计算机科学
COMPUTER SCIENCE

基于图强化学习的多边缘协同负载均衡方法

郑龙海，肖博怀，姚泽玮，陈星，莫毓昌

引用本文

郑龙海，肖博怀，姚泽玮，陈星，莫毓昌．基于图强化学习的多边缘协同负载均衡方法[J]．计算机科学，2025，52(3)：338-348.

ZHENG Longhai, XIAO Bohuai, YAO Zewei, CHEN Xing, MO Yuchang. Graph Reinforcement Learning Based Multi-edge Cooperative Load Balancing Method [J]. Computer Science, 2025, 52(3): 338-348.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100091

基于图强化学习的多边缘协同负载均衡方法

郑龙海[1,2,3]　肖博怀[1,2,3]　姚泽玮[1,2,3]　陈　星[1,2,3]　莫毓昌[4]
1 福州大学计算机与大数据学院　福州 350116
2 大数据智能教育部工程研究中心　福州 350002
3 福建省网络计算与智能信息处理重点实验室　福州 350116
4 华侨大学计算科学福建省高校重点实验室　福建 泉州 362021
(longhai20221108@163.com)

摘　要　在移动边缘计算中,设备通过将计算密集型任务卸载到附近边缘服务器,可以有效减少应用程序的延迟和能耗。为了提高服务质量,边缘服务器之间需要协作而非单独工作。针对多边缘协作的负载均衡问题,现有的策略往往依赖于精确的数学模型或缺乏对边缘拓扑关系的利用。为了解决此问题,文中提出了一种基于图强化学习的卸载决策方法。首先将多边缘协作的负载均衡场景抽象为图数据;然后采用基于图卷积神经网络的图嵌入过程来提取图的信息特征,以辅助深度Q网络进行卸载决策;最后通过集中反馈控制机制找到目标负载均衡方案。在多个场景下进行仿真实验,实验结果验证了所提方法在缩短任务平均响应时延方面的有效性,并且可以在短时间内获得优于对比算法且接近理想方案的负载均衡效果。

关键词:多边缘协作;负载均衡;任务卸载;图神经网络;深度强化学习
中图分类号　TP338

Graph Reinforcement Learning Based Multi-edge Cooperative Load Balancing Method

ZHENG Longhai[1,2,3],XIAO Bohuai[1,2,3],YAO Zewei[1,2,3],CHEN Xing[1,2,3] and MO Yuchang[4]
1 College of Computer and Data Science,Fuzhou University,Fuzhou 350116,China
2 Engineering Research Center of Big Data Intelligence,Ministry of Education,Fuzhou 350002,China
3 Fujian Key Laboratory of Network Computing and Intelligent Information Processing,Fuzhou University,Fuzhou 350116,China
4 Fujian Province University Key Laboratory of Computational Science,Huaqiao University,Quanzhou,Fujian 362021,China

Abstract　In mobile edge computing,devices can effectively relieve latency and energy consumption by offloading computation-intensive tasks to nearby edge servers. In order to improve the quality of service,edge servers need to collaborate with each other rather than working alone. For the load balancing problem of multi-edge collaboration,the existing solutions often depend on accurate mathematical models or make fair use of edge topological relationships. To solve this problem,an offloading decision-making method based on graph reinforcement learning is proposed in this paper. Firstly,the load balancing scenario with multi-edge collaboration is abstracted as graph data,then a graph embedding process based on graph convolutional neural network is used to extract the information features of the graph,for assisting the deep Q-network to make offloading decisions,and finally the objective load balancing plan is found through a centralized feedback-control mechanism. Simulation experiments are conducted in multiple scenarios,the results verify the effectiveness of the proposed method in shortening the average response latency of the tasks,and the load balancing effect which is better than the comparison algorithms and close to the ideal plan can be obtained in a short period of time.

Keywords　Multi-edge collaboration,Load balancing,Task offloading,Graph neural network,Deep reinforcement learning

到稿日期:2024-01-08　返修日期:2024-05-31
基金项目:国家自然科学基金(62072108);福建省科技经济融合服务平台(2023XRH001);福厦泉国家自主创新示范区协同创新平台项目(2022FX5)
This work was supported by the National Natural Science Foundation of China(62072108),Fujian Province Technology and Economy Integration Service Platform(2023XRH001) and Fuzhou-Xiamen-Quanzhou National Independent Innovation Demonstration Zone Collaborative Innovation Platform(2022FX5).
通信作者:陈星(chenxing@fzu.edu.cn)

1 引言

随着物联网和通信技术的不断发展，市场上的移动设备数量激增，人们通过移动设备可轻松地访问各种应用程序。由于应用程序的服务提供往往需要大量的资源来维持，移动设备普遍为便携式设计，其在计算能力和存储容量等方面受到限制，因此单纯依靠移动设备难以满足用户对应用低响应时延的要求。通过在资源丰富的云上处理应用程序中的密集型任务，可以有效解决移动设备计算资源受限的问题，减轻移动设备工作时的负载率，延长电池的使用时间。然而，任务从移动设备卸载到远程云的过程中，往往会产生较高的通信延迟，特别是在频繁交互的情况下，将严重影响服务质量（Quality of Service，QoS）。移动边缘计算（Mobile Edge Computing，MEC）作为一种有前景的计算范式被提出，用于缓解上述问题。在 MEC 框架中，移动设备通过无线接入点（Access Point，AP）与边缘服务器进行数据交互。通过将计算密集型和时延敏感型的应用任务卸载到附近的边缘服务器，可以有效避免时延过长的情况。通过低延迟去访问边缘服务器中的资源，可以极大拓展移动设备的计算能力，从而为用户提供更优质的体验。因此，边缘计算被认为是在无线城域网中提供更好服务的必要补充。

然而，移动设备在没有指导的情况下，通常将应用任务直接卸载到最近的边缘服务器上，这显然不是最优方案。当众多移动设备都处于同一个边缘服务器的服务覆盖范围内，短时间内同时访问边缘服务器上的资源时，将导致该边缘服务器的负载率过大，任务响应时延大幅提高，甚至可能会导致资源的阻塞和不可用。由于用户设备具有很强的位置变更性，其对边缘资源的访问难以预测，导致难以调控某一时刻边缘场景内移动应用的任务卸载情况。为了解决这一问题并提高系统的整体服务质量，多边缘协作被认为是一种可行的解决方案。当某个边缘服务器的任务负载量过大时，可以将一部分任务卸载到相邻的边缘服务器上处理，通过边缘设备之间的任务调度和协同合作，可以提高整体计算效率和性能。然而，在复杂的移动边缘环境下，很难正确地在边缘服务器之间调度任务来实现边缘与边缘之间的负载均衡。

由于边缘环境具有动态变化性，边缘服务器之间的负载均衡问题通常可以建模为混合整数非线性规划问题，该问题是一个 NP 难（NP-Hard）的优化问题。近年来，许多研究提出了不同方法来解决负载均衡问题。启发式算法是最常见的解决方法，如人工神经网络、差分进化算法、粒子群优化算法与遗传算法相结合的算法（Particle Swarm Optimization-Genetic Algorithm，PSO-GA）[1]等。但是上述在线启发式算法通常需要大量的参数且针对全新的边缘场景需要花费大量时间来重新求解目标负载均衡方案，导致泛化能力差，并且不符合应用任务对完成时延敏感的需求。强化学习作为一种算法模型被广泛用于求解任务卸载问题[2]，算法将 MEC 中的卸载决策抽象成马尔可夫决策过程（Markov Decision Process，MDP）[3]，基于预设的奖励函数，通过智能体和环境交互，无监督自主学习出最优的任务调度策略。训练完全的模型在运行时决策中可以迅速求解出目标负载均衡方案，降低整体任务的平均响应时延，符合边缘场景中移动设备应用对低时延服务的需求。但是，场景内边缘服务器等实体数量的增加将导致强化学习状态空间和动作空间过大，并产生较高的计算复杂度和存储空间复杂度。深度强化学习（Deep Reinforcement Learning，DRL）将强化学习（Reinforcement Learning，RL）和深度神经网络（Deep Neural Network，DNN）相结合，有效地解决了传统强化学习存在的局限性。利用深度强化学习强大的建模能力可以对边缘场景内的实体和实体关系进行表征，学习输入状态的特征，并通过函数逼近的方式指导智能体做出正确的动作。然而，深度神经网络只适合处理欧氏空间数据，不能很好地处理图数据，难以直接表征 MEC 中实体之间的拓扑关系[4]。将深度神经网络直接作为函数近似器的深度强化学习对复杂边缘场景的学习和适应能力较差。鉴于图神经网络（Graph Neural Network，GNN）在处理图结构数据时的优异表现，本文针对多边缘的负载均衡问题提出了一种将 GNN 和 DRL 相结合的方法，以降低系统的最大任务平均响应时延，提高整体服务质量。

