

无人机辅助下的多边缘博弈负载均衡方法

沈启航^{1,3}, 翁杰^{1,3}, 林兵^{2,3}, 陈星^{1,3}

¹ (福州大学 计算机与大数据学院, 福州 350116)

² (福建师范大学 物理与能源学院, 福州 350117)

³ (福建省网络计算与智能信息处理重点实验室(福州大学), 福州 350116)

E-mail: chenxing@fzu.edu.cn

摘要: 随着第五代移动通信技术和物联网的不断发展, 用户对于计算任务的时延响应和服务质量要求不断提高. 边缘计算通过在近终端设备侧部署服务器集群, 为物联网设备提供了低时延、高响应的计算服务能力. 为了解决流量激增所导致的边缘服务质量骤降的问题, 本文提出了一种无人机辅助下的多边缘负载均衡的网络系统模型, 并定义了负载均衡博弈和无人机接入的联合优化问题. 针对该问题, 提出了无人机自适应分布式负载均衡算法(UAV Adaptive Distributed Load Balancing Algorithm, UADLBA), 通过无人机的自适应接入策略和基站边缘的非合作博弈制定最优的负载均衡策略. 实验结果表明, 相较于基准方法, 所提出的方法能够显著降低任务完成时延, 并适应高负载的边缘计算场景.

关键词: 边缘计算; 负载均衡; 博弈论; 无人机部署

中图分类号: TP301

文献标识码: A

文章编号: 1000-1220(2026)04-0982-08

Game Theory-based Load Balancing in UAV-assisted Multi-edge Networks

SHEN Qihang^{1,3}, WENG Jie^{1,3}, LIN Bing^{2,3}, CHEN Xing^{1,3}

¹ (College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China)

² (College of Physics and Energy, Fujian Normal University, Fuzhou 350117, China)

³ (Fujian Key Laboratory of Network Computing and Intelligent Information Processing (Fuzhou University), Fuzhou 350116, China)

Abstract: With the continuous development of the fifth generation mobile communication technology and Internet-of-Things (IoT), the user's requirements for the latency response and quality of service for computing tasks are increasing. The edge computing provides low-latency and high-response computing service capabilities for IoT devices by deploying server clusters on the near-terminal device side. In order to solve the problem of plummeting edge quality of service caused by traffic surge, this paper proposes a network system model for multi-edge load balancing with the assistance of unmanned aerial vehicles (UAV), and defines a joint optimization problem for load balancing game and UAV access. To address this problem, the UAV Adaptive Distributed Load Balancing Algorithm (UADLBA) method is proposed to formulate the optimal load balancing strategy through the adaptive access strategy of UAVs and the non-cooperative game at the edge of the base station. Experimental results show that the proposed method can significantly reduce the task completion delay and adapt to high-load edge computing scenarios compared to the benchmark method.

Keywords: edge computing; load balance; game theory; UAV deployment

0 引言

近年来,随着第五代移动通信技术的进步和物联网的迅猛发展,物联网设备已普遍应用于众多行业,极大地提升了我们的日常生活质量^[1]. 然而,受制于终端设备(UE)在计算能力、存储容量和电池续航等方面的固有局限^[2],面对计算密集型与时延敏感型任务时,现有架构难以满足用户对服务时延、能耗效率和可靠性的多维需求. 这一技术瓶颈促使学术界和产业界积极探索新型计算架构以应对这些挑战.

移动云计算(MCC)^[3]通过将终端任务卸载至云端数据中心进行处理,利用其强大的计算集群和弹性存储资源有效

处理海量计算请求^[4,5]. 但该架构存在显著缺陷:终端需通过广域网与远端云服务器进行数据交换^[6],这种长距离传输不仅引入额外通信时延,还可能导致数据包丢失率上升. 特别是在动态网络环境下,难以保障时延敏感型任务的服务质量(QoS)^[7].

为解决上述问题,移动边缘计算(MEC)这一革新性架构应运而生. 欧洲电信标准化协会(ETSI)于2014年提出的MEC框架^[8,9],其核心理念是将计算节点下沉至网络边缘侧. 这种分布式架构通过将服务器部署于基站侧,使计算资源与终端设备的空间距离缩短至单跳范围内. 这种拓扑优化带来了双重技术优势:一方面,端到端传输时延的显著降低(通

常可达毫秒级)有效提升了实时任务的响应性能,另一方面,还大幅减少了回传链路的流量负载,从而优化了整体网络能效^[10]。

尽管移动边缘计算能够克服移动云计算的缺点,但在实际应用中,也面临新的挑战。计算资源的大小通常在每个安装 MEC 服务器的基站中是预先设置的,而用户的资源需求是实时变化的,特别当存在一些社会活动或事件引起的突发流量时,固定化的资源供给与弹性需求之间的失配将导致系统过载,进而引发任务响应延迟的指数级增长^[11]。为了解决这一问题,一种可行的解决方案是通过迁移边缘服务器的计算任务,均衡边缘服务器的任务负载。然而,由于 MEC 服务器部署位置的分散性和计算资源的有限性,如何设计有效的负载均衡策略成为一项具有挑战性的研究课题。另一种可行的解决方案是利用无人机的灵活性优势,通过在无人机上部署小型 MEC 服务器,通过在用户侧的任务卸载、计算资源分配和无人机轨迹控制等方法以补充 UE 的计算资源来解决上述问题^[11-13],许多公司的实际项目中也已经考虑了相关应用^[14]。

近年来,学术界对边缘服务器负载均衡问题已被广泛研究。众多研究通过基站边缘服务器在不同场景下制定负载均衡策略以降低网络的负载。例如 Cabrer^[15] 等人提出了一种多目标动态负载均衡 (DLB) 方法以适应计算密集型任务在异构边缘系统中的高效负载均衡。Liu^[16] 等人使用基于深度强化学习 (DRL) 的物联网网络动态聚类算法,可以同时实现物联网网络的通信负载均衡和边缘服务器的计算负载均衡。Yang^[17] 等人提出了一种基于深度确定性策略梯度 (DDPG) 的辅助聚类任务卸载方法,以实现边缘服务器之间的负载均衡。然而,上述工作主要依赖于集中式的控制方法进行负载均衡,这意味着边缘服务器需要以全局的优化目标为导向,遵循中心控制系统的统一调度指令。但集中式的控制方法通常需要花费大量时间搜索合适的负载均衡方案,这种低效率难以满足 UE 对实时服务的需求。为了降低决策时延并优化负载均衡性能,部分研究聚焦于采用分布式架构设计边缘负载均衡调度策略。例如 Mattia^[18] 等人通过构建系统的微分模型,提出了一种自适应启发式算法来搜索负载均衡方案,以去中心化的方式平衡边缘节点的延迟。Xu^[19] 等人针对多边缘系统制定了一个基于 DNN 的推理模型,定义了负载均衡的最优推理卸载,在此基础上提出了一种用于 DNN 推理卸载的分布式算法,得到负载均衡的最优推理卸载策略。但现有基站负载均衡研究受限于静态资源配置模式,难以应对区域突发流量导致的极限负载场景。

