

基于能量收集技术的协作卸载计算方案

王珺, 赵浩东

(南京邮电大学通信与信息工程学院, 江苏 南京 210003)

摘要: 近年来, 物联网 (IoT) 应用对设备可使用的能量要求不断提高, 能量收集 (EH) 技术成为缓解边缘计算中设备能量短缺问题并延长电池寿命的重要途径。然而, 当环境中可再生能源不充足时, 设备电量耗尽会导致任务中断, 影响物联网性能。为了解决这一问题, 提出了一种联合能量收集和设备间 (D2D) 通信技术的任务卸载框架, 采用基于深度强化学习 (DRL) 的边缘协作卸载计算方案, 自主进行决策并使用模拟退火算法解决资源分配问题, 以最小化系统运行总成本。对稳定和极端两种能量环境进行仿真, 结果表明, 该方案在单用户多设备场景下可稳定运行且具有成本效益。

关键词: 边缘计算; 能量收集; 设备间通信; 深度强化学习

中图分类号: TP391

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2024.00361

Collaborative offloading computing scheme based on energy harvesting technology

WANG Jun, ZHAO Haodong

Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: In recent years, the energy requirements for devices in internet of things (IoT) applications have increased, making energy harvesting (EH) technology an important way to alleviate the energy shortage problem in edge computing and extend the battery life of devices. However, when there was insufficient renewable energy in the environment, the depletion of device power can cause task interruption and affect the performance of IoT. To solve this problem, a task offloading framework that combined energy harvesting and device-to-device (D2D) communication technology was proposed, using a deep reinforcement learning (DRL)-based edge collaborative offloading computing scheme to make autonomous decisions and solve resource allocation problems using simulated annealing algorithms to minimize the total cost of system operation. Simulation results on stable and extreme energy environments show that the proposed scheme can run stably and cost-effectively in single-user multiple-device scenarios.

Key words: edge computing, energy harvesting, device-to-device communication, deep reinforcement learning

0 引言

近年来随着物联网 (IoT, internet of things) 的发展, 移动设备的数量爆炸式增长, 同时人脸识

别、虚拟现实和增强现实、车联网、智慧城市等新的应用程序成为主流业务。传统的集中式云计算模式已无法支持这些数量庞大的计算密集型应用程序的需求, 在此背景下, 基于分布式的移动边缘计

收稿日期: 2023-03-27; 修回日期: 2024-06-15

通信作者: 王珺, wangjun@njupt.edu.cn

基金项目: 江苏省研究生科研与实践创新计划项目 (No. 46006CX21732); 江苏省重点研发计划 (No. BE2020084-5)

Foundation Items: Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. 46006CX21732), Jiangsu Provincial Key Research and Development Program (No. BE2020084-5)

算 (MEC, mobile edge computing) 应运而生^[1]。

边缘计算^[2]提供了在本地处理或存储关键数据的能力, 并将所有接收到的数据推送到中央数据中心或云存储库, 有效地缓解了云和用户设备的计算压力。由于物联网中的设备的电池容量有限, 当电池能量耗尽时, 计算就会中断, 设备能耗问题是亟须解决的难题。针对供电困难的应用场景, 如节点难以重新部署的无线传感器网络、要求高密封性与频繁消毒的机器人、供电昂贵的偏远地区的应用系统, 能量收集 (EH, energy harvesting) 技术通过收集可再生能源 (如风能、热能、太阳能等绿色能源), 为移动设备提供系统正常运行所需的能源保障和低能源成本, 在一定程度上缓解能源短缺问题^[3-6]。

除此之外, D2D 技术利用设备与设备之间的高效通信, 在不经基站的情况下运行任务。使用该技术, 可以令高电量设备分担低电量设备的负载, 在均衡网络负载的同时, 还能维持系统设备电量的均衡^[7-8]。在 MEC 系统中, 联合 EH 技术与 D2D 技术, 可以有效缓解绿色能源供电设备的能耗压力, 实现设备的可持续运行与绿色计算^[9]。

近年来, EH 技术在 MEC 系统中得到了广泛的应用^[10-11]。文献[12-14]中, 给定能量收集以及信道增益的先验知识, 研究基于 EH 技术的多用户 MEC 系统下的最优卸载策略; 文献[15-17]中, 提出基于执行成本最小化的高维马尔可夫决策问题和基于李雅普诺夫的动态计算卸载算法, 卸载决策与资源分配联合决策, 旨在降低能耗成本; 文献[18]中, 提出了一种考虑任务间依赖性和动态能量收集的模型, 提出贪婪算法将具有任务依赖性的子任务优先卸载到完成时间最短的位置, 再采用模拟退火算法求解任务完成时间最小化问题。文献[19-22]中, 使用强化学习算法来解决时延、能效与卸载速率等优化问题, 依据马尔可夫决策过程, 将执行过程分为若干时隙, 每一时隙根据状态空间进行卸载决策, 并将状态动作对记录在 Q 值表或神经网络中, 以此学习最优的卸载决策, 通过使用基于强化学习的在线卸载框架解决二进制卸载决策中的卸载速率最优问题。上述文献虽然在具有 EH 技术的 MEC 系统上进行了大量的工作, 然而研究仍存在一些局限性, 文献[12-14]提出的算法无法适用动态的环境, 文献[15-16, 18-22]的研究重点在于能量收集量充足情

况下, MEC 系统中的时延、卸载速率、能量收集效率以及网络学习收敛速度的优化, 这些研究并不适用于因多变天气影响, 无法收集充足可再生能源的情况。针对供电困难场景, 即在能量收集量无法满足设备基本供电需求的情况下, 可再生能源供电设备在动态的网络环境中, 如何采取合适的计算卸载决策与资源分配方案, 在维持电量可持续运行的前提下实现系统总成本的优化。

本文的主要贡献包括如下 3 个方面。

1) 本文设计了一种针对单用户多设备的绿色任务卸载框架。引入 D2D 技术, 在 IoT 设备周围部署有线供电的辅助设备, 为 IoT 设备提供除本地处理、卸载到边缘服务器外的第 3 种卸载选择, 在网络环境不佳的情况下辅助计算, 在节约成本的同时降低 IoT 设备的能耗, 维持重要设备的可持续运行。同时联合 D2D 技术, 为设备设计一种激励机制, 以设备电量作为参考基准, 约束 IoT 设备与辅助设备之间的任务卸载。

2) 针对本文提出的优化问题, 设计了基于 EH-D2D 的 DQN (EDDQN) 算法。首先, 设计一种基于深度强化学习的集中式的任务卸载算法, 在传统状态空间中添加设备电量参数, 考虑设备电量变化对动作空间的影响。然后, 设计了资源分配方案, 对每个时间片各个设备已知的卸载决策, 采用模拟退火算法计算出最佳传输功率与计算资源大小。EDDQN 算法仅依赖当前系统的状态信息即可求得近似最优解。

3) 针对新的应用, 即无法从环境中获取充足的可再生能源且有线供电困难的应用场景, 在极端能量环境下进行仿真。通过搭建仿真平台对 EDDQN 算法进行性能分析, 对比多种经典算法, 从总成本、设备能耗以及任务丢弃率 3 方面验证该算法的性能。仿真结果显示, 与传统算法相比, 本文算法在稳定和极端的能量环境下均能有效降低系统能耗, 同时保持较低的成本和任务丢弃率。

