



ELSEVIER

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng



Research
Cyber Technology—Article

面向物联网的激励感知区块链辅助的智能边缘缓存与计算迁移研究

王倩^{a,b}, 陈思光^{a,b,*}, 吴蒙^a

^a Jiangsu Key Laboratory of Broadband Wireless Communication and Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

^b School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

ARTICLE INFO

Article history:

Received 28 March 2022

Revised 26 August 2022

Accepted 31 October 2022

Available online 4 January 2023

关键词

计算迁移

缓存

激励

区块链

联邦深度强化学习

摘要

人工智能的快速发展将物联网推向了一个新阶段,面对数据的爆炸性增长和用户对更高质量服务的迫切需求,边缘计算和缓存被视为富有前景的技术解决手段。然而,边缘节点(edge nodes, ENs)中的资源并不是取之不尽的。本文提出了一种面向物联网的激励感知区块链辅助的智能边缘缓存与计算迁移方案,该方案致力于为协作ENs在资源优化和控制方面提供安全和智能的解决方案。具体地,该方案通过联合优化迁移和缓存决策以及计算和通信资源分配,最大限度地降低EN中完成任务的总成本。此外,为解决上述优化问题,本文设计了区块链激励和贡献联合感知的联邦深度强化学习算法。在本地训练期间,该算法构建了一个激励感知区块链辅助的协作机制,即在安全保障前提下增强ENs参与协作的意愿。同时,提出了一种基于贡献的联邦聚合方法,即基于EN对全局模型性能提升所做贡献来计算其梯度的聚合权重,以提升训练效果。最后,与其他基准方案相比,数值结果证明本文方案具备高效的资源优化效用,同时在总成本降低和缓存性能方面具有显著优势。

© 2022 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. 引言

随着物联网的广泛普及,人们的生活正在发生巨变,特别是从人工智能(artificial intelligence, AI)的出现以来,智慧交通和智慧医疗等智能应用层出不穷。例如,人们可以在出行前掌握所有交通情况,然后选择一条交通量较少的路线到达目的地;人们还可以通过智能穿戴设备检测其身体状况,实现对健康的实时监测。这些新型应用正被越来越多的用户使用,并每天产生大量数据。根据Cisco统计,全球各类感知终端每天生成约2.5 EB的数据[1]。通常,这些设备的缓存和计算资源极其有限,它们将数据

(执行代码等)传输到资源丰富但距离遥远的云中心以完成任务处理,这种方法不能满足用户对低延迟和低能耗的需求。边缘计算可有效地缓解上述问题,因为边缘节点(edge node, EN)更接近用户,并且往往配备了比普通终端更丰富的资源。

边缘计算已成为近年来备受关注的计算范式,如何高效利用整个边缘网络中的资源是一个热门话题[2]。例如,通过优化迁移决策, Wu等[3]将能量和延迟成本最小化,从而充分提升了用户的服务体验。除了计算资源外, EN还配备了一定的缓存资源。Zhang等[4]考虑将某些任务计算所需的内容缓存到EN中,以减少传输时间,通过联合

* Corresponding author.

E-mail address: sgchen@njupt.edu.cn (S. Chen).

优化迁移决策、缓存决策和资源分配，最小化完成所有计算任务的延迟。事实上，在整个网络中通常存在多个 ENs，每个 EN 的资源使用和工作负载都不同，这些 ENs 可以通过相互合作以进一步提高资源利用率。例如，Zhang 等[5]从边缘协作迁移的角度提出了一种新思路，利用边到边和云边协同计算迁移来完成基于 AI 的计算任务，实现延迟最小化和资源高效利用。然而，在现实场景中，每个 EN 都是自私的，具有足够资源的 ENs 通常不会自愿免费提供自己的资源。Zhao 等[6]和 Zeng 等[7]为车辆任务迁移设计了基于激励的协作机制，向邻近车辆或边缘服务器支付租金，以鼓励它们协助请求者完成任务。最终获得最优的任务分配策略以最小化系统成本。

在上述研究中，所设计的求解方法基于传统数学规划、Lyapunov 或遗传算法的原理，这些方法要么不适用于动态和复杂的网络环境，要么解决过程依赖于复杂且代价较高的条件。深度强化学习（deep reinforcement learning, DRL）由于其灵活的自主学习能力而广受欢迎，并突破了传统数学规划算法在问题解决方面的局限性。然而，集中式训练需要在云中心等具有强大计算能力的设备上进行，并且这种训练方式还会导致频繁的数据传输[8]。此外，在协作过程中，节点不可避免地会产生好奇心，这会带来隐私泄露的风险，尤其是对于患者诊断信息等这类隐私敏感的数据。作为新兴的分布式学习方法，联邦学习（federated learning, FL）提供了一种解决上述问题的新视角。它无需共享本地原始数据，而仅共享梯度信息以训练全局模型。每个智能体（即设备或 EN）利用本地数据进行训练，然后将其梯度上传到聚合节点，该节点负责执行联邦聚合以更新全局模型。例如，Huang 等[9]提出了一种基于联邦 DRL 的小型基站之间的协作计算迁移方案。通过优化迁移决策和交互协作，获得了最小总能耗，这不仅减少了训练过程中的通信开销，还保护了每个小型基站本地训练数据的安全性。在物联网边缘计算的研究中有许多优秀的资源优化解决方案，但是设计一个智能、高效与安全的资源优化方案仍然面临以下三个挑战：

(1) 一些现有方案对资源和控制要素进行了简单的优化。例如，它们优化迁移决策或缓存决策时，对资源分配的考虑有限；其他一些方案具有更全面的考虑，但都是基于理想情况。例如，选择的迁移节点始终处于空闲状态，彼此之间没有竞争关系等。此外，大多数方案假设邻居节点自愿且免费贡献其资源。

(2) 一些考虑协作因素的方案采用激励机制来增强节点参与协作的积极性。然而，并不是所有邻近节点都适合作为协作候选节点，即协作节点未经合理筛选，从而可能

降低协作的质量。此外，协作的安全性也无法得到保障。

(3) 现有工作中基于联邦 DRL 的算法是有效的，但大多数采用简单的平均聚合方式而没有准确地表达每个智能体对系统的贡献，训练效果亟待提升。

受以上挑战的启发，本文提出了一种新的方案，即面向物联网的激励感知区块链辅助的智能边缘缓存与计算迁移方案，具体贡献如下：

(1) 为了最小化 EN 中完成所有任务的总成本，本文联合优化迁移决策、缓存决策、计算资源分配和带宽分配，其中总成本包括延迟、能耗和协作成本。该方法实现了整个网络中资源和控制要素的全面优化，并充分利用 ENs 之间的协作进一步提高资源利用率。此外，基于任务偏好的协作成本定价规则更加实用。

(2) 此外，本文提出了一种区块链激励和贡献联合感知的联邦深度强化学习算法（blockchain incentive and contribution co-aware federated deep reinforcement learning, BICC-FDRL），并研究了一种激励感知区块链辅助的协作机制用于本地训练。该机制可以鼓励所有 ENs 积极参与协作，并保证安全性。该机制设计了一种低通信成本的新型激励方法，所有合格的候选 ENs 均想通过竞争成为最终合作者以获得收益。

(3) 特别地，在 BICC-FDRL 的联邦聚合过程中，为了准确直观地表达每个智能体的贡献，本文提出了一种基于贡献的联邦聚合方法。通过这种方法，表现优异的智能体将获得更多关注。因此，全局模型的训练效果可以得到提升。

最后，实验结果验证了本文提出的方案在缓存性能、总成本降低和效用优化方面具有显著优势。

本文的剩余部分组织如下：第 2 节介绍相关工作；第 3 节构建系统模型；第 4 节介绍问题的定义；在第 5 节，本文设计了 BICC-FDRL 算法；随后，在第 6 节中展示性能评估；最后，第 7 节对本文进行了总结。

2. 相关工作

近年来，面向物联网的边缘计算备受欢迎，大量关于通过资源优化来提高其性能的研究工作不断涌现。在优化变量（控制和网络资源的分配）方面，文献[10]–[15]最小化任务完成的成本。在迁移、计算和通信资源的联合优化方面，Chen 等[10]使用动态电压调节技术和交替最小化算法来最小化任务完成的能量消耗。类似地，Malik 等[11]集成了无线充电和计算迁移，以最小化系统的能量消耗。针对 EN 中有限的缓存资源，Liu 等[12]在边缘计算中提出

了一种流行数据缓存方案，通过优化缓存决策，使得服务提供商的缓存收益最大化。从更全面的角度来看，文献[13]~[15]充分利用计算迁移和缓存来提高系统性能。为了最小化所有用户的平均能耗，Chen等[13]提出了一种采用基于动态规划的算法的最佳迁移和缓存决策的方案。Bi等[14]和Zhang等[15]联合优化迁移、服务缓存、计算和通信资源的分配。为了最小化延迟和能量消耗的加权和，Bi等[14]分两步解决优化问题：首先，他们推导出了具有最优资源分配的闭式表达式，然后采用交替最小化算法优化迁移和缓存决策。针对具有多个任务的移动边缘计算系统中的多用户场景，Zhang等[15]研究了一种基于半定松弛和交替优化的算法，以最小化系统成本。以上研究利用了EN的计算和缓存能力，提高了任务完成的效率，但单个EN的能力是有限的。此外，如果每个EN只为其本地用户提供服务，将导致负载不均衡和资源浪费。

