

移动云-边环境下任务卸载多方案 研究综述



谭润提, 张龙信*, 艾明慧, 曾文亮, 陈景胜

湖南工业大学计算机学院, 湖南株洲 412007

摘要: 移动边缘计算 (Mobile Edge Computing, MEC) 作为传统云计算向网络边沿延伸的一种新兴技术, 将计算密集型任务卸载至边缘服务器以解决终端用户设备资源、算力的匮乏, 满足移动用户对应用程序任务计算的低能耗、低延迟需求, 大幅缓解云中心压力。云-边结合架构已成大势所趋。如何对终端任务执行最优卸载调度一直是 MEC 研究领域的重点问题之一。本文对任务卸载技术在当今云-边环境下的研究方向与成果进行综述。首先, 介绍了 MEC 环境与任务卸载技术的发展现状, 阐述了启发式算法、元启发式算法、强化学习等经典算法的概念及应用, 并依据所研文献的主流方案总结、归纳出任务卸载方案的两类方向——基于算法优化的传统任务卸载方案与基于强化学习的交互任务卸载方案; 而后从问题约束、优化目标、所作贡献等方面对上述两类主流方向的所研文献进行了归纳与整理, 并对其所用经典算法及创新进行相互对比; 最后, 对该领域当下所面临的挑战与未来研究目标进行了总结, 为后续研究开展提供参考与帮助。

关键词: 移动边缘计算; 任务卸载; 组合优化; 强化学习; 启发式算法

DOI: [10.57237/j.cst.2023.03.001](https://doi.org/10.57237/j.cst.2023.03.001)

Survey on Multiple Methods for Task Offloading in Mobile Cloud-Edge Environment

Tan Runti, Zhang Longxin*, Ai Minghui, Zeng Wenliang, Chen Jingsheng

College of Computer Science, Hunan University of Technology, Zhuzhou 412007, China

Abstract: Mobile edge computing (MEC) is an emerging technology that extends cloud computing to the edge of the network. It offloads computing intensive tasks to the edge server to solve the problem of insufficient computing power and resource of the terminal device, meets the low energy consumption and low latency requirements of the mobile user for application task computing, and greatly releases the pressure of the cloud center. Cloud-edge integration architecture has become a trend. How to perform optimal offload scheduling for terminal tasks has always been one of the key issue in the field of MEC research. This paper reviews the research directions and achievements of task offloading technology in today's cloud-edge environment. First, the development status of MEC environment and task offloading technology are introduced, and the concepts and applications of classic schemes such as heuristic algorithm, metaheuristic algorithm, and reinforcement learning are elaborated. Based on the mainstream solutions in the research literature, two types of task offloading solutions are summarized: traditional task offloading solutions based on algorithm optimization and

基金项目: 湖南省自然科学基金 (2023JJ50204).

*通信作者: 张龙信, longxin.zhang@163.com

收稿日期: 2023-04-09; 接受日期: 2023-05-25; 在线出版日期: 2023-06-15

<http://www.computscitech.com>

interactive task offloading solutions based on reinforcement learning. Second, the research literature on the two mainstream directions mentioned above is summarized and organized from aspects such as problem constraints, optimization objectives, and contributions made. Meanwhile, schemes are compared through the classic algorithms and innovations they used. Finally, the current challenges and future research objectives in this field are summarized to provide reference and help for the follow-up research.

Keywords: Mobile Edge Computing; Task Offloading; Combinatorial Optimization; Reinforcement Learning; Heuristic Algorithm

1 引言

近年来,以传统云计算为基础的计算平台面临着巨大的挑战。由于计算机与网络技术的高速发展与普及,物理位置处于网络边缘的终端用户密度在网络内逐渐增大,并且诸如自动驾驶、增强现实、虚拟现实等时延敏感型新兴热点在终端应用中已占据着愈发重要的地位。传统云计算服务器通常位于网络的中心,距离网络边缘处的用户距离较远。用户终端设备将产生的实时数据卸载至遥远的云中心能享受到便利的高性能计算,但随之而来的是上、下行传输过程中终端用户面临的时延、能耗、安全等问题。虽然传统云计算平台拥有强大的存储、计算能力,但网络边缘终端设备的激增、新型网络应用对时延的高敏感度,以及数据安全性与服务质量(Quality of Service, QoS)亦成为当前网络环境下用户体验的一种评价指标,仅依靠云计算模式的单一架构已难以满足当今网络的应用需求[1]。

在传统云计算模式已然捉襟见肘的处境下,研究者逐渐设想将云计算向网络边缘延伸,提出了移动边缘计算(Mobile Edge Computing, MEC)[2]。在MEC中,终端用户所卸载的任务不再直接传输至云计算中心,而是卸载至物理架构位于网络边缘的小型MEC服务器进行计算,进而保证任务卸载的低时延、高反馈,更好地满足用户的服务体验。加特纳集团将MEC定义为“分布式计算模型中数据存储、处理位于终端用户生成或使用数据的这一边缘区域的组成单元”[3]。有效的卸载策略能够降低应用的响应时间和终端设备能耗,更好地满足用户的QoS需求[4]。

MEC作为云计算中心的拓展与延伸,其目的并不是取代云计算[5]。云-边-端三层架构作为目前MEC经典架构之一,依据卸载要求的不同对云、边、端处理任务类型进行了划分,云计算主要负责处理计算量大、运行周期长的任务,而边缘计算更适用于对时延、响

应速度更为敏感的任务,边缘服务器与云中心形成了一种互补协同。通过将边缘服务器引入传统云-端架构,网络中用户设备与云计算中心交互的频率得到缓解,极大地释放了传统云计算架构的压力[6]。

在众多研究综述中,施魏松教授团队[7]围绕边缘计算的发展历程、研究现状与未来展望三个方面进行综述,首先整理了近几十年以来边缘计算从提出探讨到平稳发展的历程;而后总结了网络、隔离技术、体系结构、操作系统、算法框架、数据处理与安全七项支持边缘计算发展的技术,并介绍了现有边缘计算的六类应用;最后对边缘计算正面临的紧迫问题进行了归纳。Jiang等人[8]对任务卸载技术展开综述,介绍了任务卸载技术、QoS、体验质量(Quality of Experience, QoE)等方面的基本现状,并从合作与非合作博弈论应用与传统启发式任务卸载两个方面对任务卸载技术方案进行分类综述。Wei等人[9]介绍了MEC与强化学习的深度结合,首先分析了MEC中的网络不确定性问题,进一步阐述了保证高QoS所需解决的挑战;而后为应对这种挑战及其解决方案,从目标、约束、技术等方面详实总结了现如今学术界、工业界将强化学习等智能方案与MEC相结合的研究进展。Shakarami等人[10]从人工智能的角度出发,将任务卸载方案分为强化学习、有监督学习与无监督学习三种情况进行综述,阐述了基于人工智能的任务卸载方案相较于传统启发式优化算法所具有的优势。

综上,本文组织如下。第二节介绍移动边缘计算的基本概念与架构;第三节介绍任务卸载技术的基本概念与方法;第四节介绍基于算法优化的传统任务卸载方案的研究综述;第五节介绍基于强化学习的交互任务卸载方案的研究综述;第六章介绍在移动边缘计算环境下任务卸载问题面临的挑战,并对其未来研究方向进行展望;第七章对全文工作进行了总结。

2 移动边缘计算

2.1 MEC 基本概念

边缘计算 (Edge Computing, EC) [11]这一概念最早可见于内容分发网络 (Content Delivery Network, CDN)。CDN 由 Akamai Company 提出,是一种网络内容服务体系,其基于 IP 网络而构建,位于网络中更接近用户的边缘地带,以缓存的方式实现图像、视频等内容的高速传输。CDN 能够实时地根据网络流量和节点的连接、负载状况等综合信息将用户的访问指向距离最近的缓存服务器上,进而解决网络拥挤带来的低质量用户体验[12]。