得益于无人机的高机动性和快速部署能力,其可作为移动边缘计算节点或空中中继单元,动态构建临时计算资源网络。因此,近年来部分研究开始聚焦于无人机在 UE 侧的边缘计算中的应用。Cheng^[20] 等人研究了如何部署和调度搭载 MEC 的无人机以扩展 UE 的计算能力。Peng^[13] 等人针对无人机-车辆网络系统的资源分配优化问题,提出了基于多智能体深度确定性策略梯度 (MADDPG) 的分布式方法,以最大化卸载任务的数量。Sun^[12] 等人构建了一个多目标优化模型,将问题分解为 3 个子问题,通过结合 Karush-Kuhn-Tucker (KKT) 和连续凸近似 (SCA) 算法,实现了任务完成时延最小化、无人机整体能耗降低以及任务卸载量最大化的综合优化

目标。然而,上述无人机辅助边缘计算的相关研究仅关注无人机与 UE 的计算卸载,缺乏与地面基站的协同优化机制。

综上所述,当前基站边缘负载均衡研究面临显著挑战:现有工作对区域流量突发性增长的特殊场景考虑不足,当基站集群在负载均衡后仍处于极限工作状态时,将引发服务质量急剧下降及任务丢弃等问题。基于此,本文提出了一种无人机自适应分布式负载均衡算法。利用无人机的机动性、灵活部署和视距通信优势,为传统地面基站提供计算资源的动态补充。为实现基站边缘与无人机边缘的高效协同,本文系统研究了无人机辅助边缘环境下的负载均衡优化问题。主要贡献总结如下:

1) 空地协同负载均衡框架:本文提出多无人机辅助的多边缘服务器系统架构,融合基站静态负载均衡与无人机动态资源调度的协同模型。突破传统基站静态负载均衡优化局限,利用无人机的机动性建立动态计算资源池,缓解基站计算瓶颈,并通过联合优化实现负载的时空均衡分布,以适应极限高负载的边缘计算场景。

2) 分层优化与博弈建模:针对优化问题的复杂性,本文将其解耦为基站边缘服务器负载均衡与无人机接入策略两个协同优化的子问题。首先将基站负载均衡建模为具有纳什均衡特性的非合作博弈模型,进而提出无人机自适应分布式负载均衡算法。该方法通过无人机的动态接入策略智能选择最优基站进行辅助,同时严格证明了负载均衡博弈可收敛至纳什均衡点,在理论上保障了系统稳定性。

3) 实验验证与性能优势:为验证算法有效性,本文构建了高负载边缘计算仿真环境,选取 PSOGA 和 SBOA 作为基准对比方法。实验结果表明,UADLBA 在执行时间和平均任务完成时延上均能取得更好的优化效果,充分验证了算法在极端场景下的适用优势。相较于传统静态优化方法,UADLBA 通过空-地资源协同实现了负载的时空动态均衡,适用于高动态边缘计算场景。

本文第 1 节给出系统的网络模型并对问题进行形式化定义;第 2 节证明了构建的博弈模型存在纳什均衡点;第 3 节详细介绍了 UADLBA 方法以及给出 UADLBA 解决无人机接入和边缘负载均衡的方法;第 4 节描述了仿真实验并对实验结果进行分析;第 5 节总结全文内容。

1 系统模型和问题定义

1.1 系统网络模型

如图 1 所示,本文考虑一个由众多搭载边缘服务器的基站以及多架无人机联接而成的网络架构,在此架构中,每个基站充当核心边缘节点,而无人机作为辅助边缘节点。同时,基站与无人机所搭载的边缘服务器采用异构处理器,即二者提供的计算能力有所差异。UE 能够基于自身与邻近核心边缘节点的位置关系,将无法单独完成的计算任务卸载给核心边缘计算节点,以利用核心边缘节点的计算资源。此外,系统中的无人机作为辅助边缘节点,可以通过合理的位置部署策略,接收高负载核心边缘节点中的计算任务。

在该网络系统中,所包含的 M 个基站和 N 个无人机上的边缘服务器集合分别用 $\mathcal{M} = \{e_1^c, e_2^c, \dots, e_M^c\}$ 和 $\mathcal{N} = \{e_1^u,$

e_2^u, \dots, e_N^u 来表示, 基站边缘 e_i^c 与无人机边缘 e_i^u 的位置分别用 $p_i^c = (x_i^c, p_i^c, 0)$ 和 $p_i^u = (x_i^u, p_i^u, H)$ 来表示. 类似于 Lin 工作^[21] 中的设计, 本文假设用户终端设备卸载到核心边缘节点的计算任务遵循泊松分布模型, 并且把这些任务定义为“到达任务”. 用集合 $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_M\}$ 表示基站边缘服务器的任务到达率, 其中 λ_i 表示基站边缘服务器 e_i^c 上的任务到达率, 即单位时间内边缘服务器接收用户终端设备的任务到达量. 本文符号定义如表 1 所示.

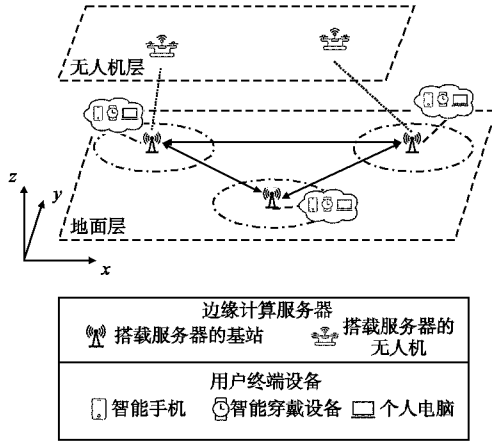


图 1 系统模型示意图
Fig. 1 System model diagram

传统边缘计算系统面临的根本矛盾在于, 边缘服务器的静态资源预配置(如基站固定计算容量)与用户需求的动态突发性(峰值可能超过基站总计算容量). 现有解决方案^[22] 主要依赖整体任务迁移策略, 但这种方法存在明显缺陷, 简单地将任务从过载节点转移到低负载节点, 可能导致接收节点发生二次过载, 本质上只是转移而非解决了负载不均衡问题. 针对这一问题, 本文提出的空地协同负载均衡框架采用了非完全任务卸载策略, 边缘节点的到达任务需要根据地面基站和无人机的实时负载状态自动调整划分粒度, 通过将这些细化后的子任务分散到各个边缘节点上, 以实现基站边缘与无人机边缘的协同负载均衡.