为了填补上述空白，文献[16]~[19]利用ENs之间的协作进一步提高资源利用率和负载均衡。Ma等[16]和Zhong等[17]的目标是将任务完成时间最小化。Ma等[16]研究了ENs之间的协作服务缓存和计算迁移，通过基于Gibbs采样的迭代算法来优化任务的缓存决策和迁移比例。该方案可以很好地适应ENs的异构性。文献[17]中提出的协作方案与文献[16]类似，但还考虑了资源分配，并提出了一种改进的广义Benders分解算法来解决优化问题。为了将所有用户的总延迟和能耗最小化，Feng等[18]研究了相邻ENs的协作数据缓存和迁移，这些ENs可以共享缓存数据和计算资源，从而提高用户的服务质量。与以往传统的ENs之间的协作缓存不同，Yuan等[19]关注协作ENs和转发组的最优数量，并提出了一种改进的交替方向乘法，可以在低协作成本的约束下获得最大的缓存命中率。这些研究提高了网络资源的利用率，但假定服务提供商自愿向请求者提供计算或缓存服务。实际上，由于缺少激励措施，它们参与协作过程的积极性并不高。

针对上述研究的不足，文献[20]~[24]设计了激励机制来提高用户、ENs或云的协作积极性，从而使参与者受益。文献[20]提出了云服务运营商和边缘服务器之间的激励机制；此外，作者还通过联合优化云运营商的迁移决策和边缘服务器的付款来设计计算迁移方案，以最大化它们的效用并达到纳什均衡。Hou等[21]提出了一种边-端激励驱动的任务分配方案，以最大化系统效用。根据设备中任务的特点和重要性，可以将其迁移到本地边缘服务器、相邻的边缘服务器或同一本地边缘服务器的覆盖范围内的设备群中。在文献[22]中，作者充分利用了空闲边缘网关和移动设备的计算资源，形成边缘云。具体而言，建立了

一个边缘云和移动设备之间的资源交易模型，旨在通过利用基于市场的定价和拍卖理论来最大化边缘云的利润，并满足移动设备的计算要求。Luo等[23]在设备对设备（device to device, D2D）网络中设计了一个高效的激励机制。根据合作博弈，在设备之间建立多个微型计算集群以实现协作任务计算，使每个设备从辅助计算或中继中受益，并有效降低系统的全局成本。此外，对于D2D网络，Zhang等[24]基于多赢家拍卖研究了一种协作缓存，它可以通过最佳迁移和支付策略获得所有用户的最大内容缓存收益，并保证拍卖赢家的利润公平。这些方案通过设计激励机制激发网络积极性（即网络中的节点可以主动互动并充分利用资源）。但是，他们提出的求解方法都基于传统的迭代数学规划或博弈论，获得最优解的前提较为苛刻，不适用于动态和复杂的网络环境。此外，它们忽略了边缘节点之间的协作。

为了构建一个适应动态和复杂环境的安全和智能的网络，引入了基于机器学习的方法。在文献[25]~[29]中，它们结合了DRL的自主学习能力和FL的安全分布式计算，可以克服上述方案的缺点。例如，文献[25]~[27]提出了基于双深度Q网络的FL算法。Zarandi和Tabassum [25]将设备视为智能体并在EN上执行联邦聚合，它通过联合优化迁移决策以及计算和通信资源的分配，实现终端侧任务完成成本的最小化。文献[26]的研究与文献[25]类似，并且对DRL和FL进行了更详细的研究和分析。此外，它还额外考虑了缓存优化。Ren等[27]设计了ENs和设备之间的奖惩机制（即设备将向提供计算服务的EN支付奖励，但EN将因任务计算失败受到惩罚）。该机制通过优化迁移决策和能量单元的分配来实现设备支付的最小化。尽管FL可以通过将数据保留在本地来保护数据隐私，但智能体在上传梯度的过程中容易受到恶意攻击。为了进一步增强数据隐私保护，Cui等[28]将区块链技术引入FL，并为每个智能体设计了四个智能合约，以确保其数据的安全性。此外，他们通过压缩智能体上传的梯度信息，减少聚合的通信开销，并通过优化缓存决策，在缓存命中率和安全性方面获得了出色的性能。激励理念也存在于区块链技术中，Yu等[29]在其未来工作中提到，在区块链中利用激励理念进行资源借用，不仅可以加强参与者交互的安全性，还可以发挥高效资源利用的作用。

通过对上述研究的分析，可以看出它们都有各自优势，并为本文提供了很大的启发。然而，这些方案在资源的高效利用和求解效果方面仍有提升空间。不仅需要在整个边缘网络中全面优化资源和控制，还需要提供一种竞争与协作共存的安全智能协作方案，这将显著提高系统性能。

3. 系统模型

随着物联网和人工智能技术的快速发展, 各种智能应用得到了广泛发展, 例如, 智慧交通中的车辆路径优化和智慧医疗中的健康监测。这些应用导致传感器产生的数据急剧增长, 用户对服务延迟和隐私保护的要求更加严格[30]。受此启发, 本文设计了一种激励感知区块链辅助的智能边缘计算迁移与缓存模型。如图1所示, 该模型包括三层: 用户层、边缘层和云层。整个网络的覆盖面积被划分为 M 个子区域, 每个子区域包括 N 个用户和一个EN, 因此用户层有 $M \times N$ 个用户, 边缘层有 M 个具有一定缓存与计算能力的ENs。

当每个用户通过无线链路向其本地EN发送任务请求时, EN可以直接为其本地用户提供服务, 或者向相邻的EN发送结果共享/计算迁移请求, 因为相邻节点可以相互协作。此外, 为了通过有益且安全的方式提高它们参与的积极性, ENs之间形成了激励感知区块链辅助的协作。在它们的协作过程中, 云层的云中心将部署相应的智能合约, 为结果共享和计算迁移设置交易规则。各层的具体功能定义如下:

(1) 用户层。每个用户在时刻 t 生成一个计算任务, 不同用户可能生成同一类任务。整个用户层在时刻 t 生成的任务集为 $\Gamma = \{1, 2, \dots, F\}$, 即有 F 类任务。由于用户自身能力的有限, 这些任务可能会被发送到它们的本地ENs进行高效处理, 因此本文设定EN m 接收到的任务集为

Γ_m , 其中, $m \in \mathcal{M} = \{1, 2, \dots, M\}$, \mathcal{M} 是ENs的集合。

(2) 边缘层。边缘层由 M 个ENs组成, 这些节点配备具有一定缓存和计算能力的服务器。此处, ENs被视为智能体, 它们可以使用从本地用户收集的信息进行模型训练。由于单个EN的能力有限, 邻居之间可以互相协助, 提高任务处理和资源利用的效率, 并且将激励感知区块链辅助的协作模型应用于此过程对协作的ENs来说是受益的。具体来说, 当EN从本地用户层接收到服务请求时, 如果结果已经被缓存, 则直接返回给请求用户(即缓存命中)。否则, EN将向云中心发送请求, 以寻求来自邻居EN的结果共享, 因此, 它将获得可交易的EN地址。根据智能合约, EN在接收到结果后会向邻居EN支付相应的结果共享报酬。如果该任务结果在任何地方都没有被缓存, 则EN将决定是否在本地处理或将其迁移到邻近的EN。如果任务需要迁移, 则类似于缓存共享的过程(即向云中心发送请求, 将任务迁移到邻近的EN)。最终, EN将获得结果并将其返回给用户。每个EN使用本地数据训练其模型后, 将其模型参数上传到云中心进行聚合。

(3) 云层。云层的云中心有两个主要功能: ①协助ENs之间的协作并为它们部署智能合约, 这些智能合约可以自动执行; ②聚合每个参与聚合的EN的模型参数。具体来说, 当一个EN发送结果共享或计算迁移的请求时, 云中心将在验证其身份合法后向整个区块链网络广播其请求, 并向其返回一个可交易的邻居EN的地址。同时, 为它们之间的交易部署相应的智能合约(交易规则)。此