综上所述，本文考虑了包括多个边缘的 MEC 场景，每个边缘都包含一台具有计算资源的边缘服务器。任意时刻，边缘服务器上都有来自附近的移动设备所卸载的任务以待计算。移动设备对边缘服务器资源的使用具有随意性，导致部分边缘服务器负载率过高，影响整体服务质量，因此，本文提出了多边缘协作负载均衡环境中基于图强化学习的卸载决策方法，引入 GNN 以辅助 DRL 对边缘图拓扑信息进行特征学习，快速求解出场景中的目标边缘负载均衡方案，最小化系统的最大任务平均响应时延。与现有的解决方法相比，本文的主要贡献如下：

1）考虑了 MEC 中多边缘协同的负载均衡问题场景。每个边缘都包含了一个具有计算能力的边缘服务器，用于接收来自用户设备卸载的应用任务。所有应用任务为无依赖型计算任务，并且可以按照所需的计算资源划分成相应的规模。边缘服务器可以通过无线连接将任务卸载到相邻的边缘服务器上处理，以降低自身的负载率。优化的目的是找到目标多边缘负载均衡方案，以最小化边缘服务器集合中的最大任务平均响应时延。

2）提出了一种多边缘协作负载均衡环境中基于图强化学习的卸载决策方法（Offloading Decision-Making Method Based on Graph Reinforcement Learning for Cooperative Load Balancing of Multi-Edge，DG-CBE）。首先，将 MEC 场景中的多边缘负载均衡问题建模为图数据，其中边缘服务器用图节点表示，边缘服务器之间的无线连接用边表示。其次，建立多边缘协作的负载均衡框架。有别于传统深度强化学习，引入了图卷积神经网络（Graph Convolution Network，GCN）来提取图数据的信息特征，以提升算法模型的推理能力和学习能力。最后，通过不断与环境交互学习，DG-CBE 可以得到快速适应边缘场景的任务调度模型。

3）针对所提算法设计了充分的实验，并与理想方案、基于规则的方法以及基于强化学习的不同方法进行对比和分析，

验证了 DG-CBE 方法的有效性。实验结果表明，DG-CBE 方法在多边缘负载均衡场景中具有出色的实用性，能够在短时间内得到效果接近理想方案的负载均衡策略，且性能优于其他 4 种对比算法。

本文第 2 章分析了相关工作；第 3 章对多边缘协作的负载均衡问题进行了系统建模和问题定义；第 4 章详细描述了本文提出的 DG-CBE 方法；第 5 章设计了仿真实验，验证了本文所提出方法的有效性；最后总结全文并展望未来。

2 相关工作

多边缘协作的负载均衡通过边缘服务器之间合理地调度任务，来实现短时间内处理来自用户设备卸载的时延敏感型和计算密集型任务。作为在无线城域网、工业互联网等领域有广阔应用前景的技术，负载均衡这一课题受到了各界广泛的关注，并且许多学者都取得了重要研究成果。

Cabrera 等[5]设计了一个通用启发式引擎来实现多目标动态负载均衡，该引擎可以在迭代问题中执行多种负载均衡策略。Mehrabi 等[6]设计了一种基于贪心的调度算法，用于寻找客户端和边缘服务器之间的最优映射，以优化负载均衡。Wang 等[7]设计了一种基于二进制的灰狼遗传算法，用于优化边缘服务器的部署成本及其负载均衡。Yang 等[8]利用差分进化算法来解决无人机部署问题，以实现无人机在覆盖约束下的负载均衡。Jia 等[9]比较了启发式算法和分布式遗传算法在平衡云之间的工作负载的性能，旨在优化系统中移动应用程序的响应时间延迟。Tran 等[10]将 MEC 服务器卸载问题定义为混合非线性规划问题，并提出了一种适用于多 MEC 服务器网络的启发式搜索方法，通过不断迭代调整卸载决策来优化卸载方案。Xu 等[11]研究了基于非正交多址的车辆边缘计算中任务卸载和资源的联合优化问题，并提出了一种博弈论方法。以上工作都是基于启发式的方法来解决多边缘负载均衡和任务卸载问题，然而，大部分启发式优化方法需要精确的数学建模和特定的专家知识，并且往往需要花费大量时间来求解目标卸载方案，难以应用于拓扑复杂的动态 MEC 场景。

Liu 等[12]提出了一种基于 DRL 的均衡感知聚类解决方案，其满足物联网边缘系统中通信与计算平衡的需求。Li 等[13]将负载均衡表述为非线性规划问题，并设计了一个基于卷积神经网络的框架来优化车联网中端到端的负载均衡。Xu 等[14]提出了一种基于 DRL 的移动负载平衡算法，采用两层架构来解决超密集网络中的大规模负载平衡问题。Chen 等[15]在工业物联网环境中设计了一种基于 RL 的协同负载均衡方法，每个边缘都可以根据局部信息独立调度任务，并在相邻边缘之间实现负载平衡。Li 等[16]提出了一种基于 RL 的优化框架，用于解决无线 MEC 的资源分配问题，并提出了基于 Q-learning 和 DRL 的两种解决方案。Zhang 等[17]提出了一种 RL 方法，通过部分卸载任务来优化无线供电 MEC 网络的整体计算率。利用深度强化学习强大的建模能力，可以对 MEC 环境进行表征，并通过与环境交互学习快速得到负载均衡策略。然而，传统基于 DNN 的强化学习难以直接应用于边缘服务器及由服务器之间的无线连接所形成的图网络结构，并且学习的过程中往往忽略了边缘服务器之间的拓扑信息。

Li 等[18]提出了一种基于深度图的 RL，用于无人机辅助 MEC 系统的联合巡航控制和任务卸载。Chen 等[19]为了解决多用户场景下边缘计算的依赖任务卸载，提出了一种基于 GNNs 的 MEC 系统有向无环图多任务卸载方法。Dai 等[20]利用基于图卷积的多智能体强化学习方法对无人机群进行控制，该方法能够捕获和利用无人机之间的相互作用，从而有效地提高信号覆盖率和公平性，同时降低总体能耗。Zhang 等[21]针对具有能量采集功能的 MEC 场景，设计了 GNN 增强的 DRL 来解决 MEC 中的联合优化问题，采用 GCN 进行特征提取，以学习不同移动设备与能量采集之间的关系，更好地控制部分卸载的比例。Sun 等[4]研究了 MEC 系统中的多设备协同卸载策略，将 MEC 环境下的计算卸载问题建模为一个图状态迁移问题，并提出了一种基于 GNN 的任务卸载机制，该机制可以直接对具有消息传递和聚合功能的图数据进行学习。Gao 等[22]提出了一种大规模 MEC 系统中基于多智能体图强化学习的快速自适应任务卸载和资源分配方法，将 MEC 构建为一个共享的 MD agents-ESs 图，并设计了一个基于 AGR 的消息网络，使每个 MD 能够聚合 ESs 和其他 MD 的信息，解决了 MD 智能体在 MEC 系统中的部分可观察性问题。上述工作将图神经网络引入 MEC 环境中的任务计算卸载研究，获得了具有竞争力的卸载性能。然而，以上研究大多集中于无人机联合控制和任务卸载等问题，对于基于图强化学习的无线城域网中多边缘协作卸载负载均衡问题的研究还较少。

受到上述研究的启发，本文考虑了 MEC 中多边缘协同的负载均衡场景，提出了一种多边缘协作负载均衡环境中基于图强化学习的卸载决策方法，通过全局代理集中式反馈控制边缘服务器之间的任务调度，逐步找到多边缘协作的目标负载均衡方案，降低整个系统的最大任务平均响应时延。