2009 年, Satyanarayanan 等人[13]提出了一种基于云的、资源富集的移动计算架构 Cloudlet。Cloudlet 同样位于网络的边缘地带,是一种受信任的、计算能力优秀的中小型计算机或集群,能简易地为用户提供云中心的服务,并且由于其与用户仅一跳的网络延迟,更易于实现清晰的交互响应并便利的计算服务。

时至今日,“边缘”这一概念在“万物互联背景下随之产生的大量边缘数据何去何从”这一问题下愈发受到行业重视,移动边缘计算、雾计算、海云计算等云计算模式的创新随之得到加速发展,收获了国内外专业学会的认可与期待。

MEC 的设计目的是部署具有一定计算能力的节点在网络边沿,其作为一种云计算中心的延伸与拓展,边缘节点能在数据源附近进行计算,为终端用户提供低延迟的服务并支持移动性接入,此外还具有持续地控制、管理终端用户任务卸载信息的功能[14]。MEC 网络中的边缘服务器将云计算模式延伸到大型网络中用户密集的边缘地带,进而弥补了云计算中数据集成存储的安全性问题以及服务响应的高延迟问题[15]。在表 1 中,从模型架构、硬件大小、计算资源位置等方面对云计算与 MEC 进行了对比。

表 1 MEC 与云计算的对比

	MEC	云计算
模型架构	分布式管理	集中式管理
硬件大小	中体量	大体量
资源位置	网络边沿	网络中心
通信网络	WLAN、5G	广域网
网络时延	低	高
传输能耗	少	多
计算能力	适中	强
适宜应用	延迟敏感型	计算密集型

由表 1 可知,与云计算模式下以云为中心辐射式构成的集中式计算网络不同,MEC 采用分布式管理的方式在一个网络中部署了大量的中小型服务器为用户提供服务。在计算资源方面,相较于距离终端用户遥远的云计算具有着大体量的计算资源,MEC 中的边缘服务器虽然拥有的轻量的计算资源,但与终端用户的距离更近,进而用中等强度的计算能力交换而来的是更低的网络时延与更少的传输能耗,这十分适合解决来自终端用户的计算量要求适中但时延极其敏感的应用,例如自动驾驶的实时路况、车况调整以及物联网设备的即时控制。

2.2 MEC 基本架构

MEC 基本架构根据具体的部署方案分为三种:云-边-端协同、云-边协同、边-端协同。其中,依据网络部署的最优分配策略综合考虑,架构通常建模为三层,即云-边-端三层结构,分别为云中心层、边缘层与终端层,如图 1 所示。

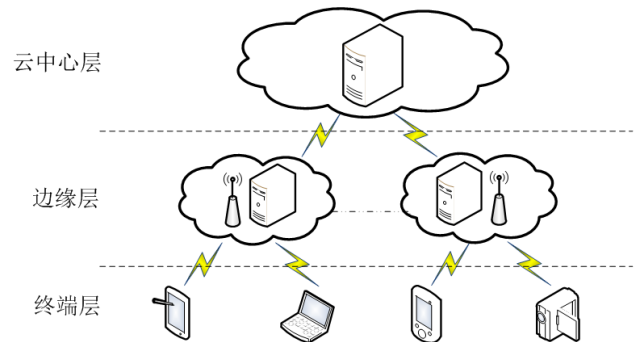


图 1 MEC 基本架构

在图 1 中,云计算数据中心通常具有超大的资源规模,其强大的计算能力能在很短的时间内解决终端用户卸载而来的高计算量任务,在终端设备与边缘服务器难以有效地处理任务时,可以请求云中心协助。但由于云中心位于整个网络的中心位置,与终端用户的距离较远,进而在高速任务计算的同时会带来高额的传输能耗与时延,更适合于时延容忍型应用与网络内大型任务。

在边缘层中,边缘服务器具有着比终端设备更强的计算能力,并且与终端用户的距离较近,是解决单一云中心体系高延迟、高负荷问题的有效方案。边缘服务器在控制中心的调度下能够为其基站覆盖范围内的终端用户提供低时延、低能耗的计算服务,并且将

无法高效处理的大型任务转送到云中心处理，充当云中心层与终端层的桥梁。

在终端层，终端设备包括用户的智能设备、道路智能摄像头、智能车辆等具有基础计算能力的小型计算单元。由于网络中存在着大量的空闲终端设备，通过对其进行整合亦可以提供巨大的计算价值与资源，为本地用户或其他用户提供服务。当本设备资源空闲时，可以提供计算服务给发出任务请求的其他设备；同样地，当本设备需要卸载任务时，可以作为服务的请求方，将任务卸载至其他拥有空闲计算资源的终端设备，特别是物联网公共终端设备。

3 任务卸载

3.1 任务卸载基本概念

“任务卸载”这一概念在很多研究中亦称为“计算卸载”[16]，最早在云计算领域中被提出。任务卸载是指利用有效的任务卸载与资源分配方案将用户终端上亟待卸载的任务卸载到边缘服务器，利用边缘服务器较为丰富的计算能力完成被卸载任务的计算工作，减少终端设备能耗的同时由于其临近的地理位置进一步缩短任务完成时延，提高整个系统的任务吞吐性能[17]。若通过综合考量得出某任务不适合卸载至边缘服务器执行，该任务将进一步被边缘服务器卸载至资源、算力更庞大的云计算中心进行处理。任务卸载技术是移动边缘计算领域的重点研究问题之一，亦是改善用户体验、提升系统效率的关键[18]。

关于任务卸载方案的评价，其核心是对任务卸载指标进行评价，终端设备与服务器设备的计算能力不同，任务是否卸载、卸载至何处需要一个用于评价的指标与参考要素[19]。任务卸载参考要素包含有以下五种：

(1) 资源量：边缘服务器的计算资源与存储资源是任务卸载决策制定的重要参数之一。当边缘服务器的计算资源、存储资源已占用比例过高或本身难以满足被卸载任务的要求时，该任务需要排队或进一步卸载至云中心，高效的卸载决策是资源良好循环的关键。

(2) 能量损耗：云计算中心、边缘服务器以及设备终端的卸载能耗是任务卸载问题最重要的评价指标之一，其主要分为传输能耗与计算能耗。由于云计算中心通常拥有强大的支撑资源，故通常研究中只考虑边缘服务器与用户终端设备在任务卸载中的能量消耗。而又由于此二类设备的能量较为有限，故在一定的能

耗约束下优化任务卸载决策是当前研究现状下的主要方向之一。

(3) 任务时限：任务卸载的时延约束自异构云环境下的任务调度问题以来就是此类问题的重要评价指标，其主要分为传输时延与计算时延。为了提高任务卸载的时延敏感度，被卸载任务通常会具有截止日期属性充当限制约束，在一定的时延约束下优化任务卸载决策是当前研究现状下的主要方向之一。

(4) 任务大小：任务的属性通常具有任务传输数据量与每比特数据量所需 CPU 计算周期两个属性，进而影响任务卸载的总体能耗与时延。

(5) 任务可拆分情况：“任务是否可拆分”决定了任务卸载的方案，通常分为两种方案——“0-1”卸载与部分卸载[20]。“0-1”卸载是指任务只能作为一个整体，要么完整地在本地设备执行，要么完整地卸载至边缘服务器或云中心执行，该方案整体复杂度高，但由于只考虑整体卸载策略，故处理代价较小。部分卸载是指将任务划分为多个子任务进行并行化处理，使其部分在本地执行，部分进行卸载，从而降低任务完成时间与能耗，算法因此更为复杂。

在对卸载指标的综合考量与分析之下，任务卸载的优化目标通常分为能耗最小化、时延最小化以及能耗时延均衡最小化三种。

本章后文将对作为基于算法优化的传统任务卸载方案核心的启发式算法与作为基于强化学习的交互任务卸载方案核心的强化学习进行介绍。

3.2 启发式算法

启发式算法是指一个基于直观或经验构造的算法，在可接受的花费（通常指资源、时延与能耗）下给出待解决组合优化问题的每一个实例的一个可行解，该解一般逼近于最优解[21]。