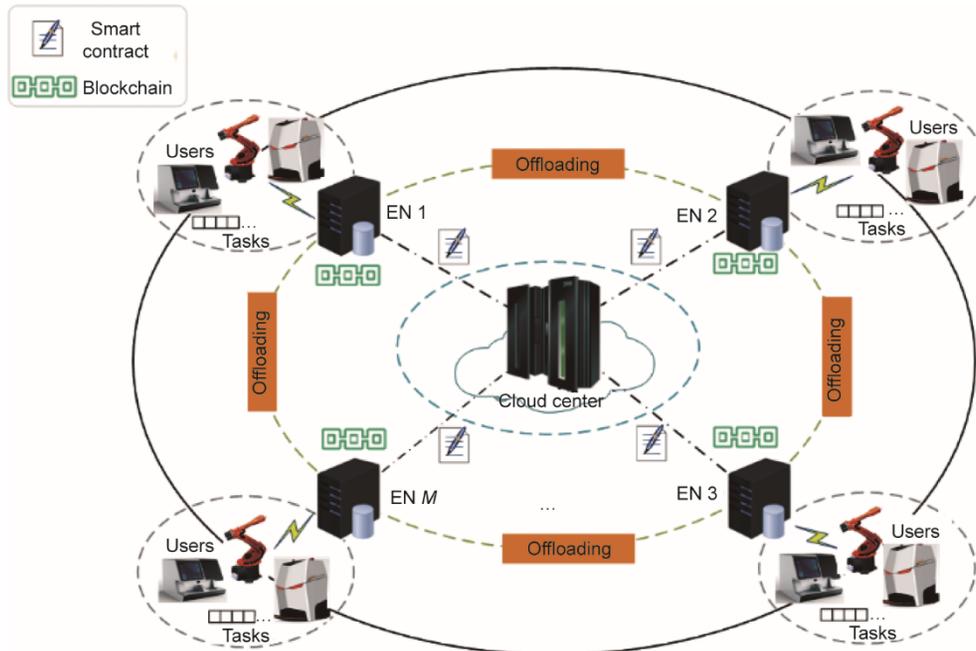


图1. 激励感知区块链辅助的智能边缘计算迁移与缓存模型。 M : 子区域的数量。

外，云中心负责聚合来自ENs的训练模型参数，然后将聚合后的参数下发给每个EN，以更新ENs中的本地模型。

3.1. 缓存模型

为了尽快将结果返回给用户并减轻EN的计算压力，将一些结果缓存在EN中是一种很好的方法。此外，缓存计算结果而不是任务本身不仅节省了缓存空间，而且提高了对用户原始数据的隐私保护。然而，由于EN的缓存空间有限，不可能缓存所有计算结果。因此，本文考虑在ENs之间共享结果，以缓解上述问题。此外，为了提高结果的缓存命中率，将根据其流行度和偏好在ENs中定期更换。将 α_m^f 定义为任务 f 的缓存决策，如果 $\alpha_m^f=1$ ，这意味着将任务 f 的结果缓存在EN m 中；否则 $\alpha_m^f=0$ 。任务的全局流行度和EN偏好的详细模型定义如下。

首先，EN会提前将高流行度的计算结果缓存在其缓存池中。整个网络的任务的全局流行度遵循Mandelbrot Zipf分布[31]，其定义为：

$$P_f = \frac{(O_f + \tau)^{-\sigma}}{\sum_{i \in \Gamma} (O_i + \tau)^{-\sigma}}, \quad \forall f \in \Gamma \quad (1)$$

式中， P_f 是任务 f 的全局流行度； O_f 和 O_i 分别表示任务 f 和任务 i 的结果按全局流行度降序排列的等级； τ 为平稳因子； σ 为偏度因子； i 表示 Γ 中的第 i 类任务。

对于EN m ，本文假设接收到任务 f 的服务请求的概率为 δ_m^f ($\sum_{m \in \mathcal{M}} \delta_m^f = 1$)，并且任务 f 的偏好是 $P_f^m = N_m^f / N$ ，其满足 $\sum_{f \in \Gamma_m} P_f^m = 1$, $P_f^m = P_f \delta_m^f$ ，其中， N_m^f 是EN m 接收到任务 f 的数量。本文将EN m 收到的所有任务的数量集表示为 $N_m = (N_m^f)^{1 \times F}$ 。

当结果未缓存在本地而是缓存在邻居EN时，本地EN会为了获取结果向可交易的邻居EN n 支付相应的结果共享报酬，即结果共享成本。关于任务 f 的结果共享成本 R_f^{cache} 的具体定义如下：

$$R_f^{\text{cache}} = \lambda_f^{\text{cache}} e^{-P_f^m} k_f, \quad n \in \mathcal{M} \setminus \{m\} \quad (2)$$

式中， λ_f^{cache} 表示结果共享支付的权重系数。具体地， $e^{-P_f^m}$ 为单位结果的共享成本，它与EN n 中任务 f 的偏好成反比， k_f 是任务 f 的结果大小。

3.2. 本地模型

当没有任何一个EN缓存任务 f 的结果时，就需要对任务 f 做迁移决策 $x_{m,f}^n$ 。如果 $\forall n \in \mathcal{M} \setminus \{m\}$, $x_{m,f}^n = 0$ ，则任务 f 会在本地EN m 计算，本文将任务 f 的大小定义为 S_f (bits)，将完成任务 f 所请求的中央处理器 (CPU) 周期数定义为 d_f (cycles)，将计算结果的大小定义为 k_f (bits)。任

务 f 在本地EN m 的完成时间可定义为：

$$t_f^m = \frac{d_f}{\beta_m^f u_m} \quad (3)$$

式中， u_m (cycles/s) 是EN m 的计算能力； β_m^f 表示为分配给任务 f 的计算资源占比。

因此，任务 f 的计算能耗 (e_f^m) 如下：

$$e_f^m = p_m^c t_f^m \quad (4)$$

式中， p_m^c 是EN m 的计算能力。因此，本地完成任务 f 的成本 (R_f^{local}) 可以表示为：

$$R_f^{\text{local}} = \left(1 - \sum_{n \in \mathcal{M} \setminus \{m\}} x_{m,f}^n \right) (\lambda_f^t t_f^m + \lambda_f^e e_f^m), \quad n \in \mathcal{M} \setminus \{m\} \quad (5)$$

式中， λ_f^t 和 λ_f^e 分别是时间和能量成本的权重系数。

3.3. 迁移模型

当 $x_{m,f}^n = 1$ 时，任务 f 从EN m 迁移到邻居EN n 。特别地，当EN m 和 n 在同一时刻 t 均从各自的本地用户接收到任务 f 的服务请求时，EN m 不需要将任务 f 发送到EN n 。这样，EN m 只需要向提供计算服务的邻居EN n 支付一定的服务报酬 R_f^{com} ，其定义为：

$$R_f^{\text{com}} = \lambda_f^{\text{com}} e^{-P_f^m} d_f \quad (6)$$

式中， λ_f^{com} 表示计算服务共享成本的权重系数。

否则，任务 f 需要传输到邻居EN n ，其传输时间和相应的能耗定义如下：

$$t_f^n = \frac{S_f}{v_{m,f}^n}, \quad f \notin \Gamma_m \cap \Gamma_n \quad (7)$$

$$e_f^n = p_m^t \frac{S_f}{v_{m,f}^n}, \quad f \notin \Gamma_m \cap \Gamma_n \quad (8)$$

式中， p_m^t 是EN m 的传输功率，并且 $v_{m,f}^n$ 表示任务 f 在EN m 和 n 之间的传输速率，定义为：

$$v_{m,f}^n = \gamma_m^f B_m^n \log_2 \left(1 + \frac{p_m^t h_m^2}{w_m^n \gamma_m^f B_m^n} \right) \quad (9)$$

式中， γ_m^f 是分配给任务 f 的带宽占比； B_m^n 是EN m 和 n 之间的带宽大小； h_m 和 w_m^n 分别是EN m 的信道增益和噪声； l_m^n 是 m 和 n 之间的距离。

因此，任务 f 的迁移成本表示为：

$$R_{m,f}^n = \sum_{n \in \mathcal{M} \setminus \{m\}} x_{m,f}^n \times \begin{cases} \lambda_f^t t_f^n + \lambda_f^e e_f^n + R_f^{\text{com}}, & f \notin \Gamma_m \cap \Gamma_n \\ R_f^{\text{com}}, & f \in \Gamma_m \cap \Gamma_n \end{cases} \quad (10)$$

4. 问题的定义

根据上述模型构建和相关定义，在这个激励感知区块链辅助的边缘计算网络中，作为衡量系统性能的重要指标，完成任务 $f \in \Gamma_m$ 的成本 C ($\alpha_m^f, x_{m,f}^n, \beta_m^f, \gamma_m^f$) 包括延迟、能

耗和协作成本，其定义为：

$$C(\alpha_m^f, x_{m,f}^n, \beta_m^f, \gamma_m^f) = (1 - \alpha_m^f)(1 - \alpha_n^f)(R_f^{\text{local}} + R_{m,f}^n) + \alpha_n^f R_f^{\text{cache}} \quad (11)$$

式中， α_m^f 和 α_n^f 分别表示任务 f 的结果在EN m 和邻居EN n 中的缓存状态， α_m^f 表示为：

$$\alpha_m^f = \begin{cases} 1, & \alpha_m^f(t-1) = 1 \\ 0, & \alpha_m^f(t-1) = 0 \end{cases} \quad (12)$$

式中， $\alpha_m^f(t-1)$ 是任务 f 在前一个时刻的缓存决策，则所有任务的缓存决策集表示为 $a_m = (\alpha_m^f)^{1 \times F}$ 。