3 系统模型与问题定义

本章将阐述 MEC 中的多边缘协同负载均衡环境，并结合无线城域网的具体问题场景展开建模。如图 1 所示，一组边缘和无线接入点均部署在无线城域网中固定的位置，每一个边缘都包括一个边缘服务器，并通过无线连接与相邻的边缘服务器进行交互。假设用户设备卸载到边缘服务器上的应用任务为无依赖任务，即任务可以在边缘服务器上动态划分和调度[9]。例如，某个边缘服务器上 50%的任务量可以在当前服务器上执行，其余 50%在几个相邻的边缘服务器上执行。用户设备的移动性以及任务卸载到边缘服务器的不确定性，必然会导致无线城域网中的边缘服务器负载率波动大。当边缘服务器处于高负载率时，会导致其执行任务时延过大，低负载率则会造成资源的严重浪费。因此，亟需一种快速高效的任务卸载策略，通过控制无线城域网中边缘服务器之间有效地协作，来实现边缘的负载均衡，减少整体任务的平均响应时延，提高系统的资源利用率。本文中使用的主要符号如表 1 所列。

表 1 问题定义中的主要符号

Table 1 Primary symbols in problem definition

符号	定义
N	边缘服务器个数
M	无线连接条数
$es_i \in ES$	边缘服务器节点
$ec_i \in EC$	无线连接
K	边缘服务器的无线连接条数
f^{es_i}	边缘服务器es_i的服务速率
$d_{i,j}$	边缘服务器es_i和es_j之间单位任务的传输时延
λ_i	边缘服务器es_i的任务到达率
F_{mat}	全局负载均衡方案
$f_{i,j}$	单位时间内从边缘服务器es_i调度到es_j的任务量
ω_i	边缘服务器es_i的任务负载量
T_i^r	边缘服务器es_i的任务平均响应时延
$t_{i,j}$	$f_{i,j}$的平均任务处理时延
$T_a(l_j)$	单位任务在边缘服务器es_i负载率为l_j时的计算时延
$T_{\max}$	系统最大平均任务响应时延

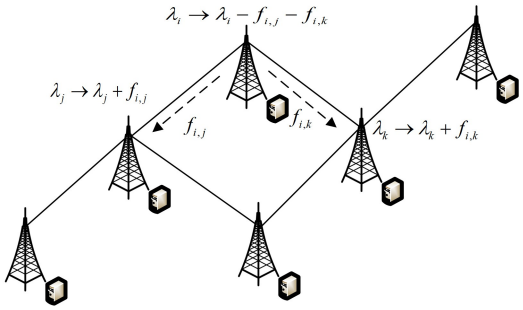


图 1 无线城域网中多边缘协作的负载均衡场景

Fig. 1 Load balancing scenario of multi-edge cooperation in WMAN

在无线城域网中，N 个边缘服务器被定义为 $ES=\{es_1, es_2, \cdots, es_N\}$，其中$es_i$表示第 i 个边缘服务器。多边缘负载均衡环境中有 M 条无线连接，定义为 $EC=\{ec_1, ec_2, \cdots, ec_M\}$，其中$ec_i$表示第 i 条无线连接。每个边缘服务器通过无线连接与 k 个相邻的边缘服务器相连。

首先，定义$F^{es}=\{f^{es_1}, f^{es_2}, \cdots, f^{es_N}\}$来表示边缘服务器的服务速率，其中$f^{es_i}$表示边缘服务器$es_i$在单位时间内可以处理的任务量。其次，定义单位任务在边缘服务器之间通过无线连接的传输时延，如式(1)所示：

$$D=\begin{bmatrix} d_{1,1} & \cdots & d_{1,N} \\ \vdots & \ddots & \vdots \\ d_{N,1} & \cdots & d_{N,N} \end{bmatrix} \tag{1}$$

其中，$d_{i,j}$表示单位任务在边缘服务器es_i和es_j之间的传输时延。当 $i=j$ 时，$d_{i,j}=0$；当边缘服务器es_i和es_j之间没有无线连接时，$d_{i,j}=+\infty$。

然后，定义边缘服务器的任务到达率 $\lambda=\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$，其中$\lambda_i$表示单位时间内从移动设备卸载到边缘服务器$es_i$上的任务量。为了更好地描述，将卸载到边缘服务器上的任务称为到达任务。通过将边缘服务器上的到达任务正确地调度到相邻的边缘服务器上，可以实现边缘服务器上的负载均衡。为了简化表述，将边缘服务器上的负载均衡称为边缘负载均衡。

定义系统的多边缘负载均衡方案如下：

$$F_{\text{mat}}=\begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix}=\begin{bmatrix} f_{1,1} & \cdots & f_{1,N} \\ \vdots & \ddots & \vdots \\ f_{N,1} & \cdots & f_{N,N} \end{bmatrix} \tag{2}$$

其中，F_i表示边缘服务器es_i的负载均衡方案，用于描述es_i对到达任务λ_i的任务调度策略。每一个边缘服务器只能对自身的到达任务进行调度，或接收来自相邻边缘服务器卸载过来的任务。当 $i=j$ 时，$f_{i,j}$表示单位时间内在本地执行的任务量；当 $i\neq j$ 时，$f_{i,j}$表示单位时间内从边缘服务器es_i卸载到es_j的任务量；当边缘服务器es_i和es_j之间没有无线连接时，$f_{i,j}=0$。

具体来说，边缘服务器的实际任务负载量定义为 $W=\{\omega_1, \omega_2, \cdots, \omega_N\}$。其中$\omega_i$表示$es_i$上需要执行的任务量，其由两部分构成，一部分是卸载到相邻边缘服务器后自身剩下的到达任务量，另一部分为接收从相邻边缘服务器卸载过来的任务量总和。对于没有经过任务调度过程的初始状态，$\omega_i=\lambda_i$。在多边缘协同负载均衡的过程中，针对可按照一定比例动态拆分的无依赖型到达任务λ_i，可以在当前边缘服务器es_i或es_i的相邻边缘服务器上执行。因此，定义边缘服务器es_i上到达任务的平均响应时延为：

$$T_i^r=\frac{1}{\lambda_i}\cdot\sum_{j=1}^{N}(f_{i,j}\cdot t_{i,j}) \tag{3}$$

其中，$t_{i,j}$表示单位任务从边缘服务器es_i调度到es_j上处理所需要的时间，其由任务传输时延和任务计算时延两部分组成。因此，定义$t_{i,j}$为：

$$t_{i,j}=T_a(l_j)+d_{i,j} \tag{4}$$

其中，$T_a(l_j)$表示边缘服务器es_i在负载率为$l_j=\omega_i/f^{es_i}$的情况下，计算单位任务的所需时间。为了进一步评估多边缘场景的服务质量，定义系统中边缘服务器上到达任务的最大平均响应时延为：

$$T_{\max}=\max\{T_1^r, T_2^r, \cdots, T_N^r\} \tag{5}$$

针对一个边缘场景，采用不同的负载均衡方案会产生不同的$T_{\max}$值。$T_{\max}$越小，代表对应的负载均衡方案性能越好。因此，为了实现 MEC 中良好的多边缘协作负载均衡，定义目标函数为：

$$\text{Minimize } T_{\max} \tag{6}$$

4 方法描述

边缘服务器通过相互之间合理的任务卸载，使系统中所有边缘服务器处于合适的负载率，以适应动态变化的边缘环境。然而边缘服务器及由服务器之间的无线连接所形成的图拓扑网络本质上是非欧几里德空间，不适合用传统基于 DNN 的方法直接进行表征。DNN 难以利用图节点之间的关系特征，将导致关键信息丢失。因此，提出了一种将图卷积网络和深度强化学习相结合的多边缘负载均衡方法 DG-CBE，具体框架如图 2 所示。首先，将多边缘场景抽象为图数据，在此基础上通过建立全局代理将 GCN 和 DQN 相结合，对边缘服务器上的任务进行集中式调度；其次，全局代理根据当前系统状态，通过基于 GCN 的图节点嵌入过程决策出在每条无线连接上执行卸载动作的概率$\hat{Q}$值，根据概率$\hat{Q}$值选择出在哪条无线连接上执行卸载动作；然后，根据较高到达任务平均时延的边缘服务器向较低一方卸载任务的贪心策略决定卸载方向，形成最终卸载动作；最后，全局代理通过执行卸载动作与动态环境进行交互，以获得反馈