1 系统模型

本文考虑的系统架构如图 1 所示, 由一个 MEC 服务器、一组 IoT 设备 $\{\text{Dev}_1, \text{Dev}_2, \dots, \text{Dev}_N\}$ 和一组辅助设备 $\{\text{Sup}_1, \text{Sup}_2, \dots, \text{Sup}_N\}$ 组成, 设备数表示为 $n \in \{1, 2, \dots, N\}$ 。其中, IoT 设备部署于难以供电的场所, 使用 EH 技术收集可再生能源作为

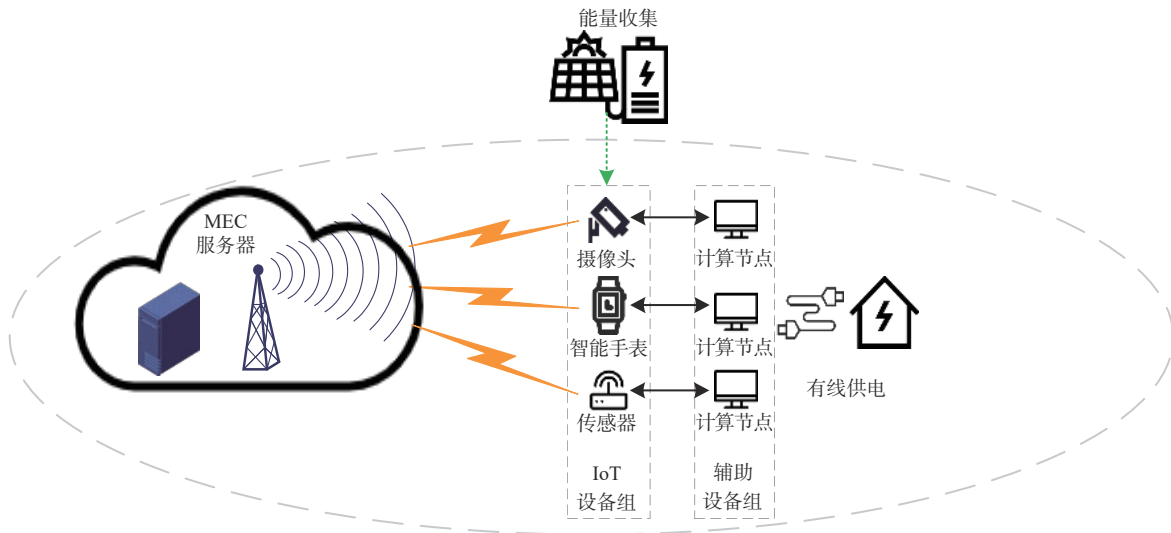


图1 系统架构

IoT设备的唯一供电来源，而辅助设备部署于IoT设备附近方便布线从而进行有线供电的场所，辅助IoT计算任务，减少IoT设备的能耗。本系统仅适用于与单用户相关的应用场景，系统中的所有设备都属于一个用户（或一个组织），如智慧家居的家庭设备组、智慧工业的工厂设备组、智慧农业中的传感设备组，其中每个IoT设备与基站进行蜂窝通信，与相邻的辅助设备进行D2D链路通信。系统采用离散时间模型，时间被分为若干时隙，表示为 $t \in \{1, 2, \dots, T\}$ ，假设时隙长度远小于信道相干时间，在每个时隙内，系统状态保持不变。本文考虑在每个时隙对IoT设备指定任务卸载策略。本文主要符号及其意义见表1。

1.1 通信模型

IoT设备一般通过无线链路向服务器或辅助设备传输数据，传输速率受传输环境和设备分配资源影响，如信道增益、信道带宽和传输功率。

在D2D链路通信中采用专用短程通信（DSRC, dedicated short range communication）^[23]，实现设备之间小范围内图像、语音和数据的实时、准确和可靠的双向传输，而在蜂窝链路通信中，采用频分多址（FDMA, frequency division multiple access）传输任务，将 t 时隙时，IoT设备与辅助设备或边缘服务器之间的链路传输速率表示为

$$R_n^t = B \log \left(1 + \frac{h_n^t p_o_n^t}{\sigma^2} \right) \quad (1)$$

其中， h_n^t 表示IoT设备 n 与辅助设备或MEC服务器之间的信道增益， σ^2 表示高斯噪声功率， $p_o_n^t$ 表示

表1 主要符号及意义

符号	意义
N	总设备数
T	总时隙数
h^t	信道增益
σ^2	噪声功率
$p_o_n^t$	IoT设备传输功率
B	平均分配带宽
c^t	t 时隙收集到的能量
b^t	t 时隙设备剩余电量
e^t	t 时隙消耗的总能量
b_{\max}	电池存储容量
B^t	电池能级
ζ	CPU芯片系数
W^t	执行任务所需CPU周期数
f^t	CPU频率
D^t	任务大小
p_l^t	本地执行能量单价
p_r^t	MEC卸载能量单价
p_d^t	D2D卸载能量单价

IoT设备传输功率。本文不考虑带宽的动态分配，将总带宽平均分配，此时每个设备分配到的带宽为 B 。

1.2 能量收集模型

在可再生能源收集技术中，设备能够收集到的能量量受多种因素影响，如可再生能源类型、天气条件、设备效率等。为每个IoT设备配置能量收集设备，用于收集太阳能、风能混合能源。可收集能源变化情况参考文献[24]给出的真实世界绿色能源收集分布轨迹，将一天可收集的能源量分为3种状态——低、中、高，见表2。

表 2 一天内能量收集分布

时间段	能源分布状态	能量量
18:00—0:00	指数分布	低
0:00—6:00		
6:00—9:00	正态分布	中
15:00—18:00		
9:00—15:00	正态分布	高

由于可再生能源的随机性和间歇性特征，只有在每一时隙的开始处观察到收集的能量等级。定义 c'_n 是设备 n 在 t 时隙内收获的能源， b'_n 是设备 n 在当前时隙的电池电量， e'_n 是设备 n 在当前时隙设备消耗的能量。由于电池存储容量有限，收集能量量不可以超过设备电池存储容量上限。电池状态演变如下

$$b_{n,t+1} = \min [\max (b_{n,t} + c'_{n,t} - e'_{n,t}, 0), b_{\max}] \quad (2)$$

为了方便研究，本文将电池电量分为 11 个能级，用 $B'_n \in \{0, 1, 2, \dots, 10\}$ 表示，其中 $B'_n = 10$ 表示电量达到存储容量上限， $B'_n = 0$ 表示电池存储容量为 0。

1.3 计算模型

每个计算任务有 3 种选择，分别为本地计算、MEC 卸载以及 D2D 卸载。将 $A'_{n,p} \in \{0, 1\}$ 表示为设备的决策状态， $p \in \{l, r, d\}$ 表示 3 种卸载决策。其中， $A'_{n,l} = 1$ 表示本地计算， $A'_{n,r} = 1$ 表示卸载到 MEC 服务器， $A'_{n,d} = 1$ 表示设备将任务卸载到匹配的辅助设备上。