在启发式算法的经典应用中，Haluk Topcuoglu 等人[22]提出了一种异构最早完成时间（Heterogeneous Earliest-Finish-Time, HEFT）算法和处理器上关键路径（Critical-Path-on-a-Processor, CPOP）算法以解决异构计算下时延最小化问题。HEFT 算法在每一步中选择具有最高向上排序值的任务，并将所选任务分配给处理器，并通过基于插入的方法使其最早完成时间最小化。CPOP 算法使用向上和向下排序值的总和来区分任务的优先级，且在处理器选择阶段将有向无环图（Directed Acyclic Graph, DAG）各层耗时最长的任务

组成一条“关键路径”调度到计算能力最强的处理器上，以最小化系统模型的总执行时间。后续实验仿真证明 HEFT 算法与 CPOP 算法较此前算法更优，并时至今日仍作为一种调度思想指导于 DAG 研究工作中。

为了解决启发式算法复用效率低下的弊端，研究工作逐渐深入并产生了一种基于启发式思想的新方向——元启发式算法。该算法可以广泛地应用到组合优化问题与高纬度非凸优化问题中，故亦称其为通用启发式算法。元启发式算法以自然界模仿思想为主，大致分为基于大群智慧的优化算法与基于自然规律算法。基于大群智慧的优化算法通过观察、分析诸如蚂蚁、蜜蜂、群栖鸟类或鱼群等自然界大型社会性群体生物的生存规律与原理，进而构建数学模型，逼近所研问题的最优解，主要包括蚁群算法[21]（Ant Colony Optimization, ACO）、蝙蝠算法[23]（Bat Algorithm, BA）、乌鸦搜索算法[24]（Crow Search Algorithm, CSA）、粒子群优化算法[25]（Partical Swarm Optimization, PSO）等。基于自然规律的启发式算法通过模拟自然界生物的进化过程进而构建任务模型实现最优化，例如遗传算法[26]（Genetic Algorithm, GA）。

在元启发式算法与异构云环境下 workflow 任务调度相结合的应用背景下，Zhang 等人[27]为解决预算成本约束下的 workflow 调度能耗最小化问题，基于传统鲸鱼群优化算法进行改进，提出了一种能量最小化鲸鱼优化算法（Energy Minimization Whale Optimization Algorithm, EM_WOA）。传统算法是通过模拟鲸鱼群逼近鱼群、盘旋定位、随机探索等行为进行捕食而构建的一种模型。EM_WOA 在原算法基础上将逆向学习方法应用至鲸鱼种群的初始化阶段，进而提高精英个体在初始种群中的占比，并利用随机频率因子避免在搜索过程中受困于区域最优解。此外，作者设计了一种服务实例选择策略用以在具有特定预算成本的工作流应用程序中选择合适的服务器。仿真实验表明，EM_WOA 在能耗节约与调度成功率方面相比于 PSO 算法、原始鲸鱼优化算法具有更明显的性能优势。

3.3 强化学习

强化学习（Reinforcement Learning, RL）讨论的问题是智能体如何在复杂的环境里最大化自己的总奖励收益[28]。智能体以“试错”的方式持续与环境交互，通过观察环境的当前状态后在环境采取一个动作以获得一个奖励并进入下一状态，根据这个正面奖励或负面奖

励的结果以供其后续行为判断，最终目标是使智能体在结束时获得当前环境下的最大收益，如图 2 所示。“迭代式累计奖励”即是“强化”的概念过程。

马尔可夫决策过程（Markov Decision Processes, MDP）是 RL 研究领域最常用的数学模型，用于在系统状态具有马尔可夫性质的环境中模拟智能体可实现的随机性策略与回报[29]。MDP 是强化学习问题的数字化理论形式，在强化学习领域具有重要的描述与建模作用。

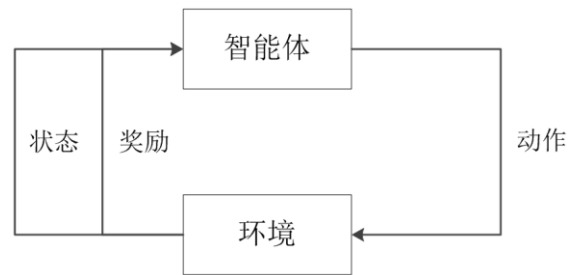


图 2 智能体与环境的交互

MDP 将强化学习中智能体与环境的交互过程描述为一个由状态、动作、状态转移概率矩阵、折扣因子与奖励组成的五元组。其中，状态转移概率矩阵用于描述从某个状态进入到下一个状态的总概率集合，通常用矩阵表示，即矩阵的每一行描述的是一个节点到所有其他节点的概率。折扣因子用于对后续状态的价值总和进行收敛与控制，折扣因子的值越大，未来奖励对当前奖励的影响越大，该策略越注重长期收益的价值。进而，一个抽象的智能体与环境的交互过程被转化成了可供加工的数字语言。

ϵ -greedy 算法是强化学习的一种探索机制，是为了智能体能够进行有意义探索的有力保障，即拒绝固守已有的经验而不进行新的尝试以获取新的状态价值。 ϵ -greedy 算法会通过设置一个 0 到 1 之间的 ϵ 值，在每一轮迭代选取下一步动作前进行随机数生成，若随机数小于 ϵ 值，则选取下一步动作集合中任意一个动作，即是探索的过程。通过设置 ϵ 值的大小，可以控制智能体的训练注重于长期奖励或即时奖励。

但是，现实生活场景模型中的状态和动作往往更加复杂多变，传统强化学习的训练能力与经验空间难以支持智能体符合目标预期。2015 年，DeepMind 团队将 Q-Learning 算法与深度神经网络相结合，提出了深度强化学习（Deep Reinforcement Learning, DRL）重要理论基础——深度 Q 学习[30]，也称为深度 Q 网络（Deep Q-Learning Network, DQN）算法。并随后衍生出了基

于价值 (Value-based) 的 DoubleDQN、DuelingDQN 等以及基于策略 (Policy-based) 的 Actor-Critic、A3C、PPO、DDPG 等强化学习算法。

传统 RL 通常利用价值表 (Q-Table) 存储每次迭代中智能体与环境交互所产生的各种参数, 以供未来选择某一动作时进行评价。DQN 将传统 RL 与深度学习相结合, 通过神经网络拟合奖励值替代 Q-Table,

以当前状态为输入值得到当前各动作对应的价值进而提高训练效率, 并且利用参数更新滞后的目标网络、经验回放机制保证稳定性与独立性。

本文将对近年来移动边缘计算领域下任务卸载技术研究的部分现有文献进行总结与归纳, 并在表 2 中针对所研文献的优化目标、所运用的技术以及文献的主要研究贡献进行详细列举与直观对比。