因此，构建优化目标如等式(13)所示，其旨在最小化完成EN m 接收的所有任务的总成本。通过联合优化缓存决策 α_m^f 、迁移决策 $x_{m,f}^n$ 以及计算和通信资源的分配比例(即 β_m^f 和 γ_m^f)，优化问题定义如下：

$$\min_{\alpha_m^f, x_{m,f}^n, \beta_m^f, \gamma_m^f} \sum_{f \in \Gamma_m} C(\alpha_m^f, x_{m,f}^n, \beta_m^f, \gamma_m^f) \quad (13)$$

$$\text{s.t. } t_m^f \leq t_m^{\max} \quad (13a)$$

$$e_m^f \leq e_m^{\max} \quad (13b)$$

$$\sum_{f \in \Gamma_m} \alpha_m^f k_f \leq C_m \quad (13c)$$

$$\sum_{f \in \Gamma_m} (1 - x_{m,f}^n) \beta_m^f \leq 1 \quad (13d)$$

$$\sum_{f \in \Gamma_m} x_{m,f}^n \gamma_m^f \leq 1 \quad (13e)$$

$$\sum_{n \in \mathcal{M}} x_{m,f}^n \leq 1 \quad (13f)$$

$$\alpha_m^f x_{m,f}^n \neq 1 \quad (13g)$$

$$\alpha_m^f, x_{m,f}^n \in \{0, 1\} \quad (13h)$$

为了确保服务的质量和效率，完成任务 f 的延迟和能耗应该被约束[即约束(13a)和(13b)]。由于EN的有限资源(缓存、计算和带宽)被多个任务共享，因此给出了约束(13c)~(13e)，即分配给所有任务的网络资源总和不超过最大值。从实际的角度出发，约束(13f)表明任务 f 最多只能迁移到一个邻居EN，而约束(13g)意味着任务 f 的缓存状态和迁移决策不能同时为1。约束(13h)表示缓存决策 α_m^f 和迁移决策 $x_{m,f}^n$ 的取值均为0或1。

显然，上述优化问题是一个混合整数非线性优化问题，当网络环境动态变化，用户数量随着时间的推移而增加时，变量维数会很高。因此，传统的方法很难直接求解，如梯度下降数学规划方法和交替方向乘子法。近年来，机器学习成为解决动态时变网络中这类资源分配和决策优化问题的一种流行且重要的手段，并被视为一种高效

的智能求解方法。鉴于DRL和FL分别在复杂动态网络环境中的自主学习和安全分布式计算的特点，本文将整合这两种机器学习思想来解决优化问题[式(13)]。

5. BICC-FDRL 算法

本节提出了一种新的求解算法(即BICC-FDRL算法)，以实现物联网中的智能边缘缓存与计算迁移。受区块链激励、优势行动者-评论家(advantage actor-critic, A2C)和FL思想的启发，所提出的算法主要包括两部分：激励感知区块链辅助的本地训练和基于贡献的联邦聚合，其详细框架如图2所示。

首先，作为一个独立的智能体，每个EN通过结合A2C概念，使用本地信息来训练本地模型。在本地训练期间，研究了一种激励感知区块链辅助的协作机制，以选择合适的协作邻居EN，这将以有益且安全的方式提高EN参与的积极性。本文假设EN m 中的本地训练总轮次为 T_m ，在EN m 完成 T_m 训练轮次后，所有ENs将同时向云中心发送其本地模型参数和相应的平均累积奖励，用于聚合操作和全局模型更新。云中心计算每个EN的贡献比重，作为其模型参数的聚合权重，然后聚合参数以获得新的全局模型参数。最后，全局模型参数被下发到所有ENs，以更新它们的本地模型。下面将对上述方案进行详细的设计和说明。

5.1. 激励感知区块链辅助的本地训练

通常，在EN中的联合优化问题可以被视为一个马尔可夫决策过程。它模拟一个智能体(即EN)根据当前状态在环境中执行动作，从而改变环境状态并获得奖励。该过程包括三个关键要素，定义如下。

(1) **状态**。EN m 在时刻 t 的状态空间被定义为 $S_m(t) = \{N_m(t), a_m(t), \psi_m(t), \mathfrak{S}_m(t)\}$ ，其中， $N_m(t)$ 代表被EN m 接收到的所有任务数量， $a_m(t)$ 是EN m 中所有任务的缓存状态， $\psi_m(t)$ 和 $\mathfrak{S}_m(t)$ 分别表示EN m 中剩余的缓存和计算资源，其定义分别为：

$$\psi_m(t) = C_m - \sum_{f \in \Gamma_m} \alpha_m^f k_f \quad (14)$$

$$\mathfrak{S}_m(t) = 1 - \sum_{f \in \Gamma_m} (1 - x_{m,f}^n) \beta_m^f \quad (15)$$

(2) **动作**。本文将EN m 在时刻 t 的动作空间描述为 $A_m(t) = \{a_m(t), x_m(t), \beta_m(t), \gamma_m(t)\}$ ，它由缓存决策集 $a_m(t)$ 、迁移决策集 $x_m(t)$ 以及分配的計算和带宽资源集 $\beta_m(t)$ 和 $\gamma_m(t)$ 组成。它们分别被定义为：

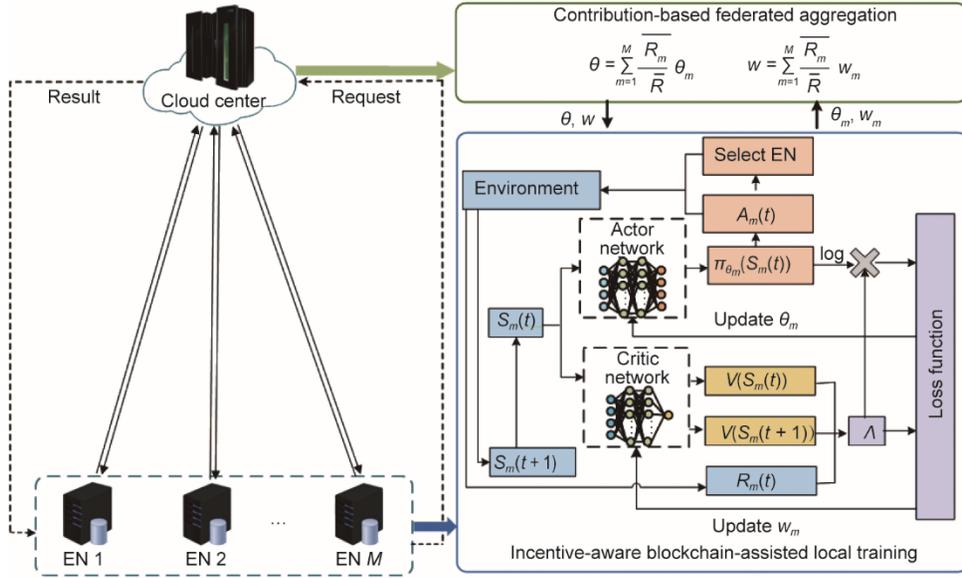


图2. BICC-FDRL的框架。 \bar{R}_m : 在EN m 中完成任务的平均累积奖励; \bar{R} : 所有ENs的平均累积奖励之和; θ_m : 在EN m 中行动者网络的参数; w_m : EN m 中评论家网络的参数; θ : 全局模型中行动者网络的参数; w : 全局模型中评论家网络的参数; $S_m(t)$: EN m 在时刻 t 的状态空间; $S_m(t+1)$: EN m 在时刻 $t+1$ 的状态空间; $A_m(t)$: EN m 在时刻 t 的动作空间; $\pi_{\theta_m}(S_m(t))$: 行动者网络生成的策略; $R_m(t)$: EN m 在时刻 t 的累积奖励; $V(S_m(t))$: 当前状态的值函数; $V(S_m(t+1))$: 下一状态的值函数; Λ : 优势函数。

$$\alpha_m(t) = \{\alpha_m^1(t), \alpha_m^2(t), \dots, \alpha_m^F(t)\} \quad (16)$$

$$x_m(t) = \{x_{m,1}^m(t), x_{m,2}^m(t), \dots, x_{m,F}^m(t)\} \quad (17)$$

$$\beta_m(t) = \{\beta_m^1(t), \beta_m^2(t), \dots, \beta_m^F(t)\} \quad (18)$$

$$\gamma_m(t) = \{\gamma_m^1(t), \gamma_m^2(t), \dots, \gamma_m^F(t)\} \quad (19)$$

(3) 奖励。EN的学习目标是使累积回报最大化，而本文优化问题的目标是使总成本最小化。因此，本文将即时回报设置为总成本的负指数，即：

$$r_m(t) = \begin{cases} -\sum_{j=1}^{\text{cost}_j} \text{cost}_j, & \text{if Eq.(13a)–Eq.(13h)} \\ \varepsilon, & \text{else} \end{cases} \quad (20)$$

式中, $\text{cost}_j = C(\alpha_m^j, x_{m,j}^n, \beta_m^j, \gamma_m^j)$, 并且 ε ($\varepsilon < 0$)是由具体环境给出的惩罚值。因此，累积奖励为 $R_m = \sum_{t=1}^T r_m(t)$ 。

在本方案中，每个EN充当DRL智能体，并基于本地信息训练其模型。受A2C概念的启发，图2中本地模型的网络架构由两个子网络组成（即行动者网络和评论家网络）。行动者网络将根据当前状态选择动作，然后与环境进行交互。然后，环境将给予奖励 $r_m(t)$ 并转移到下一个状态。特别地，奖励 $r_m(t)$ 是根据动作 $A_m(t)$ 和激励感知区块链辅助的协作机制获得的。评论家网络负责根据当前和下一个状态对行动者网络进行动作评估。与传统的行动者-评论家设计不同，本方案的评论家网络使用优势函数而不是简单的价值函数来评估行动者的表现。因为对行动的评估不仅取决于动作的好坏，还取决于动作对环境改进的效果。