1) 当 $A'_{n,l} = 1$ 时，设备 n 在第 t 个时隙选择本地执行任务，使用动态电压频率调整 (DVFS, dynamic voltage and frequency scaling) 技术^[13]，动态调整每个时隙的设备 CPU 频率，从而控制任务执行的能耗和时延。此时，本地执行时延和能耗分别表示为

$$l'_{n,l} = \frac{W'_n}{f'_{n,l}} \quad (3)$$

$$e'_{n,l} = \zeta W'_n (f'_{n,l})^2 \quad (4)$$

其中， ζ 表示设备 CPU 芯片系数， W'_n 表示执行任务所需的 CPU 周期数， $f'_{n,l}$ 表示设备 n 在 t 时隙使用的本地设备 CPU 频率。

2) 当 $A'_{n,r} = 1$ 时，设备将任务卸载到 MEC 服务器。此时，卸载执行时延、传输能耗和计算能耗分别表示为

$$l'_{n,r} = \frac{D'_n}{R'_n} \quad (5)$$

$$e'_{n,r} = \frac{D'_n \rho \alpha'_n}{R'_n} \quad (6)$$

$$e'_{n,r} = \zeta W'_n (f'_{n,r})^2 \quad (7)$$

式(5)、式(6)中， D'_n 表示任务数据量，它与式(7)中计算任务所需的周期数 W'_n 为线性关系，两者可等价替换。执行时延包含计算时延、传输时延以及排队时延。假设 MEC 服务器的计算能力远高于本地设备，其计算时延和排队时延可忽略不计。同时，由于返回结果的数据大小远小于传输的任务数据大小，返回结果的时延也可以忽略不计。因此，MEC 卸载的执行时延仅由传输时延组成。此外，卸载传输能耗 $e'_{n,d}$ 与计算能耗 $e'_{n,r}$ 分别由本地设备和 MEC 服务器承担， $f'_{n,r}$ 表示设备 n 在 t 时隙使用的 MEC 服务器 CPU 频率。

3) 当 $A'_{n,d} = 1$ 时，设备将任务卸载到辅助设备，由于设备的计算能力远不如 MEC 服务器，除了传输时延，还需要考虑在辅助设备上的计算时延，因此，本地传输时延和辅助设备计算时延之和才是 D2D 卸载的执行时延。此时，辅助设备的 D2D 卸载执行时延、传输能耗和计算能耗分别表示为

$$l'_{n,d} = \frac{W'_n}{f'_{n,d}} + \frac{D'_n}{R'_n} \quad (8)$$

$$e'_{n,d} = \frac{D'_n \rho \alpha'_n}{R'_n} \quad (9)$$

$$e'_{n,d} = \zeta W'_n (f'_{n,d})^2 \quad (10)$$

同样地，D2D 卸载产生的传输能耗 $e'_{n,d}$ 和执行能耗 $e'_{n,d}$ 分别由本地设备和辅助设备承担。其中， $f'_{n,d}$ 表示设备 n 在 t 时隙使用的辅助设备 CPU 频率。

1.4 激励机制设计

在 D2D 通信中，如果不设置激励约束，IoT 设备会将大量的任务卸载到辅助设备，造成计算资源的滥用。为了防止辅助设备的资源被过度使用，本文设计了一种激励机制，通过设置能量单价，并对本地与辅助设备的电量能级以及当前时隙的任务能耗，从而约束设备的卸载决策。

首先，本文引入能量单价这一因素，系统管理者（如所有设备的拥有者）根据设备电量的获取难度以及设备的重要程度，为不同设备主观地设置不同价位的能量单价，本地执行、MEC 卸载和 D2D 卸载的能量单价分别表示为 $p'_{n,l}$ 、 $p'_{n,r}$ 、 $p'_{n,d}$ 。由于服务器已经安置妥当，能通过接线或燃烧燃料的方式供电，能量单价在三者中最小，而辅助设备只提供自己的计算能力，在重要程度上不及功能与计算兼有的 IoT 设备，因此 IoT 设备的能量单价设置为最大。在电量充足时，三者能量单价从小到大表示为

$p'_{n,r} < p'_{n,d} < p'_{n,l}$ ，其中，能量单价越高代表设备的电量越珍贵。

同时IoT设备能量单价不是固定不变的，定义了价格上涨规则，本文将IoT设备比作顾客，商品为供不应求的设备电量，设备电量充足时价格稳定，当设备自身电量不足时供电价格上涨，IoT设备能量单价与电池能级成反比，表示为

$$p'_{n,l} \propto \frac{1}{B'_n} \longrightarrow p'_{n,l}(B'_n) = b - kB'_n \quad (11)$$

其中， b 为截距，表示设备电量不足时的能量单价； k 为斜率，表示能量单价变化快慢。除IoT设备外，服务器与辅助设备因为存在稳定的供电来源，它们的能量单价都保持稳定不变。随后，使用能耗与能量单价的乘积来表示设备 n 选择决策 p 的“能耗开销 (Overhead)”，符号为 $O'_{n,p}$ 。

$$O'_{n,l} = e'_{n,l} p'_{n,l}(B'_n) \quad (12)$$

$$O'_{n,r} = e'_{n,l} p'_{n,l} + e'_{n,r} p'_{n,r} \quad (13)$$

$$O'_{n,d} = e'_{n,l} p'_{n,l} + e'_{n,d} p'_{n,d} \quad (14)$$

其中，式(12)表示本地执行的能耗开销；式(13)表示MEC卸载的能耗开销，由设备传输能耗开销与卸载给MEC计算所支付的能耗开销组成， $e'_{n,l}$ 、 $e'_{n,r}$ 分别表示卸载传输能耗和卸载计算能耗；式(14)表示D2D卸载的能耗开销，由设备传输能耗开销与卸载给辅助设备计算所支付的能耗开销组成， $e'_{n,l}$ 、 $e'_{n,d}$ 分别表示D2D卸载传输能耗和D2D卸载计算能耗。

通过式(12)~式(14)，获得了当前时隙3种决策的能耗开销。本文从设备拥有者的角度考虑，当设备向MEC服务器或辅助设备卸载任务时需要支付一定的成本作为借用设备的代价，设备在进行卸载决策时不仅需要考虑自身的时延能耗变化情况，还要根据能耗和能量单价考虑卸载后对其他设备的影响，这种影响通过能量开销的形式表示出来。

1.5 优化模型

引入激励机制对优化问题进行改进，从传统的时延与能耗加权和改进为时延与“能耗开销”加权和。

首先定义设备 n 选择一种卸载决策的“成本”为时延与能耗开销加权和，表示为

$$C'_{n,p} = \alpha I'_{n,p} + \beta O'_{n,p} + \phi \cdot 1(I'_{n,\text{drop}}) \quad (15)$$

其中， α 和 β 分别表示时延和能耗开销的加权因子， $I'_{n,p}$ 表示采用其中一种卸载决策的时延， $O'_{n,p}$ 表示采用其中一种卸载决策的能耗开销， $1(\cdot)$ 为决策函数，