表 2 MEC 环境下的任务卸载方案总结

文献	所用技术	优化目标	研究贡献
Keqin Li.	启发式算法	能耗约束下时延最小化	将云端融合的任务卸载视为组合优化问题, 提出了多个启发式算法并证明了一个渐进性能界, 此外推导了其最优解下界。
Keqin Li.	启发式算法	能耗约束下时延最小化	在组合优化的框架下, 从任务数据、通信大小位置的情况出发, 提出了一种离线非透视任务卸载算法与随机在线非透视任务卸载算法, 并给出了高概率性能保证的分析结果。
Cheng et al.	启发式算法	资源、时延约束下能耗最小化	在数据共享的移动边缘计算环境下, 对任务是否可分离、是否有依赖的各情况进行了论证与分析, 并各提出了启发式算法进行求解。
Guo et al.	启发式算法	时延、数据依赖约束下能耗最小化	提出了一种在线混合动态调度算法, 将多个关联子任务合并成单个任务, 采用工作流混合调度来减少虚拟机空闲时间, 并通过动态迁移减少损失。
Saleem et al.	遗传算法	能耗约束下时延最小化	提出了一种基于遗传算法的移动感知任务调度算法, 并通过启发式算法的方式对总任务执行时延实现了优化。
Yao et al.	拉格朗日乘法, 遗传算法	时延约束下能耗最小化	提出了一种基于随机模拟的两阶段随机规划算法, 通过拉格朗日乘法与遗传算法获得本地与边缘的资源最优分配策略。
Chen et al.	粒子群优化算法	截止日期约束下能耗最小化	设计了一种考虑周全的智能物联网能耗模型, 提出了一种基于 GA 的粒子群优化算法卸载策略。
Chen et al.	博弈论	资源约束下能耗最小化	将多用户分布式任务卸载问题表述为博弈游戏, 证明了其具有纳什平衡, 提出了一种基于博弈论的分布式任务卸载算法并推广至多信道无线竞争环境。
Chen et al.	博弈论	均衡时延、能耗最小化	将 MEC 任务卸载决策问题建模为一个非合作博弈模型, 提出了一种动态非合作博弈算法。
Liu et al.	博弈论, 元启发式算法	均衡时延、能耗最小化	将超密集网络下任务卸载问题分解为任务卸载目标选择与任务比例切割两个子问题, 提出了一种基于非合作博弈论的卸载目标选择算法与一种基于萤火虫优化的卸载比例算法。
Yang et al.	拉格朗日乘法, 博弈论	资源、能耗约束下时延最小化	提出了一种基于博弈论的协同卸载策略, 利用拉格朗日乘法求解边缘服务器上的资源分配。
Sadiki et al.	DQN, PPO	均衡时延、能耗最小化	结合 DQN 与函数逼近提出了基于深度 Q 网络的能耗最小化策略, 并由于 DQN 的离散性, 提出了 Police-based 近端策略优化的能耗最小化策略。
Chen et al.	MADDPG	资源、时延约束下能耗最小化	为处理 AR 应用子任务相互依赖性与混合多用户合作与竞争问题, 提出了一种基于多智能体深度确定性策略梯度 (MADDPG) 的高效资源分配和任务卸载算法。
Liu et al.	DoubleDQN	能耗约束下时延、QoS 最优化	将问题建模为高斯-马尔科夫随机模型, 提出了一种注重服务质量的 ϵ -贪心算法并基于此提出一种基于 QoS 的动作选择策略。
Gao et al.	Q-Learning, 启发式算法	均衡时延、能耗最小化	提出了一种基于 Q-Learning 的任务卸载算法, 建立了一个状态损失函数最小化问题并通过拟凸二分法与多项式分析方法进行优化。
Huang et al.	DRL	能耗约束下时延最小化	提出了一种基于 DRL 的在线任务卸载模型, 通过任务分解避免物理约束, 并提出了一种保序动作生成方法以便高效生成动作。
Tang et al.	LSTM, DuelingDQN	均衡时延、能耗最小化	当用户卸载情况未知时引入长短期记忆网络预测边缘节点未来负载水平, 并结合 DoubleDQN、DuelingDQN 以改进长期成本估计。
Tan et al.	DQN, 蚁群算法	资源约束下能耗最小化	提出了一种基于蚁群系统与 DQN 的联合迭代算法, 并利用启发式算法优化协作决策和卸载决策。
Qian et al.	FL, DQN	时延最小化并保障隐私安全	提出了一种基于 FL 的隐私感知服务放置算法, 根据用户的喜好最大限度地满足需求的服务数量。

4 基于算法优化的传统任务卸载方案

本章节将从三种经典方案——传统启发式算法的理论创新到元启发式算法、博弈论理论, 综述移动边缘计算领域中传统任务卸载方案的研究现状, 并进行归纳与比较。

4.1 传统启发式算法研究

从基础理论的角度着手, Li [31]针对任务卸载领域中调度长度上、下界问题, 研究了设备-边缘-云融合的任务卸载与设备-边缘-云融合的能耗约束下任务卸载的组合优化问题。作者在设备-边缘-云融合的任务卸载模型下设计了三种传统启发式算法, 并通过证明得到了一个渐进性能界; 在设备-边缘-云融合的能耗约束下任务卸载问题中, 作者以交叉方式求解任务调度与功率分配两个子问题, 提出了一种启发式算法。此外, 作者在二者中都通过代数推导得到了两个问题的上、下界, 并利用其对文中算法进行优化。通过实验对比, 文章所提算法非常接近理论最优算法的性能。

此外, 对于实际环境的单位常存在不可透视性, Li [32]在组合优化的框架下, 考虑 MEC 环境中随机任务的非透视卸载问题。几乎所有的现有研究都从透视角度考虑任务卸载问题, 假设单个任务的数据、通信大小已知, 或任务集合内的任务皆可靠。针对离线非透视任务卸载问题, 提出了一种非透视任务卸载算法, 该算法能够在任务计算量和通信量未知的情况下确定任务卸载策略。针对在线非透视任务卸载问题, 提出了一种随机在线任务卸载算法, 该算法能够在任务到来未知的情况下对到达任务进行卸载决策, 并给出了高概率性能保证的分析结果。

数据共享在确认最优任务分配中的数据分布问题中往往具有较高的重要性。Cheng 等人[33]从该背景入手, 首先根据任务是否具有依赖性、是否可拆分将数据共享 MEC 环境下的任务卸载分成四类。而后, 对于无序不可分任务, 作者提出了求解不同情况的线性规划算法与贪婪启发式算法并证明其为 NP-complete; 对于有序不可分任务, 作者基于任务间依赖重新建模, 在时延与资源约束下提出了一种基于分簇的启发式算法以最小化能耗; 此外, 作者通过两种近似算法对 NP-hard 的可拆分任务进行了处理。实验证明, 作者对

于三种情况下的处理皆取得了很好的性能优势。

同样是在时延约束下通过启发式算法研究实现最小化能耗问题, Guo 等人[34]提出了一种在线混合动态调度算法(Online Hybrid Dynamic Scheduling algorithm, OHDS), 并最大化服务资源利用率。该算法由六个子算法组成, 其具有以下特点: (1) 通过将多个有关联的工作流任务合并成单个任务, 进而减少在任务调度过程中引起的数据传输开销; (2) 本算法根据任务的最早完成时间为每个任务分配调度优先级, 并根据任务的最晚完成时间为工作流任务分配初始子截止期; (3) 采用工作流混合调度来减少虚拟机的空闲时间。根据当前调度信息动态调整未分配任务的直接子任务优先级和子截止期; (4) 动态地迁移负载服务上的 VM 以减少服务能量损失。基于三种不同类型的工作流数据集, 作者将所提算法与三种 HEFT 改进算法进行实验对比, 结果表明 OHDS 算法在执行能耗、调度成功率和服务资源利用率等方面具有更好的性能。

4.2 元启发式算法研究

为了解决终端用户的移动性对边缘环境下动态卸载决策的复杂性, Saleem 等人[35]引入元启发式算法解决离散组合优化问题, 在终端设备的能耗约束下以最小化任务时延为目标, 提出了一种结合遗传算法的移动预测可知任务调度算法。作者首先使用遗传算法处理任务卸载模型下的混合整数非线性规划问题, 而后通过启发式算法进行优化以获取低时延的任务卸载策略。在仿真部分, 该算法的性能在现实人类移动轨迹下的广泛评价得到了较好发挥, 并实现了比其他对比算法更好的时延表现。

姚枝秀等人[36]在时延约束下以能耗最小化为目标, 提出了一种基于两阶段随机计划的随机仿真(Stochastic Simulation based two-stage Stochastic Programming, SS_2SSP)任务卸载和资源分配算法。作者通过 Lagrange Multiplier 与 GA 分别获得本地与边缘服务器的资源最优分配策略, 进而对比分析终端本地处理与边缘服务器处理的时延和能耗综合获取最优任务卸载方案。通过仿真实验, 结果表明 SS_2SSP 算法在系统能耗与收敛性方面皆优于对比算法。

Chen 等人[37]同样以最小化能耗为目标, 提出了一种融合遗传算法的自适应机制粒子群优化节能卸载策略。该策略在截止日期约束角度下, 通过将整个深

度神经网络以层划分策略分割为多个单一层部署单元，而后作者对粒子群优化算法与遗传算法进行结合，通过优化二者的问题编码、粒子更新策略、适应度函数并根据粒子适应度对各部署单元在云-边-端具体位置进行卸载，旨在将一个深度神经网络最优地映射在一个网络中的不同服务器中，实现大型神经网络的分布式部署化。通过仿真实验，结果表明该算法在系统能耗、不同工作量、不同系统资源下的表现皆优于对比算法。