在时刻 t ，行动者网络基于本地EN m 的当前状态 $S_m(t)$ 生成策略 $\pi_{\theta_m}(S_m(t))$ 。由于行动者网络的输出是连续值，而 $A_m(t)$ 的缓存和迁移决策是离散变量。因此，本文需要近似策略 $\pi_{\theta_m}(S_m(t))$ ，以便将其中的缓存和迁移决策转换为离散值，其定义为：

$$\alpha_m(t) = x_m(t) = \begin{cases} 0, & \alpha_m(t), x_m(t) < 0.5 \\ 1, & \alpha_m(t), x_m(t) \geq 0.5 \end{cases} \quad (21)$$

为了激励每个EN以有益和安全的方式参与结果共享和计算迁移过程，本文研究了一种激励感知的区块链辅助协作机制，如图3所示。该机制将选择一个可以向本地EN提供服务的邻居EN，并且 $x_{m,f}^n$ 的值也将被确定。

根据当前的状态 $S_m(t)$ 和动作 $A_m(t)$ ，当结果没有缓存在本地EN ($\alpha_m^f = 0$)中或本地计算不能满足用户需求 ($x_{m,f}^n = 0$)时，可交易的邻居EN将被选中以获取任务结果，然后本地EN将向提供服务的邻居EN支付相应的报酬。因此，奖励 $r_m(t)$ 是基于 $A_m(t)$ 和该协作机制的成本来计算的。与其他基于区块链的激励方法不同，该方案适用于本文考虑的此类场景，即该方案的实现与边缘缓存和计算迁移密切相关，并且它们相互促进。本方案考虑了ENs之间的两种协作情况，具体如下：

(1) 结果共享。通过鼓励每个EN缓存其偏好的任务结果，这种激励感知区块链辅助的协作不仅可以满足本地需求，还可以帮助其他邻居，从而提高缓存利用率。

当结果未缓存在本地EN中时，本地EN将向云中心

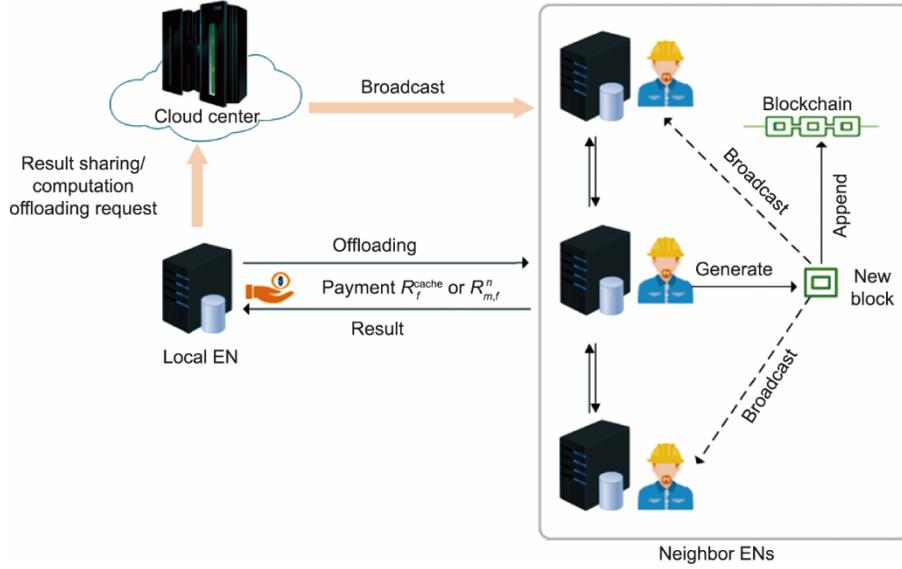


图3. 激励感知区块链辅助的协作机制。

发送结果共享请求。云中心将在验证其身份合法后，向整个区块链网络广播其请求，以便知道哪些邻居ENs缓存了此结果，即 $a_n^f(t)=1, n \in \mathcal{M} \setminus \{m\}$ 。这些邻居节点将成为候选节点，并根据任务 f 的偏好从低到高进行排序。

随后，云中心为交易部署智能合约。为了节省尽可能多的成本，最终将会选择具有最低结果共享成本 R_f^{cache} 的邻居EN n 。根据等式 (2)， R_f^{cache} 是 P_f^n 的负指数函数。换言之，EN m 将选择对任务 f 具有更高偏好的邻居EN n 。此时， $r_m(t)$ 中的 cost_f 被定义为 $\text{cost}_f = R_f^{\text{cache}}$ 。

此后，ENs 完成基于智能合约的共享结果的合法交易，邻居EN n 将事务处理过程记录为一个块。最后，它将向所有候选人广播这一块，以达成共识。与向所有ENs 广播相比，这种方法减少了通信开销。

(2) 计算迁移。通过鼓励本地EN选择具有类似偏好或更充足计算资源的可交易邻居EN来进行计算迁移，这种激励感知区块链辅助的协作将提高任务完成效率。

当本地EN m 中的任务 f 被迁移到相邻EN进行计算时（即 $\sum_{n \in \mathcal{M} \setminus \{m\}} x_{m,f}^n = 1$ ），提供计算服务的邻居EN将获得一定的报酬，这与结果共享的过程类似，但是选择候选者的方式不同。邻居ENs 根据可用计算资源量按降序排列，并且将超过平均计算资源的邻居被视为候选者。然后，具有最低迁移成本的 $R_{m,f}^n$ 邻居EN n 会被选择。根据等式 (10) 可知，接收到相同请求并且具有对任务 f 更高偏好的邻居EN 更有可能被选择。此时， $r_m(t)$ 中的 cost_f 被定义为 $\text{cost}_f = R_{m,f}^n$ 。

在上述过程之后，环境给出奖励 $r_m(t)$ ，并且环境状

态转移到下一个状态 $S_m(t+1)$ 。然后将当前状态和下一个状态输入到评论家网络中，输出结果为它们的值函数 $V(S_m(t))$ 和 $V(S_m(t+1))$ 。由此可以将优势函数 Λ 定义为：

$$\Lambda = Q(t) - V(S_m(t)) \quad (22)$$

$$Q(t) \approx R_m(t) + \rho V(S_m(t+1)) \quad (23)$$

式中， $Q(t)$ 表示在当前 $S_m(t)$ 状态下执行动作 $A_m(t)$ 后的期望值函数； ρ 是折扣因子。如果 $\Lambda > 0$ ，那么这意味着执行动作 $A_m(t)$ 将给系统状态带来积极的激励，并且 Λ 值越大，网络中的改进就越大。此外，利用优势函数可以减少评估中的方差，加快收敛速度。

训练过程中使用Adam优化器来更新模型参数。行动者网络中的策略损失函数的梯度表示为 $A \nabla_{\theta_m} \lg \pi_{\theta_m}(S_m(t))$ ，评论家网络中的价值损失函数的梯度表示为 $A \nabla_{w_m} V(S_m(t))$ 。最后，这两个网络的更新过程为：

$$\theta_m \rightarrow \theta_m + \eta A \nabla_{\theta_m} \lg \pi_{\theta_m}(S_m(t)) \quad (24)$$

$$w_m \rightarrow w_m + \mu A \nabla_{w_m} V(S_m(t)) \quad (25)$$

式中， η 和 μ 分别是行动者网络和评论家网络的学习率。

5.2. 基于贡献的联邦聚合

FL 中的一个重要步骤是模型参数的聚合。即当每个EN 都经过了几轮训练后，参与FL 的ENs 同时将其训练的模型参数上传到云中心进行聚合。

最经典的聚合方法是谷歌提出的联邦平均 (FedAvg) 聚合[32]，该方法基于EN 的训练样本量采用简单的平均聚合。FedAvg 的模型参数聚合操作如下所示：

$$\theta = \sum_{m=1}^M \frac{K_m}{M} \theta_m \quad (26)$$

$$w = \sum_{m=1}^M \frac{K_m}{M} w_m \quad (27)$$

式中, θ 和 w 分别是全局模型中行动者网络和评论家网络的参数。本文假设有 M 个 ENs 参与 FL, 并且 EN m 的样本量为 K_m 。

通常, 样本量被用作聚合的基础, 因为更多的训练样本将带来更好的模型训练效果。然而, 这并非必然。为了更准确地描述 ENs 的模型训练效果, 本文考虑使用更直接的方法 (R_m) 即累积奖励。累积奖励越高, 表明 EN 的网络模型参数越好, 并将对未来全局模型的性能改进做出更大的贡献。

受此启发, 本文设计了一种基于贡献的联邦聚合方法, 该贡献指的是 EN 的平均累积奖励占比 \bar{R}_m/\bar{R} 。在联邦聚合过程中, 每个 EN 需要将其贡献和模型参数上传到云中心, 云中心聚合这些参数以获得全局模型的参数, 具体定义为:

$$\theta = \sum_{m=1}^M \frac{\bar{R}_m}{\bar{R}} \theta_m \quad (28)$$

$$w = \sum_{m=1}^M \frac{\bar{R}_m}{\bar{R}} w_m \quad (29)$$

式中, $\bar{R}_m = R_m/|I_m|$ 是完成任务的平均累积奖励, 并且 $\bar{R} = \sum_{m=1}^M \bar{R}_m$ 。此外, 该贡献的定义避免了由于不同 ENs 中任务数量不同而导致的不公平评估。