当内部逻辑为true时， $1(\cdot) = 1$ ；反之， $1(\cdot) = 0$ 。

$I'_{n,\text{drop}} = \text{true}$ 表示设备 n 将执行时延超过时延容忍上限的任务以及因电量不足无法执行的任务丢弃。本文添加任务丢弃成本 ϕ 并给予成本惩罚，设置较高的成本惩罚有助于卸载决策时排除更多的不佳选项，并在仿真分析中获得明确对比，具体参数在后续仿真分析中设置。根据式(15)，本文将整个系统的目标函数表示为

$$\begin{aligned} P1: \min \lim_{A'_n, p'_n, T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N (A'_{n,l} C'_{n,l} + A'_{n,r} C'_{n,r} + A'_{n,d} C'_{n,d}) \\ \text{s.t.} \quad & C1: I'_n \leq L_{\max} \\ & C2: 0 \leq f'_n \leq f_{\max} \\ & C3: 0 \leq b'_n \leq b_{\max} \\ & C4: 0 \leq \text{po}'_n \leq \text{po}_{\max} \\ & C5: A'_{n,l} + A'_{n,r} + A'_{n,d} = 1 \end{aligned} \quad (16)$$

其中，C1表示时延约束，所有设备执行时延不超过任务的容忍时延 L_{\max} ；C2表示计算资源约束，对于使用DVFS技术的设备用于处理任务的计算能力不超过设备计算能力上限 f_{\max} ；C3表示电量约束，设备电量不超过储存电量上限 b_{\max} ；C4表示设备传输功率不超过传输功率上限 po_{\max} ；C5表示卸载决策为3种选择，本地执行、MEC卸载和D2D卸载，在同一时刻同一个设备只能选择其中一种卸载决策。

2 基于EH-D2D的DQN算法

2.1 EDDQN算法设计

传统应用场景中，可以收集太阳能、风能混合能源为设备供能，在能量环境不够稳定（多云无风）的情况下，短时间内切换到替代的不可再生能源供电，以保证设备的可持续运行。

然而，在难以供电的应用场景，设备只能依靠可再生能源供电，这种情况下，短时间内比较恶劣的能量环境就会造成设备的能源供应短缺以及任务中断。

本文联合EH技术与D2D技术，设计了一种基于EH-D2D的DQN算法 (EDDQN)，用于优化两种场景（稳定能量环境和恶劣能量环境）下的系统成本问题。EDDQN算法将电量能级也纳入状态空间，通过激励机制影响卸载决策。由于辅助设备的存在，将二元动作扩展为三元，即本地执行、MEC卸载和D2D卸载。EDDQN算法分为两个部分，一部分使用DQN算法求解最优卸载决策，另

一部分使用模拟退火算法求解最优资源分配问题。EDDQN算法流程如图2所示。

2.2 DQN算法求解最优卸载决策

Q学习(Q-learning)的核心思想是智能体agent通过状态-动作价值函数 $Q(s, a)$ 评价状态 s 下执行动作 a 的好坏程度,然后将这些评价记录在一张 Q 值表中,并通过贝尔曼方程理论^[25]更新 Q 值

$$Q^*(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (17)$$

通过在 Q 值表中存储大量的记录,针对不同的状态 s ,智能体都可以通过查询 Q 值表获得最优决策 $\pi^*(s)$

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a) \quad (18)$$

为了防止智能体出现探索-利用窘境(exploration-exploitation dilemma)问题,即在学习初期,所有动作都没有被评价,此时采取某个动作获得正向激励导致后续状态只采用此动作而陷入局部最优,因而本文采用优化的 ϵ 贪心策略,在学习初期以较大概率 ϵ 随机选择动作,并随着学习步数提高,减小随机选择概率 ϵ ,直到稳定后,以大概率 $1 - \epsilon$ 根据 Q 值选择动作。

$$a = \begin{cases} \operatorname{argmax}_a Q(s, a), & p = 1 - \epsilon \\ \text{random}, & p = \epsilon \end{cases} \quad (19)$$

DQN在Q-learning基础上,引入深度学习(DL, deep learning)的神经网络,将 Q 值表存储

问题转换为函数拟合问题,通过更新参数 θ 使 $Q(s, a; \theta)$ 函数逼近 Q 值^[25-29]。在此基础上,使用经验池回放技术,通过随机抽样训练网络解决样本相关性及非静态分布问题^[30]。

1) 状态空间 $s[t]$

系统的状态空间用于描述环境参数,基于本文提出的系统模型,时隙 t 时的状态空间 $s[t]$ 可以定义为

$$s[t] = \{D', h', B'\} \quad (20)$$

其中, D' 表示本地设备的任务大小; h' 表示网络状态,即信道增益; B' 表示设备的电量能级。

2) 动作空间 $a[t]$

通过观察状态空间,智能体获得当前时隙的环境信息,并基于该环境,选择对应 Q 值最大的动作。

$$a[t] = \{0, 1, 2\} \quad (21)$$

其中,0表示本地执行,1表示MEC卸载,2表示D2D卸载。

3) 奖励函数 $r[t]$

当智能体根据当前状态做出一个动作时,会给予该动作一个即时奖励,该即时奖励最终会影响到智能体对强化学习的结果。很显然,可以使用式(14)的系统成本来表示奖励函数,因为系统成本与奖励成反比,可以将奖励函数表示为

$$r[t] = R(s, a) = -C_{\text{policy}}^t \quad (22)$$

本文提出的EDDQN算法首先依据 T 时隙时的任务信息、网络状态信息以及设备电量信息,选择动作,然后通过模拟退火算法分配设备计算能力和

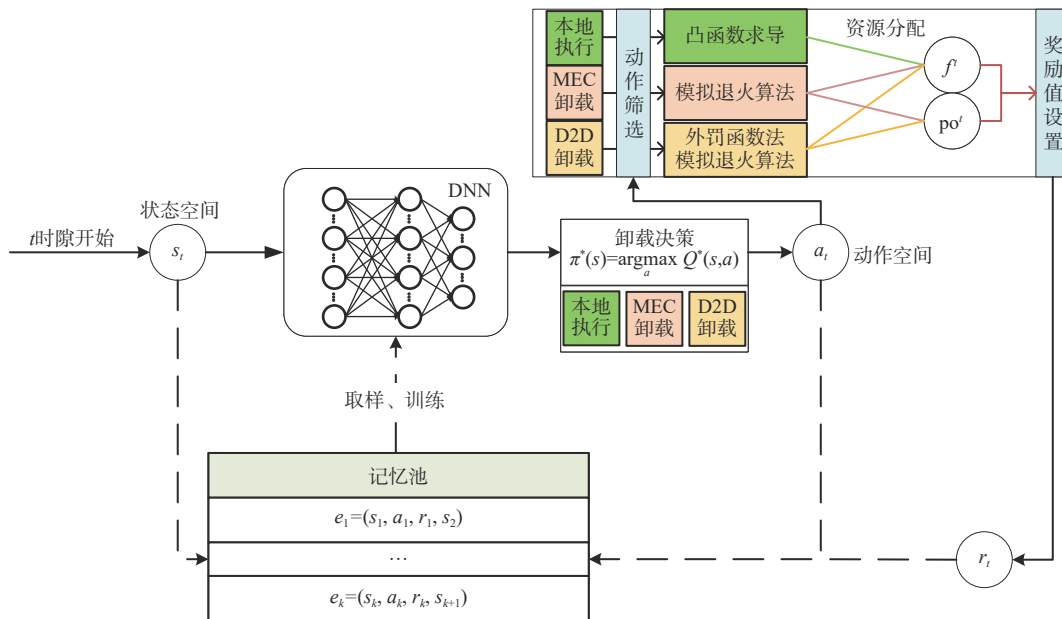


图2 EDDQN算法流程

传输功率，接着根据目标函数计算系统成本获得动作对应的奖励值，最后将状态—动作—奖励对放入经验池。如此迭代若干次后，对经验池随机抽样训练DNN，优化动作选择，获得长时间内系统成本最低的卸载决策方案。具体步骤如算法1所示。