奖励，并基于此进行策略更新，最终通过集中式反馈控制，逐步找到多边缘协作的目标负载均衡方案。

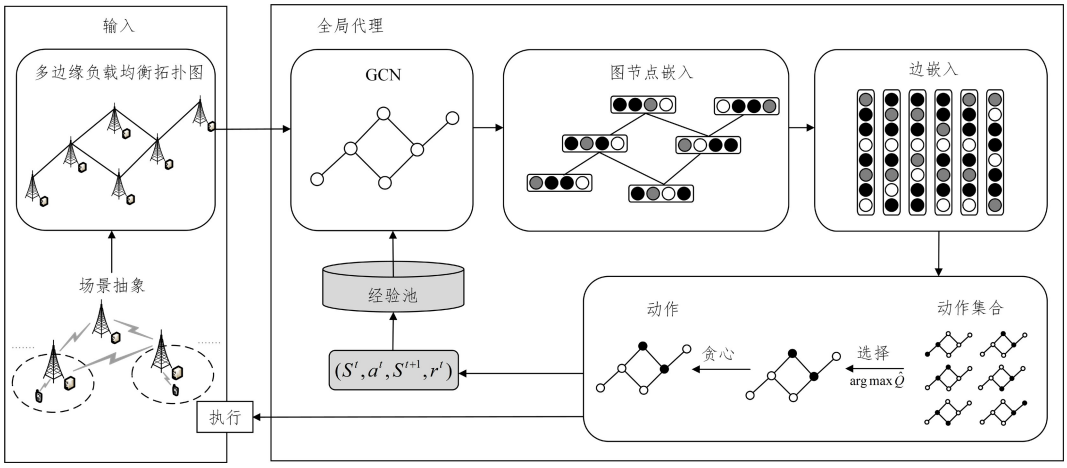


图 2 多边缘负载均衡环境中基于图强化学习的决策方法框架

Fig. 2 Framework of DG-CBE

下文将对所提方法的框架进行详细描述。首先将第 3 章中构建的问题定义与马尔可夫决策过程相结合，对全局边缘服务器的负载均衡过程进行构建，即通过构建状态空间、动作空间、状态转移函数和奖励函数四元组来求解目标负载均衡方案；然后介绍深度强化学习过程中结合图卷积网络进行图特征信息学习的过程；最后介绍 DG-CBE 方法的训练过程。

4.1 马尔可夫决策过程

马尔可夫决策过程（Markov Decision Process，MDP）是一种通过与环境进行交互式学习来优化目标函数的理论框架，是强化学习在数学上的理想化形式[23]。在使用图强化学习来解决多边缘负载均衡问题的过程中，基于图卷积神经网络处理后的全局边缘服务器信息，决策出边缘服务器之间最合适的卸载动作。对于高负载率、高任务响应时延的边缘服务器，通过将部分到达任务卸载到相邻的边缘服务器，可以达到多边缘负载均衡状态。MDP 可以对任务卸载过程进行建模，以最大化指定的长期奖励。通常情况下，MDP 被定义为四元组$\langle S, A, R, P\rangle$，分别代表状态空间、动作空间、状态转移函数和奖励函数。

4.1.1 状态空间

状态空间是全局代理进行决策的依据，用于表示场景中所有边缘服务器在当前环境下的系统状态。本文将系统在第 t 个时间步的状态空间定义为$S^t=\langle F^{es}, W^t, T^{r,t}\rangle$，其中$F^{es}$表示系统内所有边缘服务器的服务速率。假设在系统进行负载均衡调度的过程中，所有边缘服务器的服务速率保持不变。W^t表示第 t 个时间步时所有边缘服务器的实际任务负载量，是评估当前时刻系统任务负载情况的关键。初始系统中所有边缘服务器的实际任务负载量即为到达任务。$T^{r,t}=\{T_1^{r,t}, T_2^{r,t}, \cdots, T_N^{r,t}\}$表示第 t 个时间步时系统根据当前负载均衡方案计算得到的所有边缘服务器的到达任务平均响应时延。

4.1.2 动作空间

动作空间是全局代理对系统中可优化变量的调整，通过对多边缘场景下全局负载均衡方案F_{mat}的不断调整，来最小化系统最大任务平均响应时延。记一次调整的粒度大小为 δ，即一次动作是以 δ 大小来调整两个边缘服务器之间的任务卸载量。定义全局任务负载均衡过程中的动作空间 $A=\{tran_1^{ec_1}, tran_2^{ec_1}, \cdots, tran_1^{ec_M}, tran_2^{ec_M}\}$，其中$a^t\in A$表示系统在第 t 个时间步执行的一次任务卸载动作。具体来说，$tran_1^{ec_1}$表示对无线连接ec_1所连接的两个边缘服务器中的任务卸载情况进行更改，下标 1 和 2 代表边缘服务器之间的任务卸载方向。

4.1.3 状态转移函数

状态转移函数定义第 t 个时间步，状态S^t执行动作a^t后进入下一个状态S^{t+1}的过程表示。通过动作a^t可以确定需要执行任务卸载动作的边缘服务器和卸载对象，并按照一次 δ 大小的任务量进行卸载。例如，ec_k表示边缘服务器es_i和es_j之间的无线连接，边缘服务器es_i和es_j在第 t 个时间步的任务负载量分别为w_i^t和w_j^t。通过执行动作$tran_1^{ec_k}$，边缘服务器es_i将 δ 大小的任务量卸载到es_j上。下一个状态S^{t+1}时，边缘服务器es_i和es_j中的任务负载量变化为$w_i^t-\delta$和$w_j^t+\delta$。执行动作会使状态空间中的边缘服务器实时任务负载量和到达任务平均响应时延发生相应的变化。为了使边缘服务器的负载率在负载均衡的过程中始终维持在合理的范围内，定义每个边缘服务器的负载率阈值 φ。如果执行动作a^t后会导致边缘服务器负载率超过阈值 φ，则不执行该动作，并将其判为非法动作，给予一定的处罚。

4.1.4 奖励函数

奖励函数用于引导全局代理学习如何在给定的状态下获得更大的长期收益。在针对本文描述的问题场景中，采用了集中式的算法框架，并以最小化系统的最大到达任务平均响应时延作为全局优化目标。对于合法动作a^t，奖励函数r^t被定义为系统最大任务平均响应时延相关的负数，可以直接体现优化目标在执行单次任务动作之后的改变，使得全局代理更容易学习经验以获得更大的长期优化目标收益；对于非法动作，奖励函数r^t定义为固定处罚值，具体如下：

$$r^t=\begin{cases}T_{\max}^t-T_{\max}^{t+1}, & \text{valid action}\\ -1, & \text{invalid action}\end{cases} \tag{7}$$

4.2 基于 GCN 的图节点嵌入过程

在真实的动态多边缘负载均衡场景中，状态空间和动作空间是巨大的，传统强化学习难以准确地表示。因此，引入深度神经网络来增强算法的表征能力，并通过神经网络参数的更新来不断逼近目标卸载方案[4]。本文第 3 章构建的多边缘负载均衡场景适合用图数据表征，其中图上的节点表示边缘服务器实体，边表示边缘服务器之间的无线连接。

有别于传统基于深度神经网络的 DQN，本文使用了基于 GCN 的图节点嵌入来提取全局信息，以更好地利用边缘场景的拓扑结构和边缘服务器相邻节点信息进行卸载决策。图节点嵌入是一种在连续向量空间中表示节点的方法，将每个节点映射到一个连续的向量空间中，以捕获节点之间的结构和特征信息。通过图节点嵌入过程获得的节点嵌入可以被看作一个特征向量，其包含了图中节点及其周围环境的信息，用于更好地指导深度强化学习进行策略选择和更新。基于图卷积网络的图节点嵌入过程，最终获得卸载动作的概率$\hat{Q}$值，如图 3 所示。