1.2 传输模型

1.2.1 基站边缘传输模型

假设基站边缘节点通过专属的电缆网络相互链接, 每一条链接都有其独立的信道提供支持, 数据在这些信道中以序列串行化的形式进行传输, 以此来维护数据的完整性和传输的次序. 同时, 这些网络链接实现了双向通信, 并且不论数据传递的方向如何, 其通信的延迟保持一致. 与现有的研究^[23] 类似, 本文将单位任务在基站边缘服务器之间的传输时间定义为:

$$D^c = \begin{bmatrix} d_{1,1}^c & \dots & d_{1,M}^c \\ \vdots & \ddots & \vdots \\ d_{M,1}^c & \dots & d_{M,M}^c \end{bmatrix} \quad (1)$$

其中 $d_{i,j}^c$ 表示基站边缘服务器 e_i^c 将单位任务卸载到基站边缘服务器 e_j^c 所需的时间, 特别地, 如果基站边缘服务器 e_i^c 与基站边缘服务器 e_j^c 没有连接, 则有 $d_{i,j}^c = 0$, 将与边缘 e_i^c 相连的

基站边缘服务器集合 L_i^c 定义为:

$$L_i^c = \{e_j^c | e_j^c \in \mathcal{N}, d_{i,j}^c \neq 0\} \quad (2)$$

表 1 符号定义

Table 1 Symbol definition

变量符号	变量含义
\mathcal{M}	基站边缘服务器集合
\mathcal{N}	无人机边缘服务器集合
M	基站数量
N	无人机数量
D^c	基站边缘服务器的传输时间
D^u	基站与无人机边缘服务器的传输时间
L_i^c	与边缘 e_i^c 相连的基站边缘服务器集合
L_i^u	接入边缘 e_i^c 的无人机边缘服务器集合
R^{c2u}	地面边缘的上行传输数据速率
X_i	基站边缘 e_i^c 的卸载向量
X_i^{c2c}	基站边缘 e_i^c 的基站卸载向量
X_i^{c2u}	基站边缘 e_i^c 的无人机卸载向量
F^c, F^u	基站和无人机边缘服务器的服务速率
W^c	基站边缘的负载任务到达率
p_i^c, p_i^u	基站边缘 e_i^c 与无人机边缘 e_i^u 的位置
e_i^c	基站边缘服务器中的第 i 个服务器
e_i^u	无人机边缘服务器中的第 i 个服务器
λ_i	基站边缘 e_i^c 的任务到达率
$d_{i,j}^c$	单位任务在边缘 e_i^c 和 e_j^c 的传输时间
$d_{i,j}^u$	单位任务在边缘 e_i^c 和 e_j^u 的传输时间
g^{c2u}	地面基站的上行信道增益
$x_{i,j}^{c2c}$	边缘 e_i^c 向各个基站边缘卸载的任务量
$x_{i,j}^{c2u}$	边缘 e_i^c 向无人机边缘卸载的任务量
w_i^c	边缘 e_i^c 的负载任务到达率
T_i^{wait}	边缘 e_i^c 的任务平均排队等待时延
T_i^{proc}	边缘 e_i^c 的任务计算执行时延
T_i^{resp}	边缘 e_i^c 的任务响应时延
$T_{i,j}^{total}$	边缘 e_i^c 和 e_j^c 的迁移任务完成时间

1.2.2 无人机边缘传输模型

假设无人机到基站的通信信道主要由视距 (LOS) 链路信道主导, 此外无人机可以通过正交频分复用接入地面基站, 因此无人机-地面基站链路之间的干扰可以被忽略. 从地面基站到无人机的上行信道增益可以用自由空间路径损耗模型表示为:

$$g^{c2u}(e_i^c, e_j^u) = \alpha_0 d(e_i^c, e_j^u)^{-2} \quad (3)$$

其中 α_0 表示 1 m 距离参考处的信道增益常数, $d(e_i^c, e_j^u)$ 表示基站边缘服务器 e_i^c 到无人机边缘服务器 e_j^u 的上行距离, 根据香农公式, 从地面边缘 e_i^c 到无人机边缘 e_j^u 的上行传输数据速率可以由下式计算得出:

$$R^{c2u}(e_i^c, e_j^u) = B \log_2 \left(1 + \frac{g^{c2u}(e_i^c, e_j^u) P^c}{\sigma^2} \right) \quad (4)$$

其中 σ^2 表示白高斯噪声方差, B 表示信道带宽, P^c 表示地面基站的发送功率. 本文假设由边缘服务器卸载过的任务不会进行二次卸载, 并且计算结果的输出数据量通常远小于输入数据量, 因此由无人机边缘服务器下行传输时间可以忽略不计. 因此基站边缘和无人机边缘之间的传输时间 D^u 可以定义为:

$$D^u = \begin{bmatrix} d_{1,1}^u & \cdots & d_{1,N}^u \\ \vdots & \ddots & \vdots \\ d_{M,1}^u & \cdots & d_{M,N}^u \end{bmatrix} \quad (5)$$

其中, $d_{i,j}^u = \frac{1}{R^{2u}(e_i^c, e_j^u)}$, 表示基站边缘服务器 e_i^c 将单位任务卸载到无人机边缘服务器 e_j^u 所需的时间。

1.3 任务卸载模型

若用 X_i 表示为基站边缘 e_i^c 的卸载向量, 则边缘服务器间的任务卸载矩阵 X 可以定义为:

$$X = \begin{bmatrix} X_1 & X_1^{2c} & X_1^{2u} \\ \vdots & \vdots & \vdots \\ X_M & X_M^{2c} & X_M^{2u} \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \cup \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad (6)$$

其中, X_i^{2c} 为基站边缘 e_i^c 的基站卸载向量, 即基站边缘 e_i^c 向各个基站边缘卸载的任务量, 可以表示为:

$$X_i^{2c} = [x_{i,1}^{2c} \quad \cdots \quad x_{i,M}^{2c}] \quad (7)$$

特别地, 当 $i=j$ 时, $x_{i,j}^{2c}$ 表示基站边缘留在本地需要执行的任务量. X_i^{2u} 为基站边缘 e_i^c 的无人机卸载向量, 即基站边缘 e_i^c 向无人机边缘卸载的任务量, 可以表示为:

$$X_i^{2u} = [x_{i,1}^{2u} \quad \cdots \quad x_{i,N}^{2u}] \quad (8)$$

考虑到无人机在投入使用时仅作为基站边缘的辅助节点, 为避免部分基站边缘服务器的瞬时请求流量过大, 导致计算资源短缺, 本文假设无人机在辅助过程中的时隙内只会接入一个基站边缘节点, 并位于其正上空进行辅助计算. 因此, 当无人机边缘服务器 e_j^u 未接入基站边缘服务器 e_i^c 时, $x_{i,j}^{2u} = 0$, 并且将接入基站边缘服务器 e_i^c 的无人机边缘服务器定义为 L_i^u .

1.4 计算模型

本文将基站边缘服务器和无人机边缘服务器的服务速率分别用集合 $F^c = \{f_1^c, \dots, f_M^c\}$ 和 $F^u = \{f_1^u, \dots, f_N^u\}$ 来表示, 其中 f_i^c 与 f_i^u 表示基站边缘服务器 e_i^c 和无人机边缘服务器 e_i^u 的计算服务速率.