算法1 基于EH-D2D的DQN (EDDQN) 算法

- 1) 初始化经验回放池 D 、动作价值对函数 Q 、随机网络权重 θ 、状态空间 s
- 2) **for** 迭代次数 Epoch **do**
- 3) **for** 每个时隙 T **do**
- 4) 获取任务信息集合 D^t 、网络信息集合 h^t 和设备电量信息集合 B^t
- 5) $s_t = \{D^t, h^t, B^t\}$
- 6) 根据状态空间，以一定概率 ε 随机选择动作集合 a_t ，以 $1-\varepsilon$ 的概率根据 $a_t^* = \operatorname{argmax}_{a_t} Q(s_t, a_t)$ 选择动作
- 7) 确定动作集合 a_t ，根据模拟退火算法计算最优的资源分配方案，分配设备计算能力 f^t 和传输功率 po^t
- 8) 执行动作 a_t ，然后得到奖励 r_t 和下一状态 s_{t+1}
- 9) 将 (s_t, a_t, r_t, s_{t+1}) 存入经验回放池 D
- 10) 从经验回放池中随机采样用于训练
- 11) 求解 $y_j = \begin{cases} r_j, & \text{结束} \\ r_j + \gamma \max_a Q(s', a'), & \text{其他} \end{cases}$
- 12) 更新网络权重 θ
- 13) **end for**
- 14) **end for**

2.3 资源分配算法

当确定执行某种卸载决策后，需要根据当前环境对设备分配相应的资源，最优化系统成本。本文针对3种不同卸载决策的优化问题，使用不同的最优化算法，求解出最佳的计算能力 f^* 和传输功率 po^* 。

2.3.1 卸载在本地设备的优化方案

当执行本地计算时，将P1问题转化为如下优化问题。

$$\begin{aligned} \text{P1.1: } \min_{f^t} & \alpha \frac{W^t}{f^t} + \beta \zeta W^t (f^t)^2 p_i^t(B^t) \\ \text{s.t.} & \quad \text{C1: } \frac{W^t}{f^t} \leq L^{\max} \\ & \quad \text{C2: } 0 \leq f^t \leq f_{\max} \end{aligned} \quad (23)$$

在每个时间片中，设备电量 B^t 已知，即本地能耗单价 $p_i^t(B^t)$ 固定不变，因此P1.1是一个关于单变量 f^t 的函数，实际上是求解凸函数 $G(f^t)$ 的最小值，对凸函数求导，令导函数为零，获得极值。

$$f^* = \sqrt[3]{\frac{\alpha}{2\beta\zeta p_i^t}} \quad (24)$$

接着，根据C1、C2求出 f^t 的可行域，与极值比较，获得本地成本最小的3种结果。

$$G^*(f^t) = \begin{cases} G\left(\frac{W^t}{L_{\max}}\right), & f^* \leq \frac{W^t}{L_{\max}} \\ G(f^*), & \frac{W^t}{L_{\max}} < f^* \leq f_{\max} \\ G(f_{\max}), & f^* > f_{\max} \end{cases} \quad (25)$$

2.3.2 卸载到边缘服务器的优化方案

当执行MEC卸载时，将P1问题转换为如下优化问题。

$$\begin{aligned} \text{P1.2: } \min_{po^t, f^t} & \alpha \frac{D^t}{B \ln\left(1 + \frac{h^t po^t}{\sigma^2}\right)} + \\ & \beta \left(\frac{D^t po^t}{B \ln\left(1 + \frac{h^t po^t}{\sigma^2}\right)} p_i^t + \zeta W^t (f^t)^2 p_r^t \right) \\ \text{s.t.} & \quad \text{C1: } \frac{D^t}{B \ln\left(1 + \frac{h^t po^t}{\sigma^2}\right)} \leq L^{\max} \\ & \quad \text{C2: } f_{\min} \leq f^t \leq f_{\max} \\ & \quad \text{C4: } 0 \leq po^t \leq po_{\max} \end{aligned} \quad (26)$$

P1.2是一个关于多变量 f^t 和 po^t 的简单约束非线性规划问题，因此使用模拟退火算法解决该优化问题。

模拟退火算法是模拟物理的退火原理，即高温物体在逐渐降温的过程中，物体内能减少，内部的粒子从扰乱状态趋于稳定。将粒子看作解空间，粒子的运动看作搜索过程，一开始，高温下粒子运动剧烈，在可行域内广泛搜索目标函数的全局最优解，从而概率跳出局部最优解，随着温度的降低，粒子搜索范围减小，局部搜索能力增强，最终收敛到最优解或近似最优解^[15,31]。

本文设置模拟退火算法初始温度 T 为 100°C ，马尔可夫链长度即内循环次数为100，在当前温度每迭代搜索100次，以0.98的降温系数降低温度，直到降低至终止温度 1°C 时获得最终解。对于每次

搜索, 定义初始搜索步长为0.5, 随优化次数增多减小搜索步长, 从而提高搜索精度, 其搜索范围受C2和C4约束条件影响。

该启发式算法可以很好地解决无约束或简单的上下限约束的优化问题, 从而计算出MEC执行任务所需的最优CPU频率 f^* 与传输功率 po^* 。

2.3.3 D2D卸载情况下的优化方案

当执行D2D卸载时, 将P1问题转化为如下优化问题。

$$\begin{aligned}
 \text{P1.3.1: } \min_{po^t, f^t} & \alpha \left(\frac{W^t}{f^t} + \frac{D^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} \right) + \\
 & \beta \left(\frac{D^t po^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} p_i^t + \zeta W^t (f^t)^2 p_d^t \right) \\
 \text{s.t. } \quad \text{C1: } & \frac{W^t}{f^t} + \frac{D^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} \leq L^{\max} \\
 & \text{C2: } 0 \leq f^t \leq f_{\max} \\
 & \text{C4: } 0 \leq po^t \leq po_{\max}
 \end{aligned} \quad (27)$$

P1.3.1是一个关于多变量 f^t 和 po^t 的非线性约束下的非线性规划问题。对于C1这种多变量不等式约束, 可以将每次产生的新解代入C1从而判断是否符合, 将符合的留下, 不符合的丢弃重新产生。

虽然这种方法比较实用, 但会浪费较多时间在重新产生新解上, 因此针对该类复杂优化问题, 本文选择对优化函数进行改进, 使用外罚函数法, 将复杂不等式约束C1罚至目标函数, 构造新的目标函数, 从而消除复杂约束条件。

$$\begin{aligned}
 \text{P1.3.2: } \min_{po^t, f^t} & \alpha \left(\frac{W^t}{f^t} + \frac{D^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} \right) + \\
 & \beta \left(\frac{D^t po^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} p_i^t + \zeta W^t (f^t)^2 p_d^t \right) + \\
 & \sigma \max \left(0, \frac{W^t}{f^t} + \frac{D^t}{B \text{lb} \left(1 + \frac{h^t po^t}{\sigma^2} \right)} - L^{\max} \right)^2 \\
 \text{s.t. } \quad \text{C2: } & 0 \leq f^t \leq f_{\max} \\
 & \text{C4: } 0 \leq po^t \leq po_{\max}
 \end{aligned} \quad (28)$$