4.3 合作与非合作博弈研究

非合作博弈强调在策略环境中各单位进行独立自主的决策，因而适宜解决二进制整数规划问题。Chen 等人[38]在资源约束下以能耗最小化为目标，针对多用户的 MEC 场景下的任务卸载问题展开了研究。作者通过分析判断卸载目标选择是一种二进制的整数约束，进而提出了一种基于非合作博弈论的任务卸载方案，并进一步将研究扩展到多通信链路的竞争环境中的任务卸载场景。仿真结果表明，该算法在能耗方面具有较大优势，并且在诸多环境中皆拥有较好的表现。

Chen 等人[39]考虑到每个终端设备的自私性和竞争性，将 MEC 任务卸载决策问题建模为一个非合作博弈模型，在截止日期、资源约束下以最小化每个终端的能耗与时延为目标，提出了一种移动多媒体通信系统计算分流的动态非合作博弈算法（Quality non-Cooperative Game based on Dynamic Game, QCOG-DG）和静态非合作博弈算法（Quality non-Cooperative Game based on Static Game, QCOG-SG）。仿真结果表明，QCOG-DG 算法能够在不同 UE 规模的 MEC 场景中有效地找到纳什平衡。

但非合作博弈往往没能考虑到整个系统的整体最优。刘振鹏等人[40]以最小化用户平均时延与总能耗为目标来解决在超密集网络环境下边缘计算环境下的任务卸载优化问题，将任务卸载过程分为任务卸载目标选择与任务比例切割两个子问题。在卸载目标问题中，作者提出了一种基于博弈论的卸载目标选择（Selecting Unloading Objects Algorithm based on Game Theory, SUO-GA）算法，定义了偏好度这一指标，在用户个人的偏好度最优与整个系统的总体偏好度最优的综合博弈下达到纳什平衡。实验对比表明，SUO-GA 算法与对比算法相比较，时延与能耗均拥有更好表现。

同样是为了解决超密集网络环境下的任务卸载问题，Yang 等人[41]在能耗约束与资源约束下以时延最

小化为目标，将任务卸载问题分为边缘服务器上资源分配问题与用户的卸载策略问题。作者假设在最优卸载方案下，利用拉格朗日乘法对边缘服务器的资源分配方案进行求解。针对卸载策略问题，作者提出了一种基于合作性博弈论的任务卸载策略（Based on Game Theory Collaborative Offloading Strategy, GT-COS），利用博弈论的思想在实现个人任务时延最小化的同时实现策略集总体时延最小化。通过实验仿真表明，GT-COS 在不同用户数、任务大小下皆优于对比算法。

综上所述，目前在移动边缘计算中的传统任务卸载方案通常在确定研究方法后，以最小化某项评价指标（时延、能耗或二者的均衡）为目标，而后考虑相关约束条件，根据子问题设计算法，最终通过仿真实验验证算法有效性。上述文献提出了新颖、优秀的传统任务卸载方案，在各自构建的场景下都解决了优化问题，但是对于大规模多用户多 MEC 服务器系统的任务卸载优化问题，仅通过算法的优化来制定任务卸载决策将事倍功半。

5 基于强化学习的交互任务卸载方案

上一章介绍了基于启发式算法的传统任务卸载方案，本章将开始介绍基于价值（Value-based）与基于策略（Policy-based）的强化学习的交互任务卸载方案。“交互”并不指代传统的人机交互，而是通过利用智能体与“环境”进行交互，从而在大规模 MEC 网络架构中通过对用户的历史卸载策略与卸载内容进行学习并试错，训练出表现更好的近似最优模型。

5.1 基于策略的强化学习方案

在 Policy-based RL 的应用中，Sadiki 等人[42]为实现在多输入多输出的大规模边缘计算中的终端设备能耗与时延最小化，首先将大规模天线技术下的 MEC 任务卸载问题以及无线信道变化、建模为马尔科夫决策模型，而后分别设计了基于深度 Q 网络的能耗最小化（Deep Q Network strategy for Cost Minimization, DQN-CM）策略与基于近端策略优化的能耗最小化（Proximal Policy Optimization strategy for Cost Minimization, PPO-CM）策略。仿真结果表明，文章所提算法优于对比算法，并且基于策略的强化学习方

案 PPO-CM 相比于基于价值的强化学习方案 DQN-CM, 具有更高的稳定性与更高效的卸载效率。

同样地, Chen 等人[43]为解决 AR 应用子任务相互依赖性与动态通信环境资源、时延约束下的能耗最小化问题, 提出一种基于多智能体深度确定性策略梯度 (Multi-Agent Deep Deterministic Policy Gradient, MADDPG) 的深度强化学习框架的智能高效资源分配与任务卸载算法以处理混合的多用户竞争与合作。实验部分从收敛性、回报率 and 能耗等方面进行了仿真模拟。结果表明, 该算法与单一智能体策略梯度算法、多智能体随机学习相比, 在降低用户能耗方面具有较好的性能

5.2 基于价值的强化学习方案

不同于基于策略的强化学习方案通过持续更新神经网络参数以获取更佳的策略来选取动作、计算回报, 基于价值的强化学习方案通常维护着一个蕴含状态、动作与对应价值的映射关系的表格或函数, 通过表格项或函数近似进而逼近最优解

在 Value-based RL 的应用中, Liu 等人[44]为了在由无人机组成的临时移动边缘计算环境的能耗约束下实现任务时延与无人机 QoS 最优化问题, 在双深度 Q 网络 (Double Deep Q Network, DoubleDQN) 的基础上提出了一种基于 QoS 的动作选择策略 (QoS-based Action Selection Policy, QoS-ASP)。作者首先提出的一种注重服务质量的 ϵ -贪心算法对 DoubleDQN 及其动作选择过程进行改进。其区别是算法在所有无人机都满足 QoS 约束时采取常规 ϵ -贪心算法, 而在不满足约束的情况时, 算法先收集不满足约束的无人机信息为一个集合, 而后抽取动作, 若该动作与集合中某无人机有关联, 则执行, 否则重新选取。仿真结果表明, QoS-ASP 在收敛性和吞吐率方面均优于传统 RL 与 DQN 算法, 且能使无人机 QoS 保证率达到 99%。

Gao 等人[45]为实现移动边缘计算环境下的最优资源分配与任务卸载方案, 在一种云-边-端协同计算系统中提出了一种基于 Q-Learning 的计算卸载 (Q-Learning based Computation Offloading Function, QLCOF) 算法, 从全局的角度预先安排单一应用链中的任务调度来实现最优化。作者首先将决策卸载过程建模为一个马尔可夫决策过程, 并设计了一种状态损失函数 (State Loss Function, STLF) 来衡量其服务质量, 进而建立了一个 STLF 最小化问题, 分别通过拟凸二分

法与多项式分析方法优化边缘云的传输功率和计算频率。实验结果表明, QLCOF 算法在不同参数下均能降低系统的 STLF, 并且在其他算法的横向对比下, 皆得到最佳表现, 但对于多用户多边缘服务器情况下的任务卸载还未有涉及。

为了解决无线供电环境的 MEC 网络在能耗约束下实现最佳的任务卸载决策与 workflow 延迟最小化, Huang 等人[46]提出了一种基于 DRL 的在线卸载框架 (Deep Reinforcement Learning-based Online Offloading, DROO)。DROO 从过去各种无线衰落条件下的卸载经验中学习, 将原始优化问题分解为卸载决策子问题和资源分配子问题, 从而避免了物理约束。此外, 作者设计了一种新颖的保序动作生成方法以有效地生成卸载动作, 并提出了一种自适应算法实时调整算法参数。通过仿真实验证明, DROO 在收敛性能、计算速率、更新间隔影响的表现优于对比算法。