本文用集合 $W^c = \{w_1^c, \dots, w_M^c\}$ 来表示基站边缘的负载任务到达率, 包含边缘服务器在本地执行的任务以及其他边缘服务器向其迁移的任务. w_i^c 可由下式计算得出:

$$w_i^c = \lambda_i + \sum_{e_j^c \in L_i^c, e_j^c \neq e_i^c} x_{j,i}^{2c} - \sum_{e_j^c \in L_i^c, e_j^c \neq e_i^c} x_{i,j}^{2c} - x_{i,k}^{2u} \quad (9)$$

需要注意的是, 当无人机边缘 e_k^u 未接入基站边缘 e_i^c 时, 则有 $x_{i,k}^{2u} = 0$. 类似的, 将无人机边缘 e_k^u 的负载任务到达率定义为 w_k^u . 本文假设任务将首先卸载到地面基站边缘, 并将任务分割成子任务, 而分割任务的决策时间与整个通信和计算延迟相比非常短, 因此可以忽略不计. 此外, 许多计算密集型应用 (如人脸识别和视频分析) 中, 计算结果的输出数据量通常远小于输入数据量. 因此, 将计算结果发送回用户设备所需的时间也可以忽略不计.

综上所述, 卸载过程中的总时延可以描述为以下四个关键组成部分: 1) 基站边缘之间的专线传输时延; 2) 基站边缘上的任务响应时延; 3) 从基站边缘到无人机边缘的上行传输延迟; 4) 无人机边缘上的任务响应时延.

考虑到边缘节点在处理用户产生的计算任务过程中会面临大量的并发请求, 并且边缘节点的计算资源有限, 在应对这

些挑战时, 排队现象是不可避免的, 因此, 本文基于排队论, 针对边缘服务器的服务系统建立 M/M/1 的排队模型. 基站边缘服务器 e_i^c 上的任务平均排队等待时延 T_i^{wait} 可以表示为:

$$T_i^{wait}(w_i^c) = \frac{w_i^c}{f_i^c(f_i^c - w_i^c)} \quad (10)$$

任务 w 的计算执行时延 T_i^{proc} 可以表示为:

$$T_i^{proc}(w) = \frac{w}{f_i^c} \quad (11)$$

边缘服务器上的任务响应时延包括排队等待时延和计算执行时延两个部分. 因此, 任务 w 在基站边缘服务器 e_i^c 上的响应时延 T_i^{resp} 可以表示为:

$$T_i^{resp}(w, w_i^c) = T_i^{wait}(w_i^c) + T_i^{proc}(w) \quad (12)$$

由此可以得到基站边缘服务器 e_i^c 迁移到基站边缘服务器 e_j^c 执行的任务平均完成时间可以由下式计算得出:

$$T_{i,j}^{total}(x_{i,j}^{2c}, w_j^c, f_j^c, d_{i,j}^c) = \frac{1}{\lambda_i} \left[\frac{w_j^c}{f_j^c(f_j^c - w_j^c)} + \frac{x_{i,j}^{2c}}{f_j^c} + x_{i,j}^{2c} d_{i,j}^c \right] \quad (13)$$

类似的, 可以求得基站边缘服务器 e_i^c 迁移到无人机边缘服务器 e_k^u 的任务平均完成时间 $T_{i,k}^{total}(x_{i,k}^{2u}, w_k^u, f_k^u, d_{i,k}^u)$.

1.5 基于博弈的问题形式化

本文旨在研究边缘服务器在追求自身利益最大化的理性决策行为下的系统优化问题, 优化目标是为了最大限度地减少所有边缘节点上任务的平均完成时间, 并实现整个边缘系统的负载均衡. 但是基站边缘和无人机边缘的直接接入对象不同, 并且任务传输过程也存在差异, 要同时对基站服务器和无人机服务器进行负载均衡十分复杂. 考虑到无人机边缘服务器在较短的时隙内只接入一个基站边缘服务器, 并且都采用排队论对服务过程进行抽象, 将基站边缘卸载给无人机边缘的任务量设为 δ , 则由式 (13) 可以得到无人机边缘 e_k^u 上单位任务的平均完成速率:

$$f_k^u = \frac{1}{T_{i,k}^{total}(\delta, \delta, f_k^u, d_{i,k}^u)} \quad (14)$$

因此可以将无人机边缘计算卸载任务的服务过程整合到基站边缘的队列服务中, 将原问题转化为优化基站边缘服务器的负载均衡卸载方案和无人机的接入方案两个子问题.

定义基站边缘服务器 e_i^c 的效用函数:

$$Q_i = \frac{1}{\lambda_i} \left\{ \sum_{e_j^c \in L_i^c} \left[\frac{w_j^c}{f_j^c(f_j^c - w_j^c)} + \frac{x_{i,j}^{2c}}{f_j^c} \right] + \sum_{e_j^c \in L_i^c, e_j^c \neq e_i^c} x_{i,j}^{2c} d_{i,j}^c \right\} \quad (15)$$

其中 $f_j^c = f_j^c + \sum_{e_k^u \in L_i^u} f_k^u$, 表示接入辅助无人机边缘的基站边缘服务器的平均任务服务速率. 因此, 每个基站边缘服务器在系统中的个体目标为: 依据当前网络状况及邻接基站边缘服务器的卸载策略 X_{-i} , 动态调整自身任务分配决策, 探寻最优任务迁移方案 X_i , 以实现自身效用函数的最小化. 对于基站边缘服务器 e_i^c , 其最优化问题可以定义为:

$$\min_{X_i, L_i^u} Q_i(X_i, X_{-i}) \quad (16)$$

$$\text{s. t. C1: } x_{i,j}^{2c} \geq 0 \quad (16a)$$

$$\text{C2: } \sum_{e_j^c \in L_i^c} x_{i,j}^{2c} = \lambda_i \quad (16b)$$

$$C3: \sum_{e_j^c \in L_i^c} x_{i,j}^{2c} < f_i' \quad (16c)$$

$$C4: L_i^u = \emptyset \mid L_i^u \subseteq \mathcal{N} \quad (16d)$$

$$C5: L_i^u \cap L_j^u = \emptyset \quad (16e)$$

其中, C1 表示基站边缘服务器的迁移任务量是非负的, C2 表示基站边缘服务器卸载前后总任务量保持不变, C3 表示基站边缘服务器的总任务量不能超过最大计算能力, C4 表示基站边缘服务器有 0 或多个无人机边缘参与辅助计算, C5 表示无人机边缘只能接入一个基站边缘。