如P1.3.2所示, 复杂不等式约束优化问题已经转化为简单上下限约束问题, 此时采用模拟退火算

法即可计算出最优解 f^* 和 po^* , 模拟退火算法流程如图3所示。

2.4 算法分析

本节将对EDDQN算法的时间复杂度进行分析。EDDQN算法是基于DQN算法和模拟退火算法的混合算法, 用于解决卸载决策和资源分配问题。

EDDQN算法的主要影响因素是迭代次数(epoch) E 、每次迭代执行时隙数 T 、模拟退火算法温度参数 t 、每次降温迭代次数 e 以及设备数 N 。

根据算法1所示, 步骤(1)表示初始化, 其中经验回放池 D 、动作价值对函数 Q 等参数都是创建算法所需的数据结构, 大小通常固定, 不随数据规模而变化, 可以视为常数时间复杂度。步骤(2)~(14)、(3)~(13)双重外循环表示DQN总执行次数, 时间复杂度为 $O(E \times T)$, 内部步骤(4)~(11)为算法核心。步骤(4)~(6)、步骤(8)~(12)对每个设备进行操作, 时间复杂度受状态空间大小和设备数影响, 由于本算法状态空间大小为3, 因此该步骤时间复杂度为 $O(3 \times N)$, 步骤(7)使用模拟退火算法求解 N 个设备的计算能力和传输功率, 解空间参数有 $2 \times N$ 个, 因此时间复杂度为 $O(e \times t \times 2 \times N)$ 。

将所有时间复杂度汇总, 分析得出EDDQN算法的总时间复杂度为 $O(E \times T \times (3 + e \times t \times 2) \times N)$ 。其中, 除了设备数 N 随着规模发生变化, 其他参数均为常数, 因此, 时间复杂度可以简化为 $O(K \times N)$, K 为常数。

3 仿真分析

3.1 系统参数设置

本节将针对所提出的算法搭建仿真平台, 测试算法性能并验证其结果。实验环境为Windows10系统, 编程语言是Python3.7, 使用的深度强化学习框架为TensorFlow, 其余仿真参数如表3所示。

本文考虑稳定能量环境和极端能量环境两种情况下所提算法的性能优化。在稳定能量环境下, 设置设备收集能量的能力与传输消耗的能量相匹配, 即每一时隙收集到的可再生能源足以支撑当前时隙的任务传输; 在极端能量环境下, 设备收集能量的能力远不如任务执行时的能量消耗。