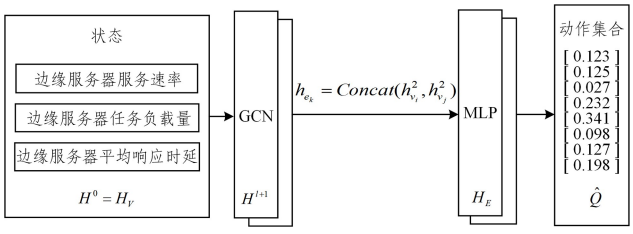


图 3 图节点嵌入过程

Fig. 3 Graph node embedding procedure

第 t 个时间步的 MEC 系统状态被抽象为无向拓扑图 $G=(V,E)$，其中节点 V 表示边缘服务器集合，边 E 表示边缘服务器节点之间的无线连接集合。图节点 $v_i(v_i\in V)$ 作为边缘服务器实体 es_i 的映射，图边 $e_i(e_i\in E)$ 作为无线连接 ec_i 的映射。其中，图节点 $v_i(v_i\in V)$ 的特征信息用 $h_{v_i}^0$ 表示，具体定义如下：

$$h_{v_i}^0=\langle f^{es_i},w_i^t,T_i^{r,t}\rangle \tag{8}$$

对于每一个图节点，其特征信息由第 t 个时间步所映射的边缘服务器上的服务速率、任务负载量和到达任务平均响应时延表示。图中的边只参与图拓扑结构的构成，用于表示图节点之间的连接关系。通过解析 t 时间步的系统状态 S^t，可以获取每一个图节点的特征信息，最终获得全局图节点嵌入信息 $H_V=\{h_{v_1}^0,h_{v_2}^0,\cdots,h_{v_N}^0\}$，作为图卷积网络的输入数据 H^0。其表达式为：

$$H^0=H_V \tag{9}$$

GCN 通过聚合邻域内其他节点的特征来学习图节点在下一状态的嵌入。通过对图节点的二阶邻域信息进行聚合，每一个边缘服务器可以充分了解周围无连接结构和相邻边缘服务器的参数信息和负载状态。图节点的信息聚合表示为：

$$H^{l+1}=\sigma(C^l(H^l,H_P^l)) \tag{10}$$

其中，σ 表示激活函数，C^l 表示拼接操作。将每一个图节点本身的嵌入信息和加权聚合后的邻域信息进行拼接后，得到下一层网络的图节点嵌入。l 代表 GCN 的层数，因为需要使用二阶邻域信息进行决策，所以 $l\in\{0,1\}$。H_P^l 表示输入数据 H^0 通过 GCN 聚合后的结果，其定义为：

$$H_P^l=(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}})\cdot H^l\cdot W^l \tag{11}$$

其中，$\hat{\boldsymbol{D}}$ 和 $\hat{\boldsymbol{A}}$ 分别表示无向拓扑图 G 的度矩阵和邻接矩阵，W^l 表示可学习的网络参数。

当 $l=1$ 时，获得经过两层 GCN 聚合后的全局图节点嵌入信息，表示为 $H^2=\{h_{v_1}^2,h_{v_2}^2,\cdots,h_{v_N}^2\}$。对于连接图节点 v_i 和 v_j 的边 e_k，其嵌入信息表示为 h_{e_k}，由所连接的两个图节点的嵌入信息联合表示，其定义如下：

$$h_{e_k}=Concat(h_{v_i}^2,h_{v_j}^2) \tag{12}$$

通过式(12)可以得到图中每一条边的嵌入信息，并用 $H_E=\{h_{e_1},h_{e_2},\cdots,h_{e_M}\}$ 表示所有边的嵌入信息。最后，考虑到特征表示之间存在的排列不变性[4]，根据边的嵌入信息，使用多层感知器(Multilayer Perceptron，MLP)对边进行分类预测。将所有边的嵌入信息 H_E 输入 MLP 中，通过式(13)可以获得在每条边上，即在每一条无线连接上执行任务卸载动作的概率$\hat{Q}$。

$$\hat{Q}=\sigma(MLP_2(ReLU(MLP_1(H_E)))) \tag{13}$$

选择在$\hat{Q}$值最大的边所映射的无线连接上执行任务卸载，并基于贪心的策略控制卸载方向，形成最终的动作 a^t。综上所述，基于图卷积网络的图节点嵌入过程如算法 1 所示。

算法 1 图节点嵌入过程

输入：G=(V,E)，H_V

输出：卸载动作的概率$\hat{Q}$值

1. $H^0=H_V$
2. for l=0,1 do
3. 根据式(11)，得到H^l聚合后的结果H_P^l
4. 根据式(10)，通过拼接得到下一个状态嵌入H^{l+1}
5. end for
6. for each $e_i\in E$ do
7. 根据式(12)，得到每条边的嵌入信息h_{e_i}
8. end for
9. $H_E=\{h_{e_1},h_{e_2},\cdots,h_M\}$
10. 根据式(13)，将H_E输入多层感知器，得到动作的概率Q

4.3 DG-CBE 的代理模型训练

DG-CBE 方法的全局代理训练过程如算法 2 所示。

算法 2 DG-CBE 的全局代理训练

输入：G=(V,E)，F^{es}，λ，γ，D，F_{mat}

输出：训练好的代理模型的$\hat{Q}$值网络参数 θ

1. 初始化：探索率 $\varepsilon=\varepsilon_{max}$，容量为 x 的经验池 B，迭代次数上限$E^{max}$，批量经验大小 m，$\hat{Q}$值网络参数 θ 和目标网络参数θ′等
2. for each episode e do
3. 初始化F_{mat}，λ，$W^t=\lambda$
4. for each step t do
5. 根据式(3)，得到边缘服务器的任务平均响应时延$T^{r,t}$
6. 观测状态$S^t=\langle F^{es},W^t,T^{r,t}\rangle$
7. 根据算法 1，得到动作集合 A 的概率$\hat{Q}$值
8. 利用 ε-greedy 策略选择动作a^t

9. 执行动作a^t，根据式(7)得到全局奖励r^t
10. 转移到下一个状态S^{t+1}
11. 将经验$\langle S^t, a^t, r^t, S^{t+1}\rangle$存入经验池
12. end for
13. 线性下降探索率 ε
14. 从经验池批量抽取经验，根据式(14)计算损失函数 $L(\theta^t)$
15. 反向传播更新$\hat{Q}$值的网络参数 θ
16. if e mod P=0 then
17. 更新目标网络参数$\theta'=\theta$
18. end if
19. end for
20. 返回$\hat{Q}$值的网络参数 θ

首先，初始化全局代理模型训练所需的参数，包括初始探索率 ε、$\hat{Q}$ 值的网络参数 θ、目标网络参数 θ'、容量大小为 X 的经验池 B。针对多边缘负载均衡拓扑场景，每一轮学习开始前，需要初始化系统中边缘服务器的任务到达量 W 和全局负载均衡方案F_{mat}，并重新计算当前所有边缘服务器的任务平均响应时延T^r。

代理通过与环境交互进行学习。在每一步过程中，以概率 ε 随机生成卸载动作，否则，基于当前的系统状态，根据式(8)－式(13)决策出在每条无线连接上执行任务卸载动作的概率 Q 值，并基于贪心生成卸载动作。探索率 ε 随训练迭代次数的增加线性下降。通过执行动作a^t并获得对应的奖励r^t，系统进入下一个状态S^{t+1}，将经验以四元组$\langle S^t, a^t, r^t, S^{t+1}\rangle$的形式存入经验池中。每一轮学习后，当经验回放池的容量足够大时，从经验回放池批量抽取大小为 m 的样本，参考文献[24]，使用均方误差作为损失函数，对 Q 值网络参数进行更新，具体计算式如下：

$$L(\theta^t)=(r+\gamma\cdot\max Q(S',a';\theta^{t\prime})-Q(S,a;\theta^t))^2 \quad (14)$$

其中，$\gamma\in[0.1]$是折扣率，θ^t和$\theta^{t\prime}$分别代表第 t 个时间步时 Q 值网络和目标网络的参数。每经历 P 轮学习，使用 Q 值网络的参数 θ 来替换目标网络参数θ'。经历$E^{\max}$次迭代后，全局代理中的算法模型趋于稳定收敛。

4.4 DG-CBE 的算法复杂度

记输入 DG-CBE 方法的边缘图数据 $G=(V,E)$的图节点数目和边数目分别为$|V|$和$|E|$，图节点的输入特征维度和输出特征维度分别表示为 F 和F'，状态空间和动作空间大小分别为$|s|$和$|a|$。计算 DG-CBE 的时间复杂度时，主要考虑以下两步。