2 博弈制定与证明

2.1 博弈制定

在本文所研究的多边缘节点的边缘计算系统中,边缘服务器与邻接节点通过局部信息交互来实现协同工作。在博弈中,边缘服务器都是自私的,它们根据相邻边缘的局部信息,尽可能最小化自身效用函数。鉴于这一特性,本文将边缘服务器在任务迁移过程中的竞争行为抽象为一个不完全信息的非合作博弈模型。在此模型框架下,每台边缘服务器作为博弈的独立参与者,仅掌握系统的片段信息。它们需凭借这些有限的局部信息,推测其他参与者的潜在策略,进而寻求最优的任务迁移策略,旨在缩短自身任务的平均任务完成时间。因此,本文将这一问题形式化为一个参与者地位对等的非合作博弈 G , 博弈表达式为:

$$G = \langle \mathcal{M}, \{X_i\}_{e_i^c \in \mathcal{M}}, \{Q_i\}_{e_i^c \in \mathcal{M}} \rangle \quad (17)$$

其中, \mathcal{M} 为参与博弈的基站边缘服务器集合, X_i 为基站边缘服务器 e_i^c 的卸载策略集, Q_i 为边缘服务器 e_i^c 的效用函数。

在博弈论中,纳什均衡被视为评估系统稳定性的关键指标。以下给出本文边缘协同负载均衡问题的纳什均衡点定义:

定义 1. 定义卸载策略 $X^* = \{X_1^*, \dots, X_M^*\}$, 为博弈的纳什均衡点。此时,任意博弈参与者 e_i^c 的策略 $\{X_i^*\}_{e_i^c \in \mathcal{M}}$ 均满足:

$$Q_i(X_i^*, X_{-i}^*) \leq Q_i(X_i', X_{-i}^*) \quad (18)$$

其中, $\{X_i'\}_{e_i^c \in \mathcal{M}}$ 为博弈参与者 e_i^c 的任意策略, $X_{-i}^* = \{X_j^*\}_{e_j^c \in \mathcal{M}, e_j^c \neq e_i^c}$ 。

由定义 1 可知,当系统处于纳什均衡时,任何一个参与者改变自身的策略均无法降低系统成本。

2.2 证明纳什均衡点存在性

本文基于无人机辅助的多边缘协同负载均衡问题所构建的非合作博弈模型进行深入剖析,采用数学分析与推导的方法,将博弈转化为变分不等式,证明了该模型纳什均衡点的存在性。

定理 1. 对于每个基站边缘服务器 e_i^c , 其卸载策略集 X_i 是闭凸的,且效用函数 Q_i 是连续可微的。

证明: 根据式(16), 每个基站边缘服务器 e_i^c 的卸载策略 X_i 满足约束 C1-C3, 其中 $\{X_i \mid \forall x_{i,j}^{2c} \in [0, \lambda_i]\}$ 。根据闭凸集的定义^[24], 卸载向量 X_i 是闭凸的。

接下来需要证明效用函数的导数的连续性。将边缘服务器 e_i^c 的效用函数改写为:

$$Q_i = \frac{1}{\lambda_i} \left\{ \sum_{e_j^c \in L_i^c} \left[\frac{1}{f_j' - w_j^c} - \frac{1}{f_j'} + \frac{x_{i,j}^{2c}}{f_j^{2c}} \right] + \right.$$

$$\left. \sum_{e_j^c \in L_i^c, e_j^c \neq e_i^c} x_{i,j}^{2c} d_{i,j}^c \right\} \quad (19)$$

令 Q_i 对 $x_{i,j}^{2c}$ 求偏导后可得:

$$\frac{\partial Q_i}{\partial x_{i,j}^{2c}} = \begin{cases} \frac{1}{\lambda_i} \left(\frac{1}{(f_j' - w_j^c)^2} + \frac{1}{f_j'} + d_{i,j}^c \right), & i \neq j \\ \frac{1}{\lambda_i} \left(\frac{1}{(f_j' - w_j^c)^2} + \frac{1}{f_j'} \right), & i = j \end{cases} \quad (20)$$

由于 Q_i 的偏导数存在且连续,说明效用函数 Q_i 是连续可微的。

定理 2. 当其他边缘服务器任务卸载策略保持不变的状态下,当前边缘服务器 e_i^c 的效用函数 Q_i 为一个严格凸函数。

证明: 依据凸函数的基本定义^[25], 需要通过分析和验证效用函数 Q_i 的 Hessian 矩阵 $H(Q_i)$ 的正定性,从而确定该函数满足凸条件。 $H(Q_i)$ 可以表示为:

$$H(Q_i) = \begin{bmatrix} \frac{\partial^2 Q_i(X_i, X_{-i})}{\partial^2 x_{i,1}} & \dots & \frac{\partial^2 Q_i(X_i, X_{-i})}{\partial x_{i,1} \partial x_{i,M}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 Q_i(X_i, X_{-i})}{\partial x_{i,M} \partial x_{i,1}} & \dots & \frac{\partial^2 Q_i(X_i, X_{-i})}{\partial^2 x_{i,M}} \end{bmatrix} \quad (21)$$

其中:

$$\frac{\partial^2 Q_i(X_i, X_{-i})}{\partial x_{i,j} \partial x_{i,k}} = \begin{cases} \frac{2}{(f_j' - w_j^c)^3}, & i = j = k \\ 0, & \text{other} \end{cases} \quad (22)$$

在博弈达到纳什均衡状态(即其他服务器卸载策略保持稳定)时,边缘服务器 e_i^c 的效用函数 Q_i 的 Hessian 矩阵 $H(Q_i)$ 呈现特殊对角结构。结合公式(16)约束条件 C3, 可证明该矩阵的对角元素 $\frac{2}{(f_j' - w_j^c)^3}$ 均为正值,从而满足正定性要求。

基于凸函数判定准则,可得出效用函数 Q_i 具有严格凸性的结论。

定理 3. 当 X_i 满足非空闭凸集约束条件,并且 Q_i 符合连续可微的凸函数特性时,博弈 G 的均衡解求解问题可转化为相应的变分不等式 VI(X, q) 的求解问题。进一步地,当效用函数的梯度 q 满足严格单调性条件时,可确保该博弈至少存在一个纳什均衡点。

证明: 通过初步分析可知, X_i 是一个非空集合。基于定理 1 与定理 2 的论证过程,可确定集合 X_i 还是一个闭凸集,且效用函数 Q_i 具有连续可微的凸函数特性。文献[26]的研究成果深入揭示了博弈论与变分不等式理论之间的本质联系。根据分析结果,博弈 G 的均衡解求解问题可等价于相应的变分不等式 VI(X, q) 的求解问题。接下来,证明效用函数的梯度 q 的单调性。

如果梯度 q 是单调的,则满足以下条件:

$$(X - X^*)^T (q(X) - q(X^*)) \geq 0 \quad (23)$$

等价于:

$$\sum_{i=1}^M (X_i - X_i^*)^T (q_i(X_i) - q_i(X_i^*)) \geq 0 \quad (24)$$

其中:

$$q_i(X_i, X_{-i}) = \nabla_{X_i} Q_i(X_i, X_{-i}) \quad (25)$$

q_i 的雅可比矩阵可以表示为:

$$J_{q_i}(X_i) = \nabla_{X_i}^2 Q_i(X_i) = H(Q_i) \quad (26)$$

基于定理 2 的证明结果,已证明 $H(Q_i)$ 具有正定性,由

此可推导出 $J_{q_i}(X_i)$ 同样为正定矩阵. 这一数学特性确保了函数 q_i 满足严格单调性条件, 通过分析可知, 对于 $\forall i \in M$, 则有 $(X_i - X_i^*)^T (q_i(X_i) - q_i(X_i^*)) \geq 0$, 进而可以推出式 (24) 必然成立, 从而证明梯度函数 q 整体保持严格单调性. 基于上述分析, 结合博弈论基本定理, 可以确定该博弈模型至少存在一个纳什均衡解.