与此同时, 选择一些经典传统算法做对比。

1) 全本地策略 (local only): 该策略不考虑系统整体情况, 将产生的任务全部交由本地, 以固定

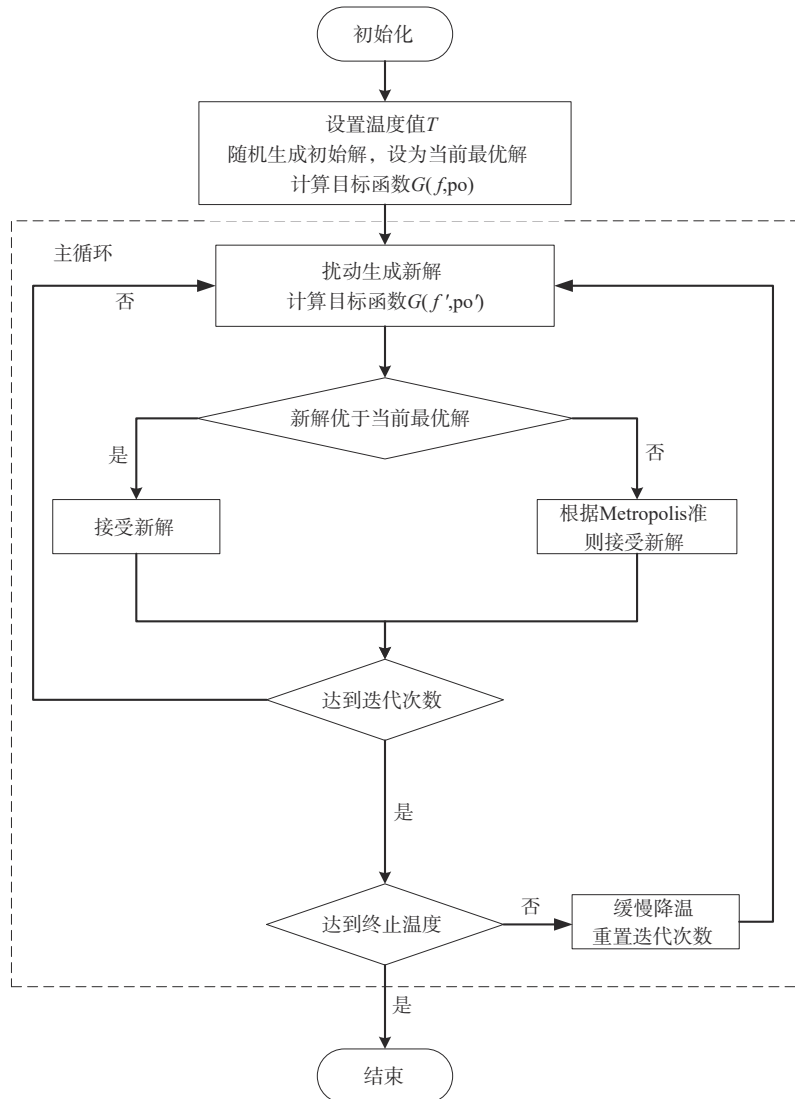


图3 模拟退火算法流程

计算能力在本地执行。

2) 全卸载策略 (remote only): 同全本地策略, 该策略将全部任务卸载到 MEC 服务器, 而不考虑网络环境变化情况, 以固定传输功率传输任务。

3) 随机策略 (random): 该策略在产生任务后, 从本地执行、MEC 卸载以及 D2D 卸载 3 种动作中随机选择一个动作。

4) 最优即时策略 (instant optimal): 该策略是贪心策略的一种, 总是选择当前时隙内回报最大的一种动作。

3.2 仿真结果分析

仿真实验分为 3 个部分, 第 1 部分测试不同环境下不同策略对能耗的影响, 第 2 部分测试不同环境下不同策略的成本性能表现, 第 3 部分对比不同

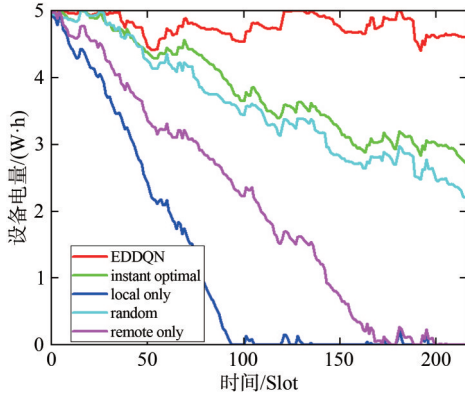
环境下不同策略的任务丢弃率。

3.2.1 能耗性能仿真

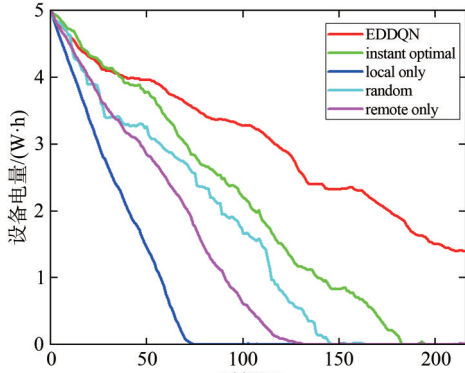
首先对稳定能量环境和极端能量环境进行设备剩余电量统计, 不同环境下设备电量变化如图 4 所示。可以看出, 在稳定能量环境下, 所提出的算法能够将设备电量维持在 4~5 W·h, 而其他算法下的设备电量随着时间以一定程度或缓慢或急速下降, 始终无法保持设备的可持续性运行。在极端能量环境下, 收集到的可再生能源无法供应任务执行消耗的能量, 所有算法的设备电量都随着时间变化而下降, 此时, 所提出的算法通过合理的决策卸载, 选择合适的计算能力和传输功率, 从而减少执行任务时的能量消耗, 尽可能地缓解该环境下能源的供应不足问题。

表 3 仿真参数

参数名称	参数符号	参数数值
设备数量(本地/辅助)	N	10
任务大小	D/kbit	[100, 700]
任务所需 CPU 周期数	$W^l/\text{MegaCycles}$	[30, 208]
本地设备计算能力	f_{loc}/GHz	2
MEC 服务器计算能力	f_{mec}/GHz	20
辅助设备计算能力	f_{d2d}/GHz	3
本地执行能量单价	p_l^i	[0.1, 2]
MEC 卸载能量单价	p_s^i	0.005
D2D 卸载能量单价	p_d^i	0.25
最大容忍时延	L_{max}/ms	95
最大传输功率	$p_{o_{max}}/\text{W}$	2
设备分配带宽	B/MHz	1
高斯噪声功率	σ^2	2×10^{-13}
芯片系数	ζ	10^{-28}
设备电量上限	$b_{max}/(\text{W} \cdot \text{h})$	5
时延权重因子	α	0.4
能耗权重因子	β	0.6
成本丢弃惩罚	ϕ	0.3



(a) 稳定能量环境



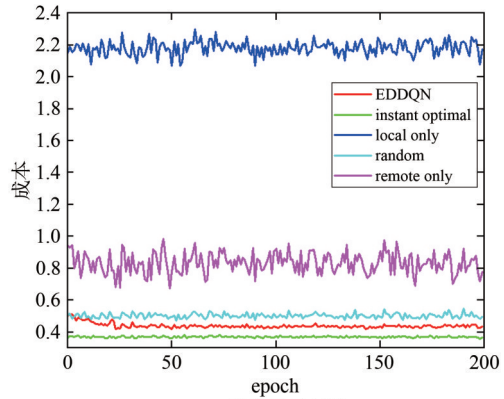
(b) 极端能量环境

图 4 不同环境下设备电量变化

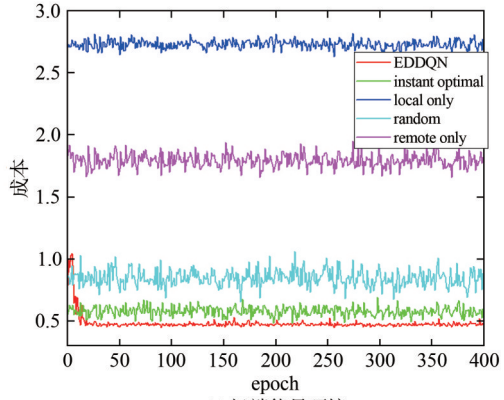
3.2.2 成本的性能仿真

第 2 部分实验分别对稳定能量环境和极端能量

环境进行成本性能测试。所提出的 EDDQN 算法是一种在线学习算法，一开始不需要提供完整的训练数据集，随着轮数增多，数据积累，模型会在操作中不断更新，不同环境下成本比较如图 5 所示，在前 10 个 epoch 中，EDDQN 算法会随机选择卸载决策，积累数据用于更新深度学习中的随机网络权重 θ ，加强神经元之间的联系，因此此时的系统总成本与随机策略几乎一致，随后训练出最佳的卸载决策，成本逐渐收敛。



(a) 稳定能量环境



(b) 极端能量环境

图 5 不同环境下成本比较

在稳定能量环境中，最优即时策略算法每次选择当前时隙成本最小的卸载决策，保证系统总成本要优于 EDDQN 算法，然而该算法无法考虑设备电量问题，即使能量供应充足，依然无法维持设备持续运行。在极端能量环境中，这一问题更为明显，随着设备电量的提前耗尽，设备将任务丢弃带来较大的成本惩罚，最终成本低于 EDDQN 算法。

而 EDDQN 算法不仅考虑当前时隙的最优选择，还考虑此次选择对未来长期造成的影响，即设备电量达到下限，任务丢弃带来的惩罚。在若干次学习之后，所提出的算法以一定成本为代价保证了

设备电量的供应问题，在稳定能量环境中保证良好的性能，在极端能量环境中更进一步维持成本与电量的平衡。

3.2.3 任务丢失率的仿真

不同环境下任务丢失率比较如图6所示。全本地策略因本地执行任务提前耗尽设备电量导致任务丢失，全卸载策略则因网络环境不佳依然选择传输任务，导致时延与能耗较高，受超时丢失任务以及设备电量不足丢失任务两种情况影响，造成较高的任务丢失率。随机策略与最优即时策略有概率选择D2D卸载，可以将负载分担至辅助设备，从而降低任务丢失率。而EDDQN算法可以根据网络状况、任务信息、设备电量情况选择最优卸载决策，并合理分配资源，避免超时丢失与低电量丢失，降低任务丢失率。

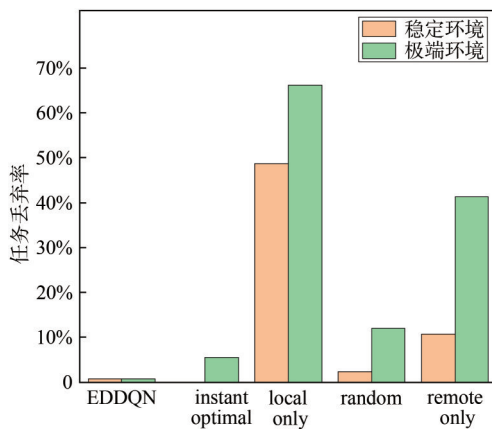


图6 不同环境下任务丢失率比较

4 结束语

本文基于供电困难场景设计一种成本优化的卸载决策与资源分配方案，通过联合使用EH、D2D以及DVFS技术解决场景中的设备能耗问题，以可再生能源作为唯一电源，辅助设备向本地设备提供计算能力，协助计算任务从而减少本地设备能耗，并使用模拟退火算法解决资源分配问题。