1)Action selection：设 GCN 和 MLP 的隐藏层数为L^G和L^M。在图卷积聚合的过程中，第 $i(i\in[0,L^G))$层邻接矩阵乘以特征矩阵的时间复杂度为 $O(|V|*F_i*|E|)$，汇聚邻居节点的时间复杂度为 $O(|V|*F_i*F_i')$，总的图卷积聚合过程的时间复杂度为$C_t^G=O(\sum_{i=0}^{L^G-1}(O(|V|*F_i*|E|+O(|V|*F_i*F_i'))$。设第 $k(k\in[0,L^M))$层 MLP 的神经元个数为U_k^M，则第 k 层神经网络的复杂度为 $O(U_{k-1}^M*U_k^M+U_k^M*U_{k+1}^M)$，总的神经网络推理的时间复杂度为$C_t^M=O(4*F'_{L^G-1}*U_0^M+\sum_{k=1}^{L^M-2}(U_{k-1}^M*U_k^M+U_k^M*U_{k+1}^M)+U_{L^M-1}^M*|a|*1/2)$。系数 4 代表输入 MLP 的边嵌入信息由图节点特征自身拼接和相连图节点特征拼接而来，系数 1/2 代表网络输出为实际动作空间大小的一半。

2)Reward calculation：对于全局代理，计算奖励值的时间复杂度为$C_t^R=O(|s|)$。

通过合并以上两步，可以得到全局代理在时间步 t 时训练的时间复杂度为$C_t=C_t^G+C_t^M+C_t^R$。则 DG-CBE 的时间复杂度为 $O(T*C_t)$，其中 T 为训练所需的总时间步数。

5 实验与分析

鉴于物理环境下为进行实验所用的硬件成本过高，通过设计仿真实验，搭建了相应的负载均衡场景，并对 DG-CBE 方法的性能进行了详细的评估。为了保证实验的公平合理性，DG-CBE 方法和对比方法均在 Python 3.7 的环境下实现，并在搭载了 3.00 GHz Intel Core i5-9500 CPU 芯片和 16 GB RAM 的 Windows 10 环境中运行，所使用的图强化学习基于 Pytorch 1.7.0 环境。

5.1 实验设置

如图 4 所示，根据上海地区 AP 分布[1)]的地理信息[25-28]，从经纬度(31.1°N,121.1°E)到(31.4°N,121.5°E)的范围内随机选取一定数量的基站点，构建了 5 个多边缘协作的负载均衡场景。场景中边缘服务器的个数 N 为 15，并且每一个边缘服务器和相邻边缘服务器的无线连接条数 K 满足 $0<K\leqslant 3$。根据实验结果进行调整，单次任务卸载量 δ 设置为 $3\%\cdot f^{es_i}$，其中f^{es_i}为执行本次任务卸载动作的边缘服务器的服务速率。边缘服务器的任务到达率和服务速率均符合正态分布，表示为 $N(\mu,\sigma^2)$，其中 μ 和 σ 分别表示均值和误差。单位任务通过无线连接的传输时延取决于边缘服务器之间的距离，为了方便计算，本文将传输时延映射压缩到固定区间。仿真实验中的具体实验参数如表 2 所列。

表 2 仿真实验中的参数设置

Table 2 Parameter setting in simulation experiment

参数	取值
边缘服务器个数 N	$N=15$
边缘服务器的无线连接条数 K	$K\in\{1,2,3\}$
边缘服务器的服务速率f^{es}	$f^{es}\sim N(15,6)>\lambda+0.25$
无线连接的传输时延 d	$d\in(0.1,0.2)$
边缘服务器的任务到达率 λ	$\lambda\sim N(10,4)<f^{es}-0.25$
一次卸载的任务量 δ	$\delta=3\%\cdot f^{es}$
边缘服务器的负载率阈值 φ	$\varphi=90\%$

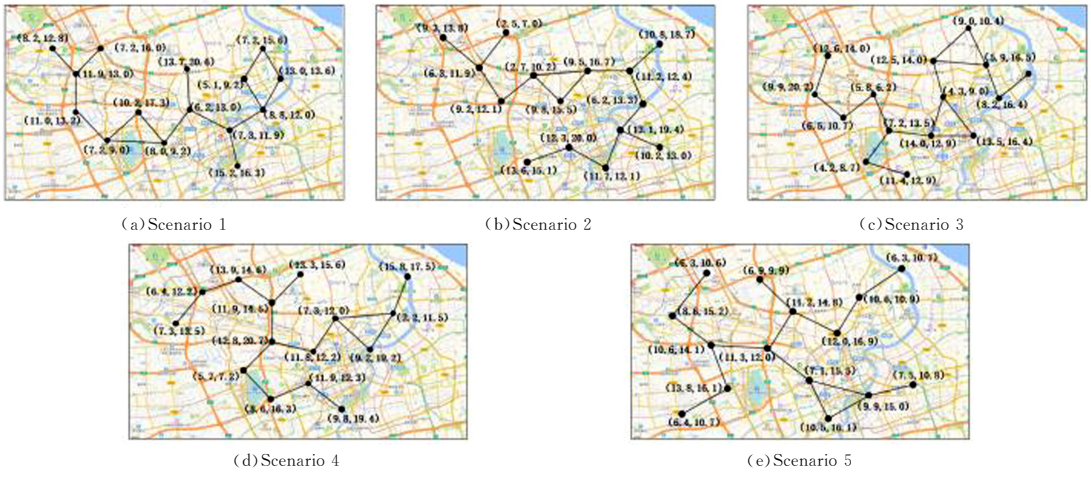
基于排队理论[29]，边缘服务器e_i上的平均任务计算时延如下：

$$T_a(l_i)=\frac{1}{v_i\cdot(1-l_i)} \quad (15)$$

根据式(15)可以看出，边缘服务器e_i上的负载率l_i必须满足$l_i<100\%$，否则无法处理自身任务。当边缘服务器的负载率过高时，将导致该服务器上的平均任务计算时延异常，加大了卸载策略的学习难度。因此，边缘服务器设置了负载率上

1) http://sguangwang.com/TelecomDataset.html

限阈值 φ，本文设置为 90%，以将边缘服务器的负载率控制在合理的范围内。



(a) Scenario 1　(b) Scenario 2　(c) Scenario 3

(d) Scenario 4　(e) Scenario 5

图 4　多边缘协作的负载均衡场景

Fig. 4　Load balancing scenario for multi-edge collaboration

5.2　实验结果与分析

在本节中，为了验证 DG-CBE 方法的有效性，在 5 个多边缘负载均衡场景中，将其与 5 种已有的方法在系统响应时延和方法执行时间上进行了充分对比。对比算法如下。

1) PSO-GA[2]。PSO-GA 是一种启发式算法，其在经典 PSO 算法的基础上，引入了 GA 算法中交叉和变异算子的思想，融合两种方法的优点来提高算法的鲁棒性和搜索性能。PSO-GA 通过粒子位置和速度的更新与变异计算操作，不断迭代寻找适应度函数值更大的解。本文实验中 PSO-GA 的参数设置如下：种群大小为 100，迭代次数为 10000，粒子变异概率为 0.05，惯性权重因子 ω 为 0.8，自我学习因子 c_1 和种群学习因子 c_2 均为 0.5。由于本文研究的多边缘协同负载均衡为 NP-Hard 问题，难以使用暴力枚举法直接搜索到最优解，因此 PSO-GA 可以被作为理想方案。

2) TDB-EC[30]。TDB-EC 是一种两阶段负载均衡决策方法。首先，利用全局信息进行负载均衡集中决策，设计基于 DNN 的预测模型，评估相邻边之间的任务调度范围；其次，利用局部信息进行分散的负载均衡决策，建立基于 DQN 的调整预测模型；最后，通过反馈控制得到目标负载均衡方案。在 TDB-EC 中，每个代理的训练次数、学习率和折扣率分别设置为 100，0.1 和 0.9。

3) RF-CLB[15]。RF-CLB 通过设计一个 Q 值预测模型，并结合 Q-learning 和支持回归向量(Support Vector Regression，SVR)来预测调整操作。采用分布式的决策模式，每个边缘服务器根据局部信息独立调度任务，在相邻边缘服务器之间实现负载均衡。通过反馈控制和多边缘协作，找到目标多边缘负载均衡方案。在 Q-learning 中，训练次数、学习率和折扣率也设置为 100，0.1 和 0.9。