在聚合过程之后, 云中心会将全局模型的聚合参数下发给每个 EN, 以更新其本地模型。每个 EN 不断学习, 直到模型收敛到最优策略。

为了更清楚地理解优化目标的求解过程, 本文将其总结为算法 1。

6. 性能评估

本节将通过仿真实验来评估所提出方案的性能, 主要包括两个部分: 6.1 小节中的 BICC-FDRL 算法的收敛性分析和 6.2 小节中的方案优势分析。

为了构建仿真实验, 本实验设置了 $M=7$ 个 ENs 和 $N=25$ 个用户, 并且都在每个 EN 的通信范围内。为了方便起见, 每个 EN 具有相同的计算能力和缓存大小 (分别为 $2.5 \text{ Gcycles}\cdot\text{s}^{-1}$ 和 6 Mb), 并且 ENs 之间的带宽统一设置为 100 MHz 。在时刻 t , 整个用户层生成了 $F=50$ 类任务, 任务大小的范围为 $[30,50] \text{ Mb}$, 并且其结果大小的范围是 $[300,500] \text{ Kb}$ 。此外, 任务流行度的平稳因子和偏度因子

算法 1 用于边缘缓存和计算迁移的 BICC-FDRL 算法

输入: 本地训练总轮次 T_m

全局训练的总次数 T

任务 P_f 的全局流行度

任务 f 的服务请求的概率 δ_f^i

输出: 具有最优 A_m^* 的最小总成本

初始化: $\theta_m = \theta, w_m = w$

For episode $\ell=1$ to T do

For each EN $m \in \mathcal{M}$ do

For $t=1$ to T_m do

基于 $S_m(t)$ 执行 $A_m(t)$

基于激励感知区块链辅助协作机制的 $A_m(t)$ 和 $cost_f$ 选择协作邻居 EN n

获得 $r_m(t)$ 并转移到下一个状态 $S_m(t+1)$

执行 $S_m(t) = S_m(t+1)$

计算 Λ , 策略损失函数的梯度 $\Delta \nabla_{\theta_m} \log \pi_{\theta_m}(S_m(t))$ 以及价值损失函数梯度 $\Delta \nabla_{w_m} V(S_m(t))$

根据公式(24)和(25)分别更新 θ_m 和 w_m

End for

End for

基于等式(28)和(29)执行基于贡献的联邦聚合

将全局模型参数下发到每个 EN, 即 $\theta_m = \theta, w_m = w$

End for

获得具有最佳 A_m^* 的最小总成本

分别被设置为 $\tau=-0.95$ 和 $\sigma=0.50$ 。其中, 本实验设定 ENs 具有不同的偏好和需求, 这种差异化的考虑可以反映 ENs 的异构性。

对于每个 EN, 其行动者网络和评论家网络都是三层全连接的神经网络, 具有 64 个单元的隐藏层使用校正线性单元 (ReLU) 激活函数。在行动者网络的输出层中, 迁移和缓存决策使用 softmax 作为其激活函数, 而其他动作使用 sigmoid 激活函数。此外, 本实验添加了一个两层全连接的神经网络作为行动者和评论家网络的共享输入层, 用于抽象复杂的系统状态。DRL 的计算复杂度主要取决于神经网络的结构和参数数量, 这是通过浮点运算 (floating-point operations, FLOPs) 来衡量的。在本文的算法中的整个神经网络需要 20 万 FLOPs 来处理一次状态输入。

6.1. BICC-FDRL 算法的收敛性分析

在图 4 中, 本文研究了行动者网络在不同学习率下累积奖励的收敛性。从图中可以看出, $\eta=10^{-5}$ 曲线的收敛速度明显低于学习率较高的其他两条曲线。虽然累积奖励 $\eta=10^{-4}$ 的收敛速度慢于 $\eta=10^{-3}$ 时的收敛速度, 但是前者更稳定, 并且它们的最终收敛值非常接近。事实上, 经过

多次实验发现，学习率过快会陷入局部最优解，而学习率过慢在有限时间内不会收敛到最优解。因此，在下面的实验中将行动者网络的学习率设置为 10^{-4} 。

不同网络学习率对损失收敛的影响如图5所示。损失函数值可以反映模型的训练效果和稳定性。随着模型训练的不断改进，损失值减少，并最终稳定在一个较小值（理想损失为0）。当 $\mu=0.050$ 和 $\mu=0.010$ 时，它们的损失收敛性能明显优于 $\mu=0.001$ 时的损失收敛性能。与上述分析类似，在一定范围内，较高的学习率具有更好的收敛性能。最后，经过多次实验，设定 $\mu=0.050$ 。

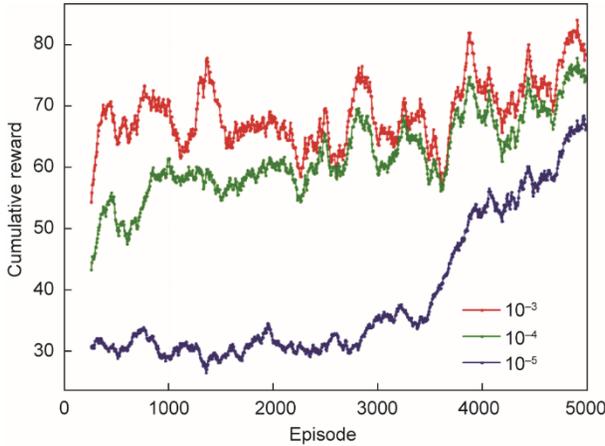


图4. 不同学习率下行动者网络的奖励收敛性。

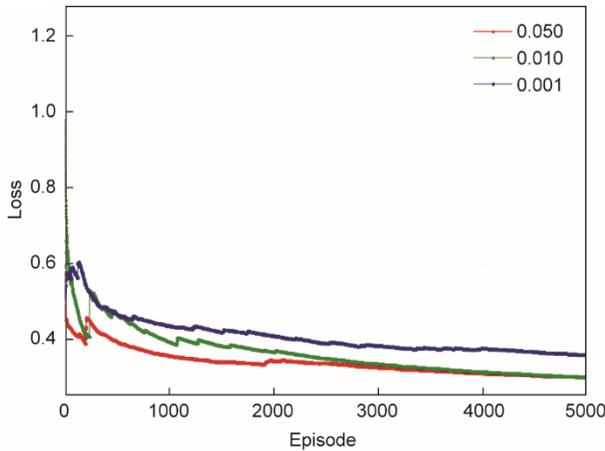


图5. 不同学习率下评论家网络的损失收敛性。

不同聚合方法的损失收敛情况如图6所示。从两种聚合方法的曲线来看，本文提出的基于贡献的联邦聚合的损失低于FedAvg的损失，这证明了本文的聚合方法具有较好的模型训练效果。因为在本文的聚合方法中，学习效果好的EN具有很大的参考价值，它对系统的贡献更大，因此应具有更大的聚合权重，这将提高未来全局训练效果，使策略更准确。

不同数量的聚合ENs的训练效果如图7所示，即参与联邦聚合的节点数量为3、5和7。有7个ENs参与的模型具有最高的累积奖励。此外，根据三条曲线的振荡幅度可知，更多的ENs参与FL将使模型训练更加稳定。随着ENs数量的增加，全局模型在整个网络中学习到的信息（即当有更多的训练样本可用时，训练效果更好）。然而，如果有太多的ENs参与联邦聚合并将其模型参数上传到云中心，就会产生通信负担。通过更多的实验发现，当ENs的数量达到一定值时，模型的收敛性能没有显著变化。在本文的网络场景中，设置 $M=7$ 个ENs。

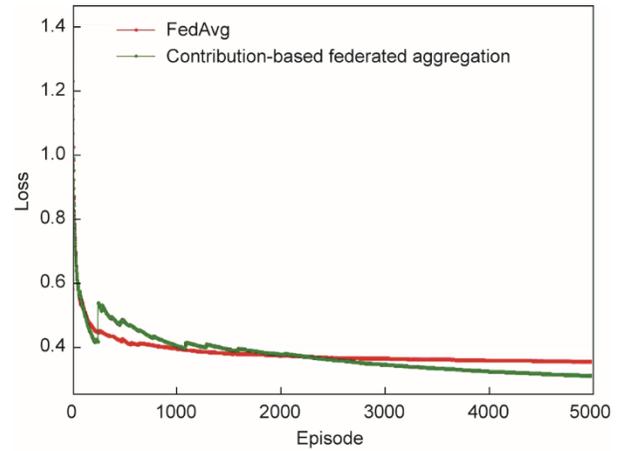


图6. 不同聚合方法的损失收敛性。

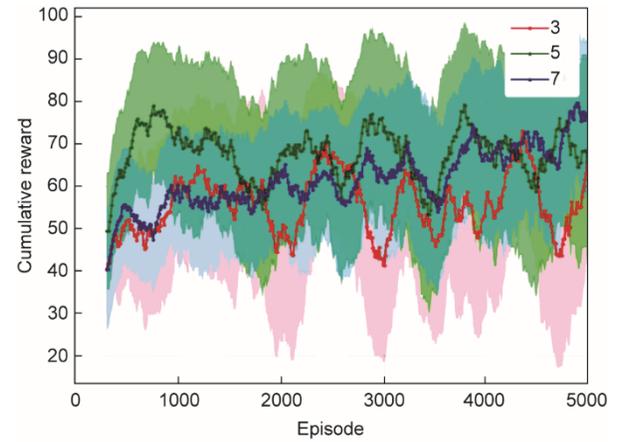


图7. 不同数量的聚合ENs的训练效果。

6.2. 方案优势分析

本子节将通过与其他三种缓存方案进行比较，即最近最少使用（LRU）、最近最少使用-2（LRU-2）和最不经常使用（LFU），来分析本文提出的缓存方案的性能优势。

由于命中率和总成本是表现缓存方案性能的关键评估指标，以下仿真将从这两个方面分析四种方案的缓存性能。命中率 H 的定义如下：

$$H = \frac{\sum_{f \in \Gamma_m} \varphi_f}{|\Gamma_m|} \quad (30)$$

其中,

$$\varphi_f = \begin{cases} 0, & \sum_{m \in \mathcal{M}} \alpha_m^f = 0 \\ 1, & \text{else} \end{cases} \quad (31)$$