重点在于研究一种新场景，即供电困难的应用场景。当短时间内能量环境恶劣时，能量收集设备如何卸载与分配资源，从而降低能耗与成本。本文并未过多考虑算法的性能优化与创新，如强化学习算法的收敛速度、深度学习算法中神经网络的学习速度等。未来将对算法进行改进，通过将训练数据直接注入经验池、优化DQN算法框架等方法优化

算法的时间复杂度。同时，考虑具体的供电困难应用场景，在实际场景下解决存在的卸载问题。

参考文献：

- [1] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.
- [2] GARCIA LOPEZ P, MONTRESOR A, EPEMA D, et al. Edge-centric computing[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(5): 37-42.
- [3] 王英恺, 王青山. 能量收集无线通信系统中基于强化学习的能量分配策略[J]. 计算机科学, 2021, 48(7): 333-339.
WANG Y K, WANG Q S. Reinforcement learning based energy allocation strategy for multi-access wireless communications with energy harvesting[J]. Computer Science, 2021, 48(7): 333-339.
- [4] MA D, LAN G, HASSAN M, et al. Sensing, computing, and communications for energy harvesting IoTs: A survey[J]. IEEE Commun. Surveys Tuts., 2020, 22(2): 1222-1250.
- [5] ZENG D Z, PAN S L, CHEN Z L, et al. An MDP-based wireless energy harvesting decision strategy for mobile device in edge computing[J]. IEEE Network, 2019, 33(6): 109-115.
- [6] PADHY A, JOSHI S, BITRAGUNTA S, et al. A survey of energy and spectrum harvesting technologies and protocols for next generation wireless networks[J]. IEEE Access, 2021(9): 1737-1769.
- [7] MEHRABI M, YOU D, LATZKO V, et al. Device-enhanced MEC: multi-access edge computing (MEC) aided by end device computation and caching: A survey[J]. IEEE Access, 2019(7): 166079-166108.
- [8] ZHOU H, WU T, ZHANG H J, et al. Incentive-driven deep reinforcement learning for content caching and D2D offloading[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(8): 2445-2460.
- [9] 马惠荣, 陈旭, 周知, 等. 绿色能源驱动的移动边缘计算动态任务卸载[J]. 计算机研究与发展, 2020, 57(9): 1823-1838.
MA H R, CHEN X, ZHOU Z, et al. Dynamic task offloading for mobile edge computing with green energy[J]. Journal of Computer Research and Development, 2020, 57(9): 1823-1838.
- [10] ZHANG L T, XIONG K, ZHANG Y. UAV-assisted wireless energy harvesting fog computing network optimization method[J]. Journal of Software, 2019, 30(1): 9-17.
- [11] BAKAR A, HESTER J. Making sense of intermittent energy harvesting[C]//Proceedings of the 6th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems. New York: ACM Press, 2018: 32-37.
- [12] TIAN X Z, YAO C, ZHAO C, et al. 5G network-oriented mobile edge computation offloading strategy[J]. Computer Science, 2020, 47(11A): 286-290.

- [13] 池凯凯, 徐欣晨, 魏欣晨. 射频能量捕获传感网中满足节点吞吐量需求的基站最少化部署方案[J]. 计算机科学, 2018, 45(S1): 332-336.
CHI K K, XU X C, WEI X C. Minimum deployment scheme of base stations to meet node throughput requirements in RF energy capture sensor networks[J]. Computer Science, 2018, 45(S1): 332-336.
- [14] YUAN F C, ZHANG Q T, JIN S, et al. Optimal harvest-use-store strategy for energy harvesting wireless systems[J]. IEEE Transactions on Wireless Communications, 2015, 14(2): 698-710.
- [15] ZHAO F J, CHEN Y, ZHANG Y C, et al. Dynamic offloading and resource scheduling for mobile-edge computing with energy harvesting devices[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 2154-2165.
- [16] MAO Y Y, ZHANG J, LETAIEF K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605.
- [17] WANG L H, SHAO H, LI J J, et al. Optimal multi-user computation offloading strategy for wireless powered sensor networks[J]. IEEE Access, 2020(8): 35150-35160.
- [18] SUN Y F, WU J G, CHEN L, et al. Latency optimization for mobile edge computing with dynamic energy harvesting[C]//Proceedings of 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom). Piscataway: IEEE Press, 2019: 79-83.
- [19] TAN M X, CHEN B, PANG R M, et al. MnasNet: platform-aware neural architecture search for mobile[C]//Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 2815-2823.
- [20] TAYLOR B, MARCO V S, WOLFF W, et al. Adaptive deep learning model selection on embedded systems[J]. ACM SIGPLAN Notices, 2018, 53(6): 31-43.
- [21] LI J, GAO H, LV T J, et al. Deep reinforcement learning based computation offloading and resource allocation for MEC[C]//Proceedings of 2018 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2018: 1-6.
- [22] MIN M H, XIAO L, CHEN Y, et al. Learning-based computation offloading for IoT devices with energy harvesting[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1930-1941.
- [23] WANG Q, LENG S P, FU H R, et al. An IEEE 802.11p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(2): 449-458.
- [24] XU J, CHEN L X, REN S L. Online learning for offloading and autoscaling in energy harvesting mobile edge computing[J]. IEEE Transactions on Cognitive Communications and Networking, 2017, 3(3): 361-373.
- [25] DU J B, ZHAO L Q, FENG J, et al. Computation offloading and resource allocation in mixed fog/cloud computing systems with Min-max fairness guarantee[J]. IEEE Transactions on Communications, 2018, 66(4): 1594-1608.
- [26] 李茹杨, 彭慧民, 李仁刚, 等. 强化学习算法与应用综述[J]. 计算机系统应用, 2020, 29(12): 13-25.
LI R Y, PENG H M, LI R G, et al. Overview on algorithms and applications for reinforcement learning[J]. Computer Systems & Applications, 2020, 29(12): 13-25.
- [27] CHEN N, ZHANG S, QIAN Z Z, et al. When learning joins edge: real-time proportional computation offloading via deep reinforcement learning[C]//Proceedings of 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS). Piscataway: IEEE Press, 2019: 414-421.
- [28] HUANG L, BI S Z, ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. IEEE Transactions on Mobile Computing, 2020, 19(11): 2581-2593.
- [29] CHEN X F, ZHANG H G, WU C, et al. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning[J]. IEEE Internet of Things Journal, 2019, 6(3): 4005-4018.
- [30] 刘朝阳, 穆朝絮, 孙长银. 深度强化学习算法与应用研究现状综述[J]. 智能科学与技术学报, 2020, 2(4): 314-326.
LIU Z Y, MU C X, SUN C Y. An overview on algorithms and applications of deep reinforcement learning[J]. Chinese Journal of Intelligent Science and Technology, 2020, 2(4): 314-326.
- [31] 庞峰. 模拟退火算法的原理及算法在优化问题上的应用[D]. 长春: 吉林大学, 2006.
PANG F. The principle of SA algorithm and algorithm's application on optimization problem[D]. Changchun: Jilin University, 2006.

[作者简介]



王珺(1975-), 女, 博士, 南京邮电大学副教授, 主要研究方向为物联网、边缘计算、下一代网络等。



赵浩东(1998-), 男, 南京邮电大学硕士生, 主要研究方向为物联网和边缘计算。