4) C-DQN。C-DQN 为集中式的 DQN 算法，其训练一个基于传统神经网络的 Q 值预测模型，在运行时决策中该模型根据系统状态计算出在每一条边上执行任务调度动作的概率，并依据贪心策略执行卸载动作。通过设置 C-DQN 对比算法，可以验证图卷积神经网络提取图信息特征对负载均衡任务调度的意义和效果。

5) Rule-Based。Rule-based 是一种基于规则的方法，采用分散的负载均衡决策，其卸载判断依据为每个边缘服务器的负载率。每个边缘服务器根据自身负载率情况，参考文献[31]中制定的规则，即当边缘服务器 e_i 负载率 $l_i \geqslant 70\%$ 时，将任务卸载到负载率最低的相邻边缘服务器中；当边缘服务器 e_i 负载率 $30\% \leqslant l_i < 70\%$ 时，不作任何处理；当边缘服务器 e_i 负载率 $l_i < 30\%$ 时，减少向负载率最高的相邻边缘服务器卸载的任务量。

DG-CBE 方法中使用的神经网络模型由一层输入层、两层图卷积网络层、一层全连接隐藏层和一层输出层组成。输入层的输入维度为 3，隐藏层的神经元个数为 128。输出层的个数为边缘场景中无线连接的条数，代表在每一条无线连接上执行任务卸载动作的概率 Q 值。目标网络和 Q 值网络具有相同的网络结构，以较低的频率更新参数。算法的超参数设置如下：迭代轮数 $epoch=200$；每一轮迭代的步数 $step=200$；经验池大小 X 为 20000，批量学习的大小为 64；学习率 $\alpha=10^{-3}$；折扣率 $\gamma=0.9$；初始时探索率 $\varepsilon=\varepsilon_{\max}=1$，经过 50 轮迭代线性下降为 $\varepsilon=\varepsilon_{\min}=0.1$。

理想方案通过对不同场景中大量的负载均衡方案进行尝试和评估，不断搜索和逼近最优性能。如图 5 所示，将 DG-CBE 方法与理想方案在不同场景中进行对比，可以看到 DG-CBE 方法和理想方案在最大到达任务平均响应时延的优化差异保持在 6%以下，平均差异在 4.4%左右。

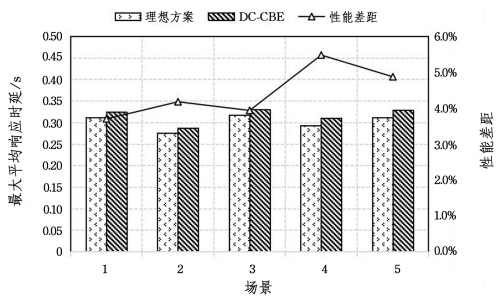


图 5　DG-CBE 与理想方案在不同场景下的最大平均响应时延对比

Fig. 5　Comparison of maximum average response time between DG-CBE and ideal plans in various scenarios

如表3所列，通过DG-CBE方法得到的各个边缘服务器负载率相比初始状态都有了更合理的调整，高负载的边缘服务器通过将自身到达任务卸载到临近边缘服务器来减轻自身负载率，低负载的边缘服务器的资源利用率也有相应的提升。相比耗费大量搜索时间得到的理想方案，DG-CBE方法可以根据当前的系统状态快速做出合适的任务卸载动作，在最终性能逼近理想方案的基础上，兼顾任务对完成时延敏感的需求。

表3 DG-CBE在不同场景下对边缘服务器负载率的调整效果

Table 3 Effect of DG-CBE on edge server load ratio adjustment in various scenarios

(%)

Scenario	Edge(e_i)	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
1	Initial state	64	92	45	83	80	59	87	48	67	61	93	73	55	46	96
	DG-CBE	64	71	67	73	59	73	60	72	67	70	78	70	60	66	78
2	Initial state	67	53	36	76	27	63	57	90	58	47	68	79	97	62	90
	DG-CBE	62	53	51	55	66	52	74	48	68	73	78	70	64	72	71
3	Initial state	90	49	61	94	53	48	89	94	82	48	89	87	36	52	57
	DG-CBE	63	75	64	42	69	58	76	78	81	60	75	57	74	76	38
4	Initial state	54	53	95	82	85	62	79	53	97	50	97	61	48	19	90
	DG-CBE	64	67	77	77	79	78	51	70	49	75	72	61	51	69	67
5	Initial state	60	86	75	57	59	94	46	65	66	69	76	70	71	97	59
	DG-CBE	63	78	70	78	64	67	74	69	48	67	77	68	77	67	71

如图6所示，在不同场景下，通过DG-CBE方法得到的最大到达任务平均响应时延优于其他对比方法。具体而言，DG-CBE方法在不同场景下的性能效果比TDB-EC和RF-CLB方法分别优化了0.6%～1.2%和2.96%～4.71%，比C-DQN方法优化了4.37%～6.78%。这是因为RF-CLB方法的分布式任务卸载决策只使用了边缘服务器局部信息，而C-DQN方法虽然使用了全局信息来决策，但是卸载动作的生成只考虑相连的两个边缘服务器的状态信息。TDB-EC方法虽然同时使用了全局和局部信息来决策任务卸载，但是基于DNN的网络结构对信息的学习水平相对有限。相比之下，DG-CBE方法使用了基于GCN的图节点嵌入过程来提取全局信息，可以很好地表征由边缘场景抽象的图信息，并利用场景中的拓扑结构和图节点领域信息进行卸载决策。此外，在不同的场景下，DG-CBE方法的效果比Rule-Based方法优化了12.67%～16.22%。这是因为基于规则的方法很难制定出适用于各种场景的规则，不同场景的负载均衡问题无法通过相同的规则来解决。

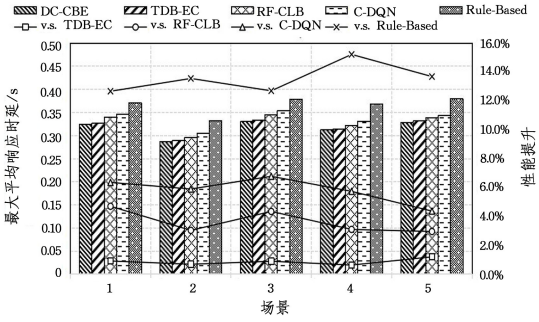


图6 DG-CBE和其他算法在最大到达任务平均响应时延上的对比

Fig. 6 Comparison between DG-CBE and other methods in terms of the maximum average response time of arrival tasks at all edge servers

如图7所示，通过实验对比了不同场景下，采用不同负载均衡方法得到的前后边缘服务器到达任务的平均响应时延。结果表明，与不进行负载均衡的初始平均响应时延相比，使用DG-CBE方法后，在不同场景下该指标平均可降低47.1%。DG-CBE在不同场景下对到达任务的平均响应时延的优化效果比TDB-EC，RF-CLB，C-DQN和Rule-Based方法平均提高了0.6%，2.1%，5.1%和12.9%。由此可见，DG-CBE方法在最小化场景最大到达任务平均响应时延的同时，可以很好地兼顾到达任务的平均响应时延的优化。

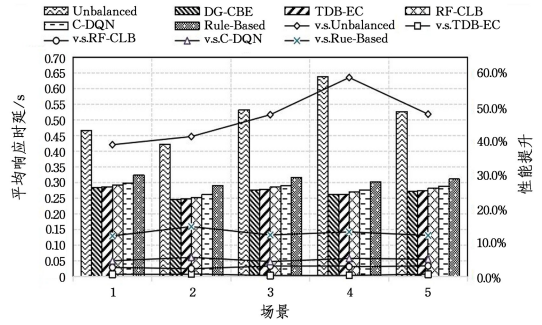


图7 DG-CBE和其他算法在到达任务的平均响应时延上的对比

Fig. 7 Comparison between DG-CBE and other methods in terms of the average response time of arrival tasks at all edges servers

表4列出了不同方法获得目标负载均衡方案的所需时间。

表4 寻找目标负载平衡计划的时间

Table 4 Time to find the objective load balancing plan

(s)