3 算法设计

本文基于无人机辅助的 MEC 场景中多边缘协同负载均衡问题构建了非合作博弈. 在整个过程中, 基站边缘作为博弈参与者, 自私地参与负载均衡, 即不考虑自身策略带给其他邻接基站边缘的决策影响, 无人机边缘作为基站边缘的辅助节点, 为其提供额外的计算资源, 作为博弈参与者的固定成本间接参与博弈过程. 因此, 算法的基本框架是通过迭代算法推导出博弈参与者任务迁移的最优策略选择. 在每轮迭代中, 根据构建出的效用函数优化模型, 求解其最小值, 最后逐步收敛得到所有博弈参与者在任务迁移场景下的最优策略组合.

在卸载过程中, 为了充分利用无人机边缘的灵活性和可扩展性, 本文提出了一种支持无人机辅助场景下的自适应负载均衡算法 (UAV Adaptive Load Balancing Algorithm, UALBA), 具体如算法 1 所示.

算法 1. 自适应负载均衡算法

输入: $\lambda, H, F^c, F^u, D^c, \delta, M, N, \alpha_0$

输出: X^*

1. 初始化邻接边缘向量 L 、根据变量 H, F^u, α_0 初始化基站-无人机时延矩阵 D 、任务量 δ 的无人机任务完成速率 f_j ;
2. **for** $i = 1 : N$ **do**
3. $\mu \leftarrow \emptyset$
4. **for** $k = 1 : M$ **do**
5. $\mu_k \leftarrow \lambda_k / f_k^c$
6. $\mu \cup \mu_k$
7. **end for**
8. $j \leftarrow \text{argmax}_k(\mu)$
9. 将无人机边缘 e_i^u 接入对应基站边缘 e_j^c , 更新基站边缘任务完成速率 f_j^c
10. **end for**
11. $X^* = \emptyset$
12. **for** $i = 1 : M$ **do**
13. $nl \leftarrow |L_i|, X_i \leftarrow \emptyset$
14. 根据 nl 生成邻接卸载向量排列组合集 B , 其中 B_i 是形如 $(0, \dots, 1)$ 的向量, 1 表示与对应索引位置的邻接边缘存在卸载关系, 0 则没有
15. 获取其他边缘卸载策略 $\{X_{-i}\}_{e_i^c \in \mathcal{M}}$
16. **for** $B_j \in B$ **do**
17. 根据 B_j 生成实际卸载向量 O_k
18. 基于 $\lambda_i, f_j^c, O_k, D^c, \{X_{-i}\}_{e_i^c \in \mathcal{M}}$, 使用 CVX 求解器计算得到 X_i^j , 其中 $X_i^j = \text{argmin} \{Q_i(X_i, X_{-i})\}$
19. **end for**
20. $X_i \cup \min \{X_i^j\}$
21. **end for**
22. $X^* \cup X_i$
23. **return** X^*

考虑到边缘进行集中式负载均衡存在频繁的全局状态同步和指数级通信开销的问题, 难以满足边缘服务的低延迟要求. 基于前文构建的博弈模型 G 以及算法 1, 本文进一步设计

了一种无人机自适应分布式负载均衡算法 (UAV Adaptive Distributed Load Balancing Algorithm, UADLBA), 以支持在无人机辅助下完成最优的任务卸载策略. 该算法将算法 1 分为两个阶段, 阶段 1 是无人机的自适应接入策略 (第 2 ~ 10 行), 阶段 2 是基站边缘的非合作博弈 (第 11 ~ 23 行). 其中阶段 1 需要无人机获取所有基站的全局信息, 在部署每架无人机时, 算法可以自适应地发现当前系统负载状态下最合适的基站边缘并接入, 阶段 2 可以以分布式的架构运行在基站服务器上^[21], 在各自的服务器上经过多次迭代找到系统的纳什均衡点. 在算法 2 中给出具体步骤: 开始时, 基站边缘服务器的默认卸载策略为本地处理, 即不进行迁移卸载, 并将服务器的部分运行状态信息广播至邻接边缘服务器 (第 1 行), 接下来, 边缘服务器遍历邻接边缘卸载向量集合, 使用 CVX 求解器计算得到任务卸载策略, 从中选择出最优的任务卸载策略 (第 3 ~ 10 行). 当完成决策后, 与上一轮决策进行比较, 若差距小于界限步长 ε , 则表明达到负载均衡, 停止迭代 (第 11 ~ 14 行), 最后, 将迭代收敛到的卸载策略返回 (第 15 ~ 16 行).

算法 2. 边缘 e_i^c 的分布式负载均衡算法

输入: $\lambda_i, F^c, D^c, \{X_{-i}\}_{e_i^c \in \mathcal{M}}$

输出: X_i^*

1. 初始化边缘的负载率, 最大迭代次数 Max , 初始化邻接边缘向量 L , 将自身服务器的信息广播到邻接边缘服务器上;
2. $X_i^* \leftarrow \emptyset$
3. **for** $iter = 1 : Max$ **do**
4. $nl \leftarrow |L_i|, X_i^{(k)} \leftarrow \emptyset$
5. 根据 nl 生成邻接卸载向量排列组合集 B
6. **for** $B_j \in B$ **do**
7. 根据 B_j 生成实际卸载向量 O_k
8. 基于 $\lambda_i, f_j^c, O_k, D^c, \{X_{-i}\}_{e_i^c \in \mathcal{M}}$, 使用 CVX 求解器计算得到 X_i^j , 其中 $X_i^j = \text{argmin} \{Q_i(X_i, X_{-i})\}$
9. **end for**
10. $X_i^{(k)} \leftarrow \min \{X_i^j\}$
11. **if** $\|X_i^{(k)} - X_i^{(k-1)}\| \leq \varepsilon$ **then**
12. 到达均衡, 停止迭代
13. **end if**
14. **end for**
15. $X_i^* \leftarrow X_i^{(k)}$
16. **return** X_i^*

4 实验结果与分析

为了验证 UADLBA 方法对于无人机辅助 MEC 系统负载均衡的有效性, 本文进行了大量的仿真实验, 并且实现了以下 4 种对比算法: 本地计算策略 (Local)、无人机辅助的本地计算策略 (ULocal)、SBOA^[21] 以及 PSOGA^[27] 方法. 本地计算策略是指无人机不接入基站边缘, 并且基站边缘的任务只在本地执行, 不进行任务卸载; 无人机辅助的本地计算策略是指无人机会使用算法 1 中的自适应接入策略, 但基站边缘也不进行任务卸载; SBOA 方法可以简单理解成无人机不接入基站边缘, 但是基站边缘之间会进行分布式负载均衡决策的方法; PSOGA 方法是指无人机会接入基站边缘, 但是基站边缘之间的任务卸载决策由 PSOGA 获得, PSOGA 算法采用集中式架构设计, 通过模拟粒子群智能搜索机制与遗传变异操作相结合的方式, 实现对复杂优化问题的高效近似求解. 本文将

各个基站边缘的任务迁移比例定义为搜索空间,将基站边缘的任务迁移策略定义为粒子.具体参数设置如下: $P_{num} = 100$, $N = 1000$, $R_{max} = 0.05$, $c_1 = c_2 = 2$, $\omega \in [0.3, 0.8]$.

4.1 仿真参数设置

本文实验在搭载 3.00GHz Intel (R) Core (TM) i5-9500F CPU 和 16GiB RAM 的系统上运行,使用具有经纬度信息上海基站分布数据集^[28-30]上进行模拟仿真.仿真环境基于 Python 3.11,利用 Pandas2.1 和 Numpy 1.24 建立系统模型,并使用 CVXPY 1.5 和 CPLEX 22.1 求解凸问题.实验具体的相关参数如表 2 所列.

表 2 实验参数设置
Table 2 Simulation parameters

参数	参数含义	值
M	基站数量	30
N	无人机数量	10
H	无人机飞行高度	100 m
λ_i	任务到达率	$N(10, 4)$ [21]
f_i^b	基站服务速率	$N(15, 6)$
f_i^u	无人机服务速率	$N(2, 1)$
α_0	单位信道增益	3×10^{-4}
σ^2	白高斯噪声方差	-100 dbm/Hz
B	信道带宽	1 MHz
P^c	基站的发送功率	10 W
δ	基站上传任务比例	0.1
D^c	基站任务传输时延	$[0.05, 0.2]$
ε	迭代精度控制参数	0.001

4.2 仿真结果分析

本文通过观测 30 个基站边缘中的随机一组邻接边缘集的任务负载和任务完成时间的变化情况,以此来证明所提出的 UADLBA 方法能有效地收敛到纳什均衡.图 2、图 3 显示

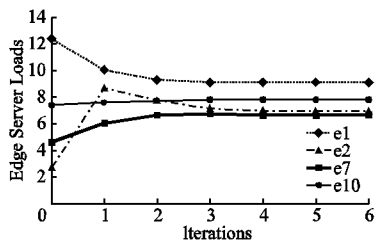


图 2 基站边缘服务器的负载变化
Fig. 2 BS edge server loads changes

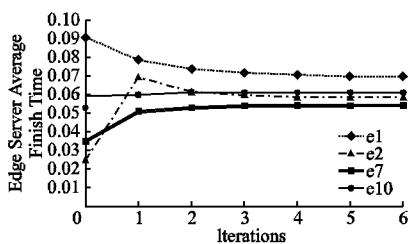


图 3 基站边缘服务器的任务完成时间
Fig. 3 BS edge server finish time

了通过观察基站边缘 e_1 及其邻接边缘 e_2, e_7, e_{10} 组成的邻接边缘集的任务负载和任务完成时间的变化情况,其中,基站边缘 e_1 接入无人机边缘 u_6 .如图 2 和图 3 所示, e_1, e_7, e_{10} 在多轮

迭代中均呈现单调递增/递减至稳定的负载值和任务完成时间,是因为初始为高负载的 e_1 将任务卸载至邻接边缘上,从而降低了负载和任务完成时间.而 e_2 的负载与任务完成时间先升高后降低,是因为 e_2 开始时处于低负载状态,其邻接边缘偏向于将任务卸载给他,从而导致负载激增, e_2 则选择将自身的初始到达任务卸载给其他邻接边缘来降低负载.同时可以发现, e_2, e_7, e_{10} 在到达稳定状态时,负载大小相近,这是因为其任务服务速率相差不大,系统达到稳定时会充分利用各个边缘的计算资源.而 e_1 在稳定状态下的的负载较高,是因为有辅助无人机边缘接入 e_1 ,从而提高了 e_1 的任务服务速率,提高了其最大负载能力.经过多轮迭代后,任务负载和任务完成时间趋于相对稳定的状态.因此, UADLBA 可以在有限的迭代次数中收敛到纳什均衡.

本文将系统平均效用值 $Q = \frac{\sum_{e_i \in \mathcal{A}} Q_i}{M}$ 作为性能指标.图

4 从 3 种场景对各个策略的性能进行比较,其中,场景 1 是由 80% 低负载率(负载率在 20% 以下)的基站边缘和 20% 高负载率(负载率在 80% 以上)的基站边缘组成的低负载基站边缘系统;场景 2 是由 50% 低负载率的基站边缘和 50% 高负载率的基站边缘组成的中等负载基站边缘系统;场景 3 是由 20% 中负载率(负载率在 50% 左右)的基站边缘和 80% 高负载率的基站边缘组成的高负载基站边缘系统.

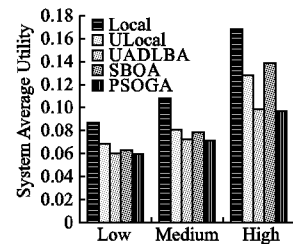


图 4 多个场景下边缘系统的平均效用值
Fig. 4 Average utility of system in multiple scenarios

结果显示,本文提出的 UADLBA 方法在 3 种场景中都能够取得较好结果.如图 4 所示, UADLBA 方法相较于本地计算策略在 3 种场景下分别提高了 30.1%, 32.8%, 41.3%;相比 SBOA 方法分别提高了 3.7%, 8.1%, 29.0%,这是因为 SBOA 法仅考虑基站之间的负载均衡,在绝大多数基站都处于高负载的极端场景中,无法将各个基站边缘的任务进行合适地卸载以减小各个边缘的负载率,所以当系统中越多边缘处于高负载状态,负载均衡策略效果越差.但是略差于 PSOGA 方法 0.9%, 1.3%, 1.5%,这是因为 PSOGA 是以执行时间换效果的方法,可以在多次迭代中找到近似最优的任务卸载策略.由上述数据可以推出,本文提出的 UADLBA 方法相较于其他策略性能更好,且在面对不同的场景,该方法也拥有较好的适应性.