5.3 多方融合的强化学习方案

神经网络作为 DRL 中的重要部分, 这方面的创新亦影响着模型的训练。Tang 等人[47]考虑不可分割和延迟敏感的任务以及边缘负载动态未知性, 并最小化期望的长期成本。为了提高算法中对长期代价的估计, 作者引入了长短期记忆 (LSTM)、DuelingDQN 与 DoubleDQN。LSTM 是循环神经网络的优化变体, 其擅于观察时间序列中各个观测值之间的依赖性, 能够用于揭示文章模型中时隙间边缘节点负载水平的变化, 进而预测未来时隙。DuelingDQN 改变了神经网络输出层结构, 其分别对优势函数与状态价值进行运算, 而后再结合成动作价值, 结构如图 3 所示。仿真结果表明, 与现有算法相比, 该算法的任务丢弃率和平均任务延迟分别降低了 86.4%-95.4% 和 18.0%-30.1%。

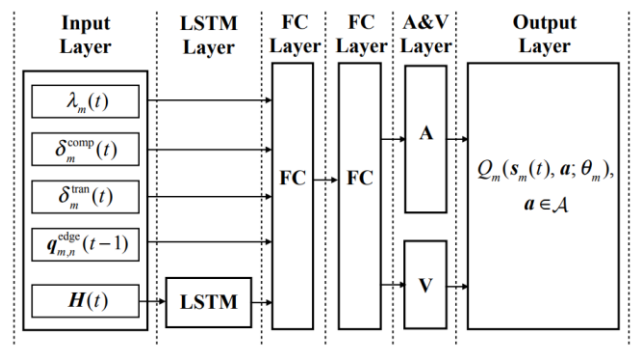


图 3 融合 LSTM 与 DuelingDQN 的三层感知机

Tan 等人[48]为了解决 MEC 网络中的多用户协同卸载和资源分配的问题并实现能耗最小化,提出了一种基于蚁群系统(Ant Colony System, ACS)与 DQN 的联合迭代算法(Joint Iterative Algorithm, JIA)。多用户协同计算即任务除了能在本地计算或者边缘节点计算外,还能传输给其他终端协同计算。JIA 中,模型上层使用启发式算法来处理已设置的协作决策和卸载决策;在底层中,通过基于当前卸载决策的深度强化学习来更新功率、子载波和计算资源的分配。通过实验结果表明, JIA 在不同网络参数设置下与参考算法相比,能耗与任务完成率的表现皆更为优秀。

由于各种用于模型训练、参数调整的用户信息(年龄或位置)皆对模型能够进行有偏好地训练,进而提高边缘服务器任务卸载的效率。但是关于用户隐私的利用涉及到严重的安全问题,而联邦学习(Federal Learning, FL)是该方向上的一种解决办法。Qian 等人[49]在资源约束下为保障隐私安全并最小化时延,提出了一种基于 FL 的隐私感知服务放置算法(Privacy-aware Service Placement, PSP)。该算法基于 FL 的理论将模型在用户本地利用用户数据进行训练,利用集中式贪心算法求解优化问题直至满足资源约束。仿真实验表明,与其他优化方案相比, PSP 算法不仅可以保护用户隐私,而且可以通过移动边缘云满足用户的服务需求。

6 挑战与展望

近年来, MEC 环境下的任务卸载技术得到了强烈且广泛的研究与关注,目前的应用模型与算法设计亦取得了一定程度的成功。然而,随着研究的持续深入,该领域某些亟待解决的挑战及其研究趋势与解决方案的重要性愈发突出,例如:通过传统单一算法优化以解决任务卸载问题的研究并不适合在长期收益、高维度、多元环境下应用;过往研究常假设任务间是相互独立的原子个体,然而在分布式计算系统中,任务间往往存在一定的关联依赖性;“移动”是 MEC 的特点,其中包括系统动态变化、移动终端在区域内的移动以及边缘服务器跨区工作,而这种“移动”往往带来更复杂的问题模型;分布式计算模式通常需要掌握用户的个人数据与隐私,任务卸载的安全性问题一直是高性能计算领域的重点问题。

本节将从挑战及其未来解决方案两方面对上述问题进行分析。

6.1 任务卸载技术的智能性

通过单一算法优化以解决任务卸载问题在某个目标下实现求解最优的传统任务卸载方案应用于动态、随机、大数据的复杂网络特性下往往会产生事倍功半的效果,传统任务卸载方案面临着巨大的挑战。由于智能体在动态的环境中利用探索、学习与寻找规律能够完成大规模任务的决策组合最优化,将人工智能技术与 MEC 环境下的任务卸载问题相结合,可以有效应对传统算法捉襟见肘的情况。

通过结合深度学习、强化学习等相关技术可以提高任务卸载策略的性能。现如今较为主流的基于深度强化学习的任务卸载算法主要以 MDP 建模为基础,围绕着 DQN 算法及其改进算法等 Value-based 方法与策略梯度算法、演员-评论家算法等 Policy-based 方法进行算法设计,已经能产生较为显著的效果,但由于仍处于发展阶段,上述神经网络模型仍能继续改进与优化。

传统的 MDP 建模在强化学习中通常为基于模型(model-based)的解决方案,意味着环境中能够提供明确的奖励与状态转移概率矩阵。然而,在一部分的场景中,智能体往往难以对环境中的状态信息进行观测。无模型(model-free)的 DRL 任务卸载方案能够对不确定环境进行建模。Min 等人[50]提出了一种基于 model-free 的任务卸载方案,其根据基站当前资源、与终端之间的传输速率以及策略奖励的估计值来综合选择目标卸载单位,进而在环境模型未知的情况下设计任务卸载策略。

在强化学习的发展中,不断产生了诸如无限动作空间下的大规模离线策略算法——深度确定性策略梯度算法[51]、基于 Soft Q-Learning 与 AC 的 Soft Actor-Critic 算法等新颖算法。通过将前沿优势算法与任务卸载相结合并发展,能更大程度地优化能耗与时延等目标。此外,随着网络规模的扩大,边缘服务器与终端用户的数量指数上升,单智能体算法或许难以处理、训练该环境下任务卸载问题的模型框架。此时,基于多智能体的强化学习(Multi-Agent Reinforcement Learning, MARL)与任务卸载技术相结合的算法应运而生。

任务卸载技术与 AI 相结合可以有效地解决复杂、动态、随机的大型网络架构下的边缘计算任务卸载问题,人工智能方向的不断推进将显著提高系统的服务性能,解决传统优化算法难以处理的问题。但加入人工智能技术后系统所增加的成本是否有意义以及如何保障算法性能的前提下降低相关技术产生的额外成本等问题亦值得推敲。

6.2 不同任务间的相互依赖性

现有任务卸载研究往往假设卸载队列中的任务具有相互独立性，但个别种类任务间可能存在相关依赖性。

因 AR 任务存在相关性[43]，作者为针对 AR 应用的任务卸载建立更精确的模型，将一个 AR 任务划分为原视频获取、环境跟踪、环境测绘、对象识别、渲染

五个具有相互关联性的任务，任务两两之间可能存在多种输入输出关系，并且原视频获取任务与渲染任务只能在终端本地执行，且子任务的输入依赖父任务的执行结果。作者通过将有向无环图与细粒度卸载相结合，在满足依赖关系约束的前提下，资源利用得到提高，系统总时延与总能耗亦得到了优化，如图 4 所示。