Scenario	PSO-GA	DG-CBE	TDB-EC	RF-CLB	C-DQN	Rule-Based
1	6723	1.823	3.80	1.29	1.54	0.103
2	6929	1.712	3.76	1.34	1.45	0.132
3	6792	1.689	3.67	1.52	1.51	0.120
4	6830	1.823	3.88	1.34	1.49	0.113
5	6991	1.703	3.76	1.46	1.53	0.112

其中，PSO-GA，DG-CBE，TDB-EC，RF-CLB，C-DQN和Rule-Based方法的平均所需时间分别为6853s，1.75s，3.77s，1.39s，1.50s和0.116s。启发式方法PSO-GA结合了遗传算法和粒子群优化的优势，在平衡了多样性和收敛性的同时，利用全局搜索和局部可以搜索出较高质量的负载均衡解。

然而，PSO-GA 求解过程中往往需要耗费大量的迭代和探索时间，过高的求解时间开销使得该方法难以适应无线城域网中不断变化的多边缘负载均衡环境。

综上所述，在不同边缘场景中，通过 DG-CBE 方法得到的最大任务平均响应时延和平均任务响应时延均小于 4 种对比方法。虽然通过 DG-CBE 方法优化的两个时延指标均略差于理想方案，但是 DG-CBE 方法仅需较短的算法执行时间就可以得到接近理想方案效果的负载均衡方案。综合考虑算法的性能和开销，DG-CBE 方法可以有效地解决多边缘协作场景中的负载均衡问题，通过控制边缘服务器之间正确地卸载任务，可以达到良好的负载均衡效果。

结束语 本文提出了一种基于图强化学习的方法来解决 MEC 中的多边缘协同负载均衡问题。首先将多边缘负载均衡场景抽象为图数据，引入 GCN 并通过图节点信息嵌入过程提取图信息，以辅助 DQN 进行推理计算和策略学习。最终通过全局代理对场景进行集中式反馈控制，逐步找到多边缘协作的目标负载均衡方案。此外，基于现实基站的分布情况，设计了仿真实验，并验证了 DG-CBE 方法的有效性。实验结果表明，DG-CBE 方法可以在较短的时间内得到接近理想方案的负载均衡效果，且优于对比方法。在未来的工作中，将在考虑边缘场景中不可动态拆分的依赖性任务的情况下，进一步优化负载均衡算法，增强 DG-CBE 方法的泛用性。

参 考 文 献

[1] YAO Z, LIN J, HU J, et al. PSO-GA based approach to multi-edge load balancing[J]. Computer Science, 2021, 48(S2): 456-463.

[2] XU J, CHEN L, REN S. 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.

[3] ALASMARI K R, GREEN R C, ALAM M. Mobile edge offloading using markov decision processes[C]// Edge Computing-EDGE 2018: Second International Conference, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, June 25－30, 2018, Proceedings 2. Springer International Publishing, 2018: 80-90.

[4] SUN Z, MO Y, YU C. Graph reinforcement learning based task offloading for multi-access edge computing[J]. IEEE Internet of Things Journal, 2023, 10(4): 3138-3150.

[5] CABRERA A, ACOSTA A, ALMEIDA F, et al. A dynamic multi-objective approach for dynamic load balancing in heterogeneous systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(10): 2421-2434.

[6] MEHRABI A, SIEKKINEN M, YLÄ-JÄÄSKI A. Edge computing assisted adaptive mobile video streaming[J]. IEEE Transactions on Mobile Computing, 2018, 18(4): 787-800.

[7] WANG Z, ZHANG W, JIN X, et al. An optimal edge server placement approach for cost reduction and load balancing in intelligent manufacturing[J]. The Journal of Supercomputing, 2022, 78(3): 4032-4056.

[8] YANG L, YAO H, WANG J, et al. Multi-UAV-enabled load-balance mobile-edge computing for IoT networks[J]. IEEE Internet of Things Journal, 2020, 7(8): 6898-6908.

[9] JIA M, LIANG W, XU Z, et al. Qos-aware cloudlet load balancing in wireless metropolitan area networks[J]. IEEE Transactions on Cloud Computing, 2018, 8(2): 623-634.

[10] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2018, 68(1): 856-868.

[11] XU X, LIU K, DAI P, et al. Joint task offloading and resource optimization in noma-based vehicular edge computing: A game-theoretic drl approach[J]. Journal of Systems Architecture, 2023, 134: 102780.

[12] LIU Q, XIA T, CHENG L, et al. Deep reinforcement learning for load-balancing aware network control in IoT edge systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(6): 1491-1502.

[13] LI J, LUO G, CHENG N, et al. An end-to-end load balancer based on deep learning for vehicular network traffic control[J]. IEEE Internet of Things Journal, 2018, 6(1): 953-966.

[14] XU Y, XU W, WANG Z, et al. Load balancing for ultradense networks: A deep reinforcement learning-based approach[J]. IEEE Internet of Things Journal, 2019, 6(6): 9399-9412.

[15] CHEN X, HU J, CHEN Z, et al. A reinforcement learning-empowered feedback control system for industrial internet of things[J]. IEEE Transactions on Industrial Informatics, 2021, 18(4): 2724-2733.

[16] LI J, GAO H, LV T, et al. Deep reinforcement learning based computation offloading and resource allocation for MEC[C]// 2018 IEEE Wireless Communications and Networking Conference(WCNC). IEEE, 2018: 1-6.

[17] ZHANG S, GU H, CHI K, et al. Drl-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network[J]. IEEE Transactions on Wireless Communications, 2022, 21(12): 10934-10948.

[18] LI K, NI W, YUAN X, et al. Deep-Graph-Based Reinforcement Learning for Joint Cruise Control and Task Offloading for Aerial Edge Internet of Things(EdgeIoT)[J]. IEEE Internet of Things Journal, 2022, 9(21): 21676-21686.

[19] CHEN J, YANG Y, WANG C, et al. Multitask offloading strategy optimization based on directed acyclic graphs for edge computing[J]. IEEE Internet of Things Journal, 2021, 9(12): 9367-9378.

[20] DAI A, LI R, ZHAO Z, et al. Graph convolutional multi-agent reinforcement learning for UAV coverage control[C]// 2020 International Conference on Wireless Communications and Signal Processing(WCSP). IEEE, 2020: 1106-1111.

[21] ZHANG J, DU J, SHEN Y, et al. Dynamic computation offloading with energy harvesting devices: A hybrid-decision-based deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(10): 9303-9317.

[22] GAO Z, YANG L, DAI Y. Fast Adaptive Task Offloading and

Resource Allocation in Large-Scale MEC Systems via Multi-Agent Graph Reinforcement Learning[J]. IEEE Internet of Things Journal, 2023, 11(1): 758-776.

[23] CAO X, TANG G, GUO D, et al. Edge federation: Towards an integrated service provisioning model[J]. IEEE/ACM Transactions on Networking, 2020, 28(3): 1116-1129.

[24] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[25] WANG S, ZHAO Y, XU J, et al. Edge server placement in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2019, 127: 160-168.

[26] WANG S, ZHAO Y, HUANG L, et al. QoS prediction for service recommendations in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2019, 127: 134-144.

[27] GUO Y, WANG S, ZHOU A, et al. User allocation-aware edge cloud placement in mobile edge computing[J]. Software: Practice and Experience, 2020, 50(5): 489-502.

[28] WANG S, GUO Y, ZHANG N, et al. Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach[J]. IEEE Transactions on Mobile Computing, 2019, 20(3): 939-951.

[29] VILAPLANA J, SOLSONA F, TEIXIDÓ I, et al. A queuing theory model for cloud computing[J]. The Journal of Supercomputing, 2014, 69: 492-507.

[30] CHEN X, YAO Z, CHEN Z, et al. Load Balancing for Multi-Edge Collaboration in Wireless Metropolitan Area Networks: A Two-Stage Decision-Making Approach[J]. IEEE Internet of Things Journal, 2023, 10(19): 17124-17136.

[31] WU Y, GUO K, HUANG J, et al. Secrecy-based energy-efficient data offloading via dual connectivity over unlicensed spectrums [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3252-3270.

ZHENG Longhai, born in 1999, postgraduate. His main research interests include load balancing and task offloading.

CHEN Xing, born in 1985, Ph.D, professor, Ph.D supervisor, is a senior member of CCF(No. 35725M). His main research interests include software engineering, system software and cloud computing.

(责任编辑:喻藜)