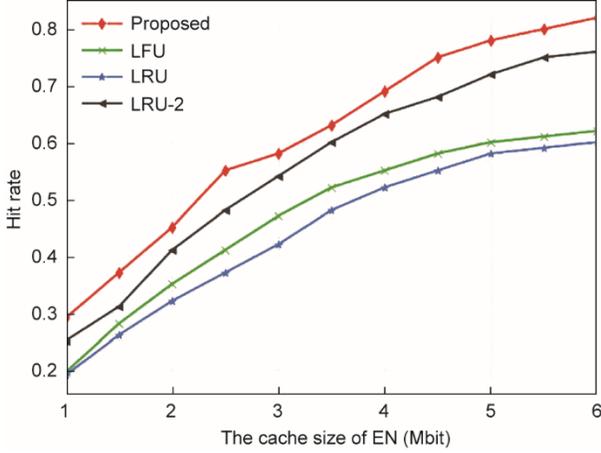


图8. 不同缓存方案的命中率。

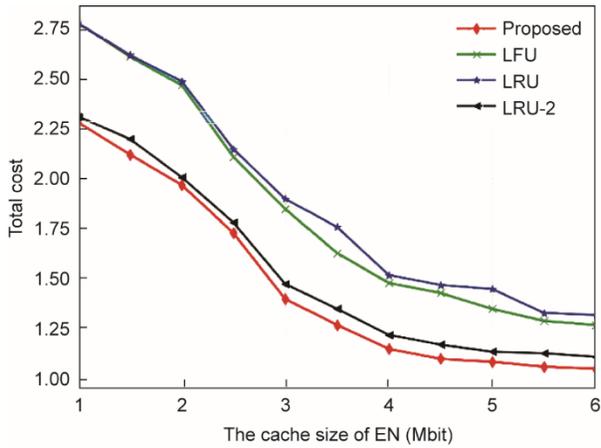


图9. 不同缓存方案的总成本。

图8显示了不同EN缓存大小的不同缓存方案的命中率。随着缓存大小的增加, EN可以缓存更多的结果, 因此四种方案的命中率逐渐增加。命中率从高到低依次为本文提出的方案、LRU-2、LFU和LRU。LRU的命中率最低, 这主要是因为流行结果被偶然的不流行结果所取代。排名第三的LFU缓存请求频率高的结果, 这表明偏好很高, 但需要很长时间才能很好地学习。排名第二的LRU-2结合了LRU和先进先出的优势, 但它仍然无法准确地缓存流行的结果。本文提出的方案命中率最高, 这是因为它能够基于结果流行度和不同EN区域的偏好来缓存结果。此外, 本文方案设计了具有激励感知区块链辅助的协作机制, 它激发了ENs共享结果的意愿, 从而提高了命中率。

图9描述了不同缓存方案的总成本, 其中, EN的缓存大小从1 Mb到6 Mb。对应于图8, EN的缓存空间越大, 命中率越高, 这节省了完成任务的成本。因此, 四种方案的总成本呈下降趋势, 本文提出的方案与其他方案相比是最低的。此外, 当缓存大小达到一定值时, 命中率达到收敛, 因此总成本将稳定下来。

根据以上仿真可知, 本方案具有更好的缓存性能, 能够显著降低总成本。

为了从更广泛的角度分析本方案的优势, 本文将其与三种基准方案[即基于联邦深度强化学习的协作边缘缓存(FADE) [33]、高效灵活的管理: 联邦深度Q网络方法(EFMFDQN) [34]和集中式A2C进行了比较。在FADE中, 它采用了类似于本文的基于流行度的缓存方法, 但不考虑迁移决策和其他资源优化。EFMFDQN通过基于深度Q网络的FL算法联合优化迁移决策、带宽分配比和传输功率。需要注意的是, 集中式A2C是理想条件下总成本最小的最优基准方案。具体地, 它将云中心视为一个智能体, 收集整个网络的信息集中训练模型, 并部署A2C算法优化总奖励。

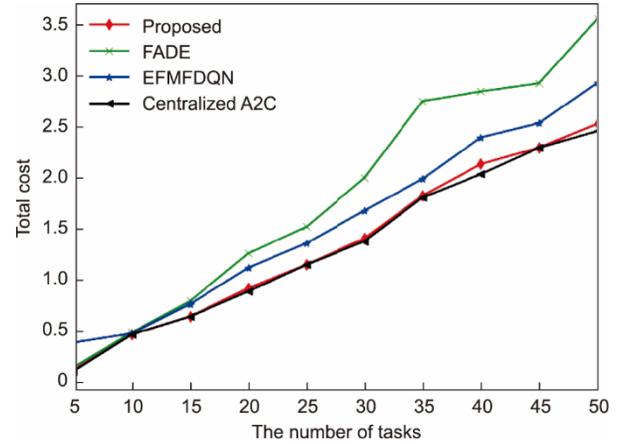


图10. 不同方案在不同任务数量下的总成本。

对于上述四种方案, 在图10中比较了它们在不同任务数量下的总成本。本方案的性能接近集中式A2C, 并且优于其他两种方案。具体而言, 当任务很少时, 所有ENs的缓存资源足以缓存这些任务的结果, 而无需本地或迁移计算。因此, 其他三种方案的总成本低于未考虑缓存优化的EFMFDQN。随着任务数量的增加, 本方案的优势越来越明显。由于EN的缓存空间有限, 因此仅考虑缓存决策优化的FADE总成本最高。这一现象是因为本文提出的方案更全面地考虑了优化因素(即迁移决策、缓存决策、计算和通信资源分配的联合优化)。此外, 本文设计的激励感知区块链辅助的协作机制, 可促进ENs之间的协作并降

低协作成本。

从更直观的角度来看，本方案的优势可以通过优化效用来分析，四种方案的优化效用定义如下：

$$U_{\text{BICC-FDRL}} = \frac{C_{\text{local}} - C_{\text{BICC-FDRL}}^*}{C_{\text{local}}} \quad (32)$$

$$U_{\text{FADE}} = \frac{C_{\text{local}} - C_{\text{FADE}}^*}{C_{\text{local}}} \quad (33)$$

$$U_{\text{EFMFDQN}} = \frac{C_{\text{local}} - C_{\text{EFMFDQN}}^*}{C_{\text{local}}} \quad (34)$$

$$U_{\text{centralized A2C}} = \frac{C_{\text{local}} - C_{\text{centralized A2C}}^*}{C_{\text{local}}} \quad (35)$$

式中， $C_{\text{BICC-FDRL}}^*$ 、 C_{FADE}^* 、 C_{EFMFDQN}^* 和 $C_{\text{centralized A2C}}^*$ 分别表示由BICC-FDRL、FADE、EFMFDQN和集中式A2C获得的最优总成本； C_{local} 是本地计算的总成本； $U_{\text{BICC-FDRL}}$ 、 U_{FADE} 、 U_{EFMFDQN} 和 $U_{\text{centralized A2C}}$ 分别是BICC-FDRL、FADE、EFMFDQN和集中式A2C的优化效用。

从图 11 可以观察到，当 ENs 的数量增加时，优化效用会增加。显然，ENs 数量的增加意味着网络中有更多的缓存和计算资源，这增强了 ENs 之间的协作，进而提高了资源利用率。此外，随着 ENs 数量的增加，参与联邦聚合的 ENs 数量也增加，类似于图 7，训练效果将得到提高。当网络中有 9 个 ENs 时，本方案的效用几乎等于集中式 A2C，并且明显优于其他两个方案。该仿真表明本方案高效地利用了缓存、通信和计算资源。

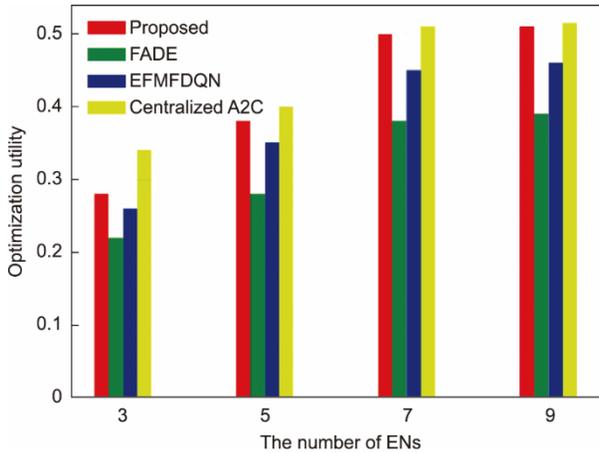


图 11. 不同方案在不同网络规模下的优化效用。

值得注意的是，尽管在图 10 和图 11 中集中式 A2C 具有最佳结果，但本方案仍然有意义。具体来说，在本方案中，A2C 算法与 FL 相结合，它可以将训练下沉到 ENs，这减轻了云中心的计算压力和通信压力。同时，由于 FL 的训练模式，模型更新只传输模型参数，而不传输训练数据，具有隐私保护的效果。