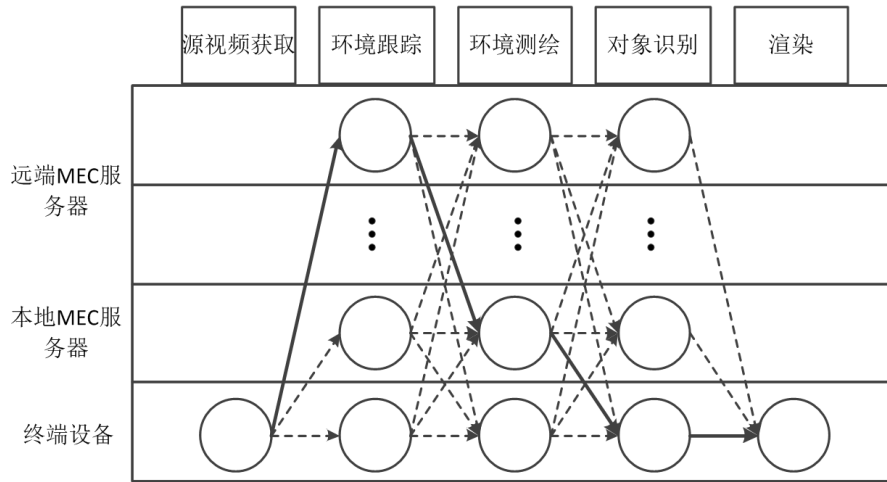


图 4 存在依赖的 AR 应用有向图

然而，由于卸载策略的制定还需要考虑边缘服务器的选择、多用户资源竞争等问题，如何同时在额外的依赖关系约束下设计任务卸载策略是一个挑战。

此外，某个设备上的任务可能需要另一个设备上的任务的输出作为输入，即不同终端设备间的任务亦存在数据依赖性。Yan 等人[52]特别地考虑两个用户终端之间的任务依赖性，基于二分搜索与一次爬升策略证明最佳策略，提出了一种求解最优卸载决策算法并扩展到多用户场景，但本文仅将依赖关系建模为顺序图，欠缺普适性；同时，在实际环境中，两个设备上具有依赖关系的任务的位置关系是系统传输模块的重要因素。因此，更加复杂的相关依赖性约束问题仍需进一步研究与讨论。

6.3 MEC 环境下的单位移动性

在 MEC 环境下，终端用户通常是处于随机运动的状态中，这种高移动性为 MEC 任务卸载带来了诸多困难。

当移动终端设备在部署有边缘服务器的基站覆盖范围内移动时，用户产生的位移会影响通信信道质量，倘若此时正处于任务卸载的过程中，数据上、下行传输会产生额外的能量消耗与时间延迟。

当移动终端设备在不同基站的覆盖范围间移动时，这种跨 MEC 服务区域的移动所造成的服务不连续性问题是任务卸载研究面临的一类更严峻的挑战，尤其是车联网、无人机等领域。大量终端用户频繁地在各基站覆盖范围边缘穿梭，网络拓扑持续变化，会导致通信链路受到干扰，传输效率受到影响，假如此刻用户正处于卸载任务过程中或等待卸载结果时，任务卸载甚至可能失败。现如今大部分研究通过路径轨迹预测、任务迁移等方案解决跨服务覆盖区域情况下的卸载错误问题，提高用户质量。Li 等人[53]以服务质量与时延最小化为目标，设计了一种基于移动预测的计算卸载切换策略，进而降低移动终端设备的不规则运动造成的任务卸载故障。二者的结合能够对设备的移动方向进行预测，预先将任务迁移至下一服务器，使任务卸载过程流畅过渡，提升了移动下的用户体验质量。但如果需要实现更精准的预测，则需要利用监督学习、深度学习等 AI 方案，这种技术的相互结合以及结合产生的额外成本需要进一步研究以寻找解决方案。

6.4 MEC 环境下的隐私安全性

MEC 环境下的任务卸载技术解决了云计算中心响

应延迟、终端用户资源有限的问题，并且避免了网络内所有用户数据上传至一个云端而产生的隐私安全问题，但边缘服务器仍需要获取用户数据，数据传输过程中被攻击仍具有较高的可能性，并且 MEC 节点的异构性与分布式的特性也给集中管理与保护带来了难度。MEC 模式仍存在着隐私泄露、边缘服务器安全与数据安全的威胁，如何有效保证 MEC 的安全性和隐私性依旧是值得探讨的问题与挑战。

针对 MEC 的隐私与安全问题，近年来研究人员逐渐将一些隐私保护技术与高性能计算相结合，形成了一种新型的计算模式——隐私计算 (Privacy Computing, PC)。PC 是在保证数据进行分析计算的过程中不泄露给数据参与方原始数据的一系列信息技术[54]。隐私计算通过 AI、密钥研究、可信硬件增强等方法，对边缘计算云环境的数据输入、计算、输出、存储等阶段进行优化，一定程度上解决了边缘节点数据保密性、攻击检测、数据保护、身份隐私、位置隐私等实际问题。

现如今，PC 拥有两类主要的研究方向：一类是基于密钥研究的隐私计算技术，主要为安全多方计算技术；另一类是与人工智能相融合的隐私计算技术，主要为联邦学习。此外，还有以可信执行环境为主的硬件增强隐私计算技术，其通过在 CPU 中独立创建一个可信的执行区域以确保数据与应用的安全问题。

安全多方计算技术由姚期智院士等人[55]于 1982 年创立。该技术是指在缺乏可信单位监督的情况下，将一个待解决函数交由多个单位合作完成计算任务，每一方仅仅了解本单位的计算结果，不能获取其他任意一方的数据，保证了整个过程的机密性，如图 5 所示。

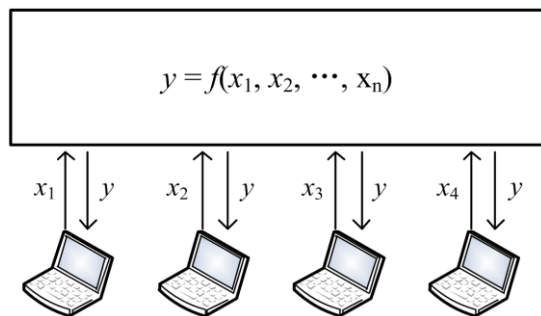


图 5 安全多方计算

联邦学习 (Federated Learning, FL) 本质是一种人工智能应用。FL 通过交互模型参数来完成多方联合的模型训练，其在多个数据源共同参与模型训练时并不需要原始数据集中流转，即“原始数据不出本地”，进而

最大限度地用户的数据隐私妥善保护。在保障数据不出客户端的情况下，兼顾分布式计算和联合计算的优势，并且资源紧缺的边缘服务器只需在本地进行计算，算力成本更低。但 FL 的代价是较低的性能与通信效率，并且虚假、有害的数据仍然可能从可信的参与方中被引入模型。

7 总结

现如今，传统云计算的服务模式已难以同时满足网络边沿用户对于低时延、低能耗、高计算量的要求，MEC 应运而生。任务卸载是移动边缘计算领域的关键技术之一，解决了单一云计算的 QoS、用户终端的低时延或低能耗等方面问题。本文首先介绍了 MEC 的基本概念、架构与任务卸载技术的基本概念、发展历程、流程、参考要素，以及目前应用于移动边缘计算环境下任务卸载领域中的两种主流方案——启发式及元启发式算法与强化学习。而后，以这两类方法作为区分标准，本文将任务卸载研究划分为基于算法优化的传统任务卸载方案与基于强化学习的交互任务卸载方案并对文献以最小化能耗、最小化时延、均衡考虑能耗时延这三类优化目标进行总结与归纳。最后，本文总结了目前任务卸载研究所存在的问题、面临的挑战并展望未来研究方向与方法，为后续移动边缘计算环境下任务卸载领域的研究工作提供帮助与参考。

参考文献

- [1] Ejaz A, Mubashir H. Mobile Edge Computing: Opportunities, solutions, and challenges [J]. *Future Generation Computer Systems*, 2017, 70: 59-63.
- [2] Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading [J]. *IEEE Communications Surveys & Tutorials*, 2017, 19 (3): 1628-1656.
- [3] Abbas N, Zhang Y, Taherkordi A, et al. Mobile edge computing: A survey [J]. *IEEE Internet of Things Journal*, 2017, 5 (1): 450-465.
- [4] 李强, 杜婷婷, 童钊, 张锦, 王胜春. 移动边缘计算中基于深度强化学习的依赖任务卸载研究 [J]. *小型微型计算机系统*, 2022, 1-8.
- [5] Tong Z, Ye F, Mei J, et al. A novel task offloading algorithm based on an integrated trust mechanism in mobile edge computing [J]. *Journal of Parallel and Distributed Computing*, 2022, 169: 185-198.

- [6] 宁振宇, 张锋巍, 施魏松. 基于边缘计算的可信执行环境研究 [J]. 计算机研究与发展, 2019, 56 (7): 1441-1453.
- [7] 施魏松, 张星洲, 王一帆, 张庆阳. 边缘计算: 现状与展望 [J]. 计算机研究与发展, 2019, 56 (1): 69-89.
- [8] Jiang C, Cheng X, Gao H, et al. Towards computation offloading in edge computing: A survey [J]. IEEE Access, 2019, 7: 131543-131558.
- [9] Wei P, Guo K, Li Y, et al. Reinforcement Learning-Empowered Mobile Edge Computing for 6G Edge Intelligence [J]. IEEE Access, 2022, 10: 65156-65192.
- [10] Shakarami A, Ghobaei-Arani M, Shahidinejad A. A survey on the computation offloading approaches in mobile edge computing: a machine learning based perspective [J]. Computer Networks, 2020, 182: 107496.
- [11] ETSI, Mobile-edge computing introductory technical white paper. White Paper. Mobile-Edge Computing Industry Initiative, 2014, 1089-7801.
- [12] Pallis G, Vakali A. Insight and perspectives for content delivery networks [J]. Communications of the ACM. 2006. 49 (1): 101-106.
- [13] Satyanarayanan M, Bahl V, Caceres R, et al. The case for VM-based cloudlets in mobile computing [J]. IEEE Pervasive Computing. 2009, 8 (4): 14-23.
- [14] Wang T, Liang Y, Tian Y, et al. Solving coupling security problem for sustainable sensor-cloud systems based on fog computing [J]. IEEE Transactions on Sustainable Computing, 2021, 6 (1): 43-53.
- [15] 周悦芝, 张迪. 近端云计算: 后云计算时代的机遇与挑战 [J]. 计算机学报, 2019, 42 (4): 677-700.
- [16] Zhang Y, Liang Y, Yin M, et al. Survey on the Methods of Computation Offloading in Mobile Edge Computing [J]. Chinese Journal of Computers, 2021, 44 (12): 2406-2430.
- [17] Mao Y, You C, Zhang J, et al. A Survey on Mobile Edge Computing: The Communication Perspective [C]. IEEE Communications Surveys & Tutorials, 2017: 2322-2358.
- [18] Wong V, Schober R, Ng D, et al. Key Technologies for 5G Wireless Systems [D]. Cambridge: Cambridge University Press, 2017: 1-2.
- [19] Zhang Y, Liu H, Jiao L, et al. To offload or not to offload: An efficient code partition algorithm for mobile cloud computing. [C] // Proceedings of the 2012 IEEE 1st International Conference on Cloud Networking. Paris, French, 2012: 80-86.
- [20] 赵志为, 闵革勇. 边缘计算: 原理、技术与实践 [M]. 北京: 机械工业出版社, 2021: 103-104.
- [21] 温文波, 杜维. 蚁群算法概述 [J]. 石油化工自动化, 2002 (1): 19-22.
- [22] Topcuoglu H, Hariri S, Wu M. Performance-effective and low-complexity task scheduling for heterogeneous computing [J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13 (3): 260-274.
- [23] Sagnika s, Bilgaiyan s, Mishra, et al. Workflow scheduling in cloud computing environment using bat algorithm [C] // In proceedings of First International Conference on Smart System, Innovations and Computing, Singapore, 2018: 149-163.
- [24] Alireza Askarzadeh. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm [J]. Computers & Structures, 2016, 169: 1-12.
- [25] Beegom A, Rajasree M. Integer-PSO: a discrete PSO algorithm for task scheduling in cloud computing system [J]. Evolutionary Intelligence, 2019, 12 (2): 227-239.
- [26] 吉根林. 遗传算法研究综述 [J]. 计算机应用与软件, 2004, 21 (2): 69-73.
- [27] Zhang L, Wang L, Xiao M, et al. EM_WOA: A budget-constrained energy consumption optimization approach for workflow scheduling in clouds [J]. Peer-to-Peer Networking and Applications, 2022, 15: 973-987.
- [28] Richard S. Sutton, Andrew G. Barto. 强化学习 [M]. 第二版. 俞凯, 等译. 北京: 电子工业出版社, 2019: 10-11.
- [29] Tong Z, Deng X, Mei J, et al. Response time and energy consumption co-offloading with SLRTA algorithm in cloud-edge collaborative computing [J]. Future Generation Computer Systems, 2022, 129: 64-76.
- [30] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518 (7540): 529-533.
- [31] Keqin Li. Design and Analysis of Heuristic Algorithms for Energy-Constrained Task Scheduling With Device-Edge-Cloud Fusion [J]. IEEE Transactions on Sustainable Computing. 2022, 1-13.
- [32] Keqin Li. Non-clairvoyant and randomised online task offloading in mobile edge computing [J]. International Journal of Parallel, Emergent and Distributed Systems, 2022, 37: 413-424.
- [33] Cheng S, Huang J, Chen Z, et al. Approximated Assignment Algorithms for Unordered and Ordered Tasks in Data Shared MEC Systems [J]. IEEE Transactions on Mobile Computing, 2023, 22 (4): 1968-1987.
- [34] Guo F, Chen X, Li X, et al. An Energy-Efficient Dynamic Scheduling Method of Deadline-Constrained Workflows in a Cloud Environment [J]. IEEE Transactions on Network and Service Management, 2022, 1-1.

- [35] Saleem U, Liu Y, Li Y, et al. Mobility aware joint task scheduling and resource allocation for cooperative mobile edge computing [J]. *IEEE Transactions on Wireless Communications*, 2020, 20 (1): 360-374.
- [36] 姚枝秀, 夏士超, 李云. 不确定网络环境下的任务卸载和资源分配算法 [J]. *中国科学: 信息科学*, 2022, 52: 1349-1361.
- [37] Chen X, Zhang J, Lin B, et al. Energy-Efficient Offloading for DNN-Based Smart IoT Systems in Cloud-Edge Environments [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2022, 33 (3): 683-697.
- [38] Chen X, Jiao L, Li W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. *IEEE/ACM Transactions on Networking*, 2016, 24 (5): 2795-2808.
- [39] Chen J, Deng Q, Yang X. Non-cooperative game algorithms for computation offloading in mobile edge computing environments [J]. *Journal of Parallel and Distributed Computing*, 2023, 172: 18-31.
- [40] 刘振鹏, 郭超, 王仕磊, 陈杰, 李小菲. 基于博弈论和启发式算法的超密集网络边缘计算卸载 [J]. *计算机工程*, 2022, 48 (12): 54-61, 71.
- [41] Yang P, Zhang Y, Lv J. Load Optimization Based on Edge Collaboration in Software Defined Ultra-Dense Networks [J]. *IEEE Access*, 2020, 8: 30664-30674.
- [42] Sadiki A, Bentahar J, Dssouli R, et al. Deep reinforcement learning for the computation offloading in MIMO-based Edge Computing [J]. *Ad Hoc Networks*, 2023, 141: 103080.
- [43] Chen X, Liu G. Energy-Efficient Task Offloading and Resource Allocation via Deep Reinforcement Learning for Augmented Reality in Mobile Edge Networks [J]. *IEEE Internet of Things Journal*, 2021, 8 (13): 10843-10856.
- [44] Liu Q, Shi L, Sun L, et al. Path Planning for UAV-Mounted Mobile Edge Computing with Deep Reinforcement Learning [J]. *IEEE Transactions on Vehicular Technology*, 2020, 69 (5): 5723-5728.
- [45] Gao Z, Hao W, Han Z, et al. Q-Learning-Based Task Offloading and Resources Optimization for a Collaborative Computing System [J]. *IEEE Access*, 2020, 8: 149011- 149024.
- [46] Huang L, Bi S, Zhang Y. Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks [J]. *IEEE Transactions on Mobile Computing*, 2020, 19 (11): 2581-2593.
- [47] Tang M, Wