}$ 中随机选择. 从图中看出,随着任务数的增加,OACP 算法和 DRL-COHS 算法的卸载任务数都在增加,Bank 算法在任务数大于 20 后不再增加,这是因为移动网络和 WiFi 网络接入侧的虚拟机总数为 20,当任务数大于 20 时,Bank 算法便不再进行卸载. OACP 算法、DCCO 算法和 DRL-COHS 算法随着任务数的增加,卸载任务数占总任务数的比例在下降,这是由于任务数增加后,信道干扰增大,信道传输速率下降,传输消耗增大,导致在本地执行任务时总消耗更低. DRL-COHS 算法相比于 OACP 算法、DCCO 算法,不仅考虑了用户信道之间的干扰,还考虑了等待时延和空闲能耗,当任务数量增加时等待时延和空闲能耗都会增大,所以选择卸载的任务数量会比 OACP 算法、DCCO 算法少.

图 5 所示为训练集和测试集中不同算法用户的平均回报,用户选择的能耗、时延和费用的权重从集合 $\{0.1,0.5,0.9\}$ 中随机选择. 从图中看出,Bank



(a) 训练集上不同算法卸载任务数



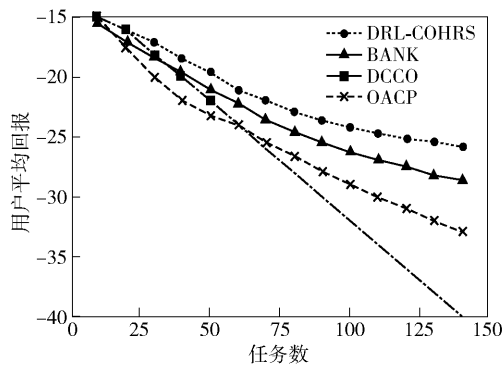
(b) 测试集上不同算法卸载任务数

图4 训练集、测试集上不同算法卸载任务数

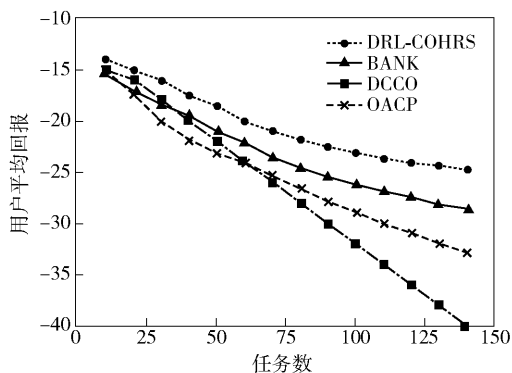
算法的平均回报在任务数大于20后呈现线性下降,这是由于大量任务在本地执行导致时延和能耗增加从而导致回报线性下降.其他3个算法的平均回报都随着任务数的增加而降低,这是由于任务数量的增加引起传输能耗和等待时延的增加.由于OACP算法只针对当前一个任务进行优化,有陷入局部最优解的问题,导致平均回报低于DRL-COHS算法.DCCO算法没有对用户付费进行优化,所以整体回报也低于DRL-COHS算法.结合图4可以看出,DRL-COHS算法可以自适应任务的数量及MEC服务器状态,为用户选择合适的卸载策略,有效降低能耗、时延和付费.

4 结束语

提出了一种基于深度强化学习的MEC任务卸载与异构资源优化算法,目标是最大化设备能耗、时延和付费的综合奖励指标,建立了一个联合计算任务卸载和资源调度的优化模型,并在此基础上将MEC边缘侧资源变化过程构建为一个马尔可夫过程决策过程,然后利用深度强化学习对MEC系统环境进行学习,并不断优化卸载决策,最后利用生成的模型对任务卸载作出决策.仿真结果表明,DRL-



(a) 训练集不同算法用户平均回报



(b) 测试集不同算法用户平均回报

图5 训练集、测试集不同算法用户平均回报

COHS算法对MEC场景下计算任务的卸载决策是非常有效的,使设备能耗、时延和付费的综合指标得到了大幅度的提高.未来的研究工作主要包括加入对移动设备无线信道传输功率的控制及用户切换异构资源的考虑等.

参考文献:

- [1] Mao Yuyi, You Changsheng, Zhang Jun, et al. A survey on mobile edge computing: the communication perspective [J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.
- [2] Huang D, Wang P, Niyato D. A dynamic offloading algorithm for mobile computing [J]. IEEE Transactions on Wireless Communications, 2012, 11(6): 1991-1995.
- [3] Chen Xu, Jiao Lei, Li Wenzhong, et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. ACM Transactions on Networking, 2016, 24(5): 2795-2808.
- [4] Ebrahimzadeh A, Maier M. Distributed cooperative computation offloading in multi-access edge computing fiber-wireless networks [J]. Optics Communications, 2019, 452: 130-139.

- ACM Press, 2000: 376-383.
- [5] Karimi P, Jazayeri R. Comparing the fault diagnosis performances of single neural network and two ensemble neural networks based on the boosting methods[J]. Journal of Automation and Control, 2014, 2(1): 21-32.
- [6] Chien Chenfu, Chang Kuohao, Wang W C. An empirical study of design-of-experiment data mining for yield-loss diagnosis for semiconductor manufacturing[J]. Journal of Intelligent Manufacturing, 2014, 25(5): 961-972.
- [7] Chien Chenfu, Liu C W, Chuang S C. Analysing semiconductor manufacturing big data for root cause detection of excursion for yield enhancement [J]. International Journal of Production Research, 2017, 55(17): 5095-5107.
- [8] Chen Y J, Wang Bocheng, Wu J Z, et al. Big data analytic for multivariate fault detection and classification in semiconductor manufacturing[C]//2017 13th IEEE Conference on Automation Science and Engineering (CASE). New York: IEEE Press, 2017: 731-736.
- [9] Khakifirooz Marzieh, Chien Chenfu, Chen Yingjen. Bayesian inference for mining semiconductor manufacturing big data for yield enhancement and smart production to empower industry 4.0 [J]. Applied Soft Computing, 2018(68): 990-999.

(上接第 69 页)

- [5] Wang Feng, Xu Jie, Wang Xin, et al. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[J]. IEEE Transactions on Wireless Communications, 2017, 17(3): 1784-1797.
- [6] Chen Min, Hao Yixue. 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.
- [7] Chen M H, Liang Ben, Dong Min. Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point[C]//IEEE INFOCOM 2017 IEEE Conference on Computer Communications. Atlanta, GA: IEEE Press, 2017: 1-9.
- [8] 孟浩, 霍如, 郭倩影, 等. 基于机器学习的 MEC 随机任务迁移算法[J]. 北京邮电大学学报, 2019, 42(2): 25-30.
- Meng Hao, Huo Ru, Guo Qianying, et al. Machine learning-based stochastic task offloading algorithm in mobile-edge computing[J]. Journal of Beijing University of Posts and Telecommunications, 2019, 42(2): 25-30.
- [9] Guo Songtao, Xiao Bin, Yang Yuanyuan, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//IEEE INFOCOM 2016 The 35th Annual IEEE International Conference on Computer Communications. San Francisco, CA: IEEE Press, 2016: 1-9.
- [10] Yang L, Cao J, Cheng H, et al. Multi-user computation partitioning for latency sensitive mobile cloud applications[J]. IEEE Transactions on Computers, 2014, 64(8): 2253-2266.
- [11] Yu Yingha, Zhang Jun, Letaief K B. Joint subcarrier and CPU time allocation for mobile edge computing[C]//2016 IEEE Global Communications Conference (GLOBECOM). Washington DC: IEEE Press, 2016: 1-6.