7. 结论

本文提出了一种面向物联网的激励感知区块链辅助的智能边缘缓存和计算迁移方案。为了最小化 EN 中完成任务的总成本，本文对迁移、缓存和资源分配进行了全面优化。特别地，基于任务偏好的定价规则有利于节约成本。此外，为了获得最优解，本文提出了一种 BICC-FDRL 算法。该算法在本地训练阶段为鼓励 ENs 协作而设计了一种安全的激励机制，即激励感知区块链辅助的协作机制，所有 ENs 都积极参与结果共享和计算迁移，以获取收益。此外，本文设计了一种基于贡献的联邦聚合方法，对全局性能提升贡献较大的 ENs 被给予更大的聚合权重，从而实现训练效果的提升。数值结果表明，与其他基准方案（如 FADE 和 EFMFDQN）相比，本文提出的方案在命中率、总成本和优化效用方面具有显著的性能优势。本文提出的 BICC-FDRL 算法适用于所有 ENs 都具有同构学习模型的情况。然而，当它们的神经网络为异构学习模型时，这种 FL 方法具有局限性。由此，在未来的工作中将研究一种适用于异构模型的高效且个性化的 FL。

致谢

本研究得到国家自然科学基金(61971235)、中国博士后科学基金(2018M630590)、江苏省博士后科研基金项目(2021K501C)、江苏省 333 高层次人才培养工程、南京邮电大学 1311 人才计划、江苏省研究生科研创新计划(KY-CX22_1017)的部分资助。

Compliance with ethics guidelines

Qian Wang, Siguang Chen, and Meng Wu declare that they have no conflict of interest or financial conflicts to disclose.

References

- [1] Evans D. The Internet of Things: how the next evolution of the Internet is changing everything. Report. San Jose: CISCO; 2011.
- [2] Chen S, Zhu X, Zhang H, Zhao C, Yang G, Wang K. Efficient privacy preserving data collection and computation offloading for fog-assisted IoT. *IEEE Trans Sustain Comput* 2020; 5(4): 526–40.
- [3] Wu F, Liu X, Li H, Fan Q, Zhu L, Wang X, et al. Energy-time efficient task offloading for mobile edge computing in hot-spot scenarios. In: *Proceedings of the IEEE International Conference on Communications*; 2021 Jun 14–23; Montreal, QC, Canada; 2021.
- [4] Zhang J, Hu X, Ning Z, Ngai ECH, Zhou L, Wei J, et al. Joint resource

- allocation for latency-sensitive services over mobile edge computing networks with caching. *IEEE Internet Things J* 2019; 6(3): 4283–94.
- [5] Zhang L, Wu J, Mumtaz S, Li J, Gacanin H, Rodrigues JJPC. Edge-to-edge cooperative artificial intelligence in smart cities with on-demand learning offloading. In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*; 2019 Dec 9–13; Waikoloa, HI, USA; 2019.
- [6] Zhao L, Yang K, Tan Z, Song H, Al-Dubai A, Zomaya AY, et al. Vehicular computation offloading for industrial mobile edge computing. *IEEE Trans Ind Inform* 2021; 17(11): 7871–81.
- [7] Zeng F, Chen Q, Meng L, Wu J. Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing. *IEEE Trans Intell Transp Syst* 2021; 22(6): 3247–357.
- [8] Zhao Z, Feng C, Yang HH, Luo X. Federated-learning-enabled intelligent fog radio access networks: fundamental theory, key techniques, and future trends. *IEEE Wirel Commun* 2020; 27(2): 22–8.
- [9] Huang X, Leng S, Maharjan S, Yan Z. Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks. *IEEE Trans Veh Technol* 2021; 70(9): 9282–93.
- [10] Chen S, Zheng Y, Lu W, Varadarajan V, Wang K. Energy-optimal dynamic computation offloading for industrial IoT in fog computing. *IEEE Trans Green Commun Netw* 2020; 4(2): 566–76.
- [11] Malik R, Vu M. On-request wireless charging and partial computation offloading in multi-access edge computing systems. *IEEE Trans Wirel Commun* 2021; 20(10): 6665–79.
- [12] Liu Y, He Q, Zheng D, Zhang M, Chen F, Zhang B. Data caching optimization in the edge computing environment. In: *Proceedings of the IEEE International Conference on Web Services (ICWS)*; 2019 Jul 8–13; Milan, Italy; 2019. p. 99–106.
- [13] Chen Z, Zhou Z. Dynamic task caching and computation offloading for mobile edge computing. In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*; 2020 Dec 7–11; Taipei, China; 2020.
- [14] Bi S, Huang L, Zhang YJA. Joint optimization of service caching placement and computation offloading in mobile edge computing systems. *IEEE Trans Wirel Commun* 2020; 19(7): 4947–63.
- [15] Zhang G, Zhang S, Zhang W, Shen Z, Wang L. Joint service caching, computation offloading and resource allocation in mobile edge computing systems. *IEEE Trans Wirel Commun* 2021; 20(8): 5288–300.
- [16] Ma X, Zhou A, Zhang S, Wang S. Cooperative service caching and workload scheduling in mobile edge computing. In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*; 2020 Jul 6–9; Toronto, ON, Canada; 2020. p. 2076–85.
- [17] Zhong S, Guo S, Yu H, Wang Q. Cooperative service caching and computation offloading in multi-access edge computing. *Comput Netw* 2021; 189: 107916.
- [18] Feng H, Guo S, Yang L, Yang Y. Collaborative data caching and computation offloading for multi-service mobile edge computing. *IEEE Trans Veh Technol* 2021; 70(9): 9408–22.
- [19] Yuan P, Shao S, Geng L, Zhao X. Caching hit ratio maximization in mobile edge computing with node cooperation. *Comput Netw* 2021; 200: 108507.
- [20] Liu Y, Xu C, Zhan Y, Liu Z, Guan J, Zhang H. Incentive mechanism for computation offloading using edge computing: a stackelberg game approach. *Comput Netw* 2017; 129: 399–409.
- [21] Hou W, Wen H, Zhang N, Wu J, Lei W, Zhao R. Incentive-driven task allocation for collaborative edge computing in industrial Internet of Things. *IEEE Internet Things J* 2022; 9(1): 706–18.
- [22] Wang Q, Guo S, Wang Y, Yang Y. Incentive mechanism for edge cloud profit maximization in mobile edge computing. In: *Proceedings of the IEEE International Conference on Communications (ICC)*; 2019 May 20–24; Shanghai, China; 2019.
- [23] Luo S, Chen X, Zhou Z, Chen X, Wu W. Incentive-aware micro computing cluster formation for cooperative fog computing. *IEEE Trans Wirel Commun* 2020; 19(4): 2643–57.
- [24] Zhang T, Fang X, Liu Y, Li GY, Xu W. D2D-enabled mobile user edge caching: a multi-winner auction approach. *IEEE Trans Veh Technol* 2019; 68(12): 12314–28.
- [25] Zarandi S, Tabassum H. Federated double deep Q-learning for joint delay and energy minimization in IoT networks. In: *Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops)*; 2021 Jun 14–23; Montreal, QC, Canada; 2021.
- [26] Wang X, Han Y, Wang C, Zhao Q, Chen X, Chen M. In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning. *EEE Netw* 2019; 33(5): 156–65.
- [27] Ren J, Wang H, Hou T, Zheng S, Tang C. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things. *IEEE Access* 2019; 7: 69194–201.
- [28] Cui L, Su X, Ming Z, Chen Z, Yang S, Zhou Y, et al. CREAT: blockchain-assisted compression algorithm of federated learning for content caching in edge computing. *IEEE Internet Things J* 2022; 9(16): 14151–61.
- [29] Yu S, Chen X, Zhou Z, Gong X, Wu D. When deep reinforcement learning meets federated learning: intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network. *IEEE Internet Things J* 2021; 8(4): 2238–51.
- [30] Chen S, Yang L, Zhao C, Varadarajan V, Wang K. Double-blockchain assisted secure and anonymous data aggregation for fog-enabled smart grid. *Engineering* 2022; 8: 159–69.
- [31] Hefeeda M, Saleh O. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Trans Netw* 2008; 16(6): 1447–60.
- [32] McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*; 2017 Apr 20–22; Fort Lauderdale, FL, USA; 2017. p. 1273–82.
- [33] Wang X, Wang C, Li X, Leung VCM, Taleb T. Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching. *IEEE Internet Things J* 2020; 7(10): 9441–55.
- [34] Guo YH, Zhao ZC, He K, Lai SW, Xia JJ, Fan LS. Efficient and flexible management for industrial Internet of Things: a federated learning approach. *Comput Netw* 2021; 192(4): 108122.