接下来,本文对 UADLBA、SBOA 和 PSOGA 算法的决策时间方面进行评估.表 3 展示了这 3 种对比方法 3 种不同场景下找到负载均衡方案的平均决策时间.在 3 种不同场景下, SBOA 与 UADLBA 的平均决策时间相差不大,但是 UADLBA 略高于 SBOA 的原因是 UADLBA 方法还会考虑无人机的自适应接入策略.而 PSOGA 是一种集中式的优化方

法,从而导致执行时间过高,无法适应实时变化的多边缘环

表3 负载均衡决策时间

场景	UADLBA	SBOA	PSOGA
Low	1.43	1.31	1234.17
Medium	1.46	1.36	1336.32
High	1.48	1.37	1387.61

境.由上述数据可以推出,本文提出的 UADLBA 方法在不同的边缘计算场景均能保持较高的执行效率,适用各种不同的边缘计算场景.

5 结论

本文主要研究了如何在无人机辅助下的边缘系统中实现高效的负载均衡问题.首先,将该问题拆分为基站边缘服务器的负载均衡和无人机的接入方案两个子问题,并将基站边缘服务器的负载均衡建模为一个存在纳什均衡的非合作博弈模型.接下来,提出了 UADLBA 方法. UADLBA 方法不仅通过无人机的自适应接入策略选择最优的基站边缘服务器进行接入辅助,还能保证基站边缘服务器的负载均衡博弈收敛到纳什均衡点.最后,本文与 PSOGA 算法与 SBOA 算法进行了性能比较.仿真实验结果表明,相较于其它基准方法,本文所提出的 UADLBA 方法在执行时间和平均任务完成时延上均能取得更好的优化效果,且适用于极限高负载的边缘计算场景.在未来的工作中,将考虑更复杂的无人机接入策略以及探索更精细的边缘计算系统模型,考虑边缘设备的能耗,建立边缘能耗时延的动态评估模型.此外,我们还将考虑用户设备行为对边缘负载均衡策略的影响,通过多维度建模与优化,构建更精确反映实际物理系统特征的系统模型.

References:

[1] Porambage P, Okwuibe J, Liyanage M, et al. Survey on multi-access edge computing for internet of things realization [J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(4): 2961-2991.

[2] Li Jing, Liang Weifa, Xu Wenzheng, et al. Maximizing user service satisfaction for delay sensitive IoT applications in edge computing [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 33(5): 1199-1212.

[3] Fan X, Cao J, Mao H. A survey of mobile cloud computing [J]. *ZTE Communications*, 2011, (1): 4-8.

[4] El Sibai R, Gemayel N, Bou Abdo J, et al. A survey on access control mechanisms for cloud computing [J]. *Transactions on Emerging Telecommunications Technologies*, 2020, 31(2): e3720.

[5] Bello S A, Oyedele L O, Akinade O O, et al. Cloud computing in construction industry: use cases, benefits and challenges [J]. *Automation in Construction*, 2021, 122: 103441.

[6] Mishra A, Jain R, Durresi A. Cloud computing: networking and communication challenges [J]. *IEEE Communications Magazine*, 2012, 50(9): 24-25.

[7] Sadeeq M M, Abdulkareem N M, Zeebaree S R M, et al. IoT and cloud computing issues, challenges and opportunities: a review [J]. *Qubahan Academic Journal*, 2021, 1(2): 1-7.

[8] Patel M, Naughton B, Chan C, et al. Mobile-edge computing introductory technical white paper [R]. Sophia Antipolis: Mobile-Edge Computing Industry Initiative, 2014.

[9] Feng C, Han P, Zhang X, et al. Computation offloading in mobile edge computing networks: a survey [J]. *Journal of Network and Computer Applications*, 2022, 202: 103366.

[10] Wang S, Tuor T, Salonidis T, et al. Adaptive federated learning in

resource constrained edge computing systems [J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(6): 1205-1221.

- [11] Cui J, Liu Y, Nallanathan A. Multi-agent reinforcement learning-based resource allocation for UAV networks [J]. *IEEE Transactions on Wireless Communications*, 2019, 19(2): 729-743.
- [12] Sun G, Wang Y, Sun Z, et al. Multi-objective optimization for multi-uav-assisted mobile edge computing [J]. *IEEE Transactions on Mobile Computing*, 2024, 23(12): 14803-14820.
- [13] Peng H, Shen X. Multi-agent reinforcement learning based resource management in MEC-and UAV-assisted vehicular networks [J]. *IEEE Journal on Selected Areas in Communications*, 2020, 39(1): 131-141.
- [14] Sam George. Building an ecosystem for responsible drone use and development on Microsoft Azure [OL]. <https://azure.microsoft.com/en-ca/blog/building-an-ecosystem-for-responsible-drone-use-and-development-on-microsoft-azure/>, 2018-10-30.
- [15] 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.
- [16] 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.
- [17] Yang C, Xu X, Bilal M, et al. Deep-deterministic-policy-gradient-based task offloading with optimized k-means in edge-computing-enabled iomt cyber-physical systems [J]. *IEEE Systems Journal*, 2023, 17(4): 5195-5206.
- [18] Mattia G P, Pietrabissa A, Beraldi R. A load balancing algorithm for equalising latency across fog or edge computing nodes [J]. *IEEE Transactions on Services Computing*, 2023, 16(5): 3129-3140.
- [19] Xu Y, Mohammed T, Di Francesco M, et al. Distributed assignment with load balancing for DNN inference at the edge [J]. *IEEE Internet of Things Journal*, 2022, 10(2): 1053-1065.
- [20] Cheng N, Xu W, Shi W, et al. Air-ground integrated mobile edge networks: architecture, challenges, and opportunities [J]. *IEEE Communications Magazine*, 2018, 56(8): 26-32.
- [21] Lin B, Weng J, Chen X, et al. A game-based computation offloading with imperfect information in multi-edge environments [J]. *IEEE Transactions on Services Computing*, 2024, 18(1): 1-14.
- [22] Fan W, Yao L, Han J, et al. Game-based multitype task offloading among mobile-edge-computing-enabled base stations [J]. *IEEE Internet of Things Journal*, 2021, 8(24): 17691-17704.
- [23] 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.
- [24] Shortle J F, Thompson J M, Gross D, et al. Fundamentals of queueing theory [M]. John Wiley & Sons, 2018.
- [25] Boyd S, Vandenberghe L. *Convex optimization* [M]. Cambridge: Cambridge University Press, 2004.
- [26] Scutari G, Palomar D P, Facchinei F, et al. Monotone games for cognitive radio systems [M]. *Distributed Decision Making and Control*, 2012: 83-112.
- [27] ZHENG L H, XIAO B H, YAO Z W, et al. Graph reinforcement learning based multi-edge cooperative load balancing method [J]. *Computer Science*, 2025, 52(3): 338-348.
- [28] Li Y, Zhou A, Ma X, et al. Profit-aware edge server placement [J]. *IEEE Internet of Things Journal*, 2021, 9(1): 55-67.
- [29] 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.
- [30] 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.

附中文参考文献:

- [27] 郑龙海,肖博怀,姚泽玮,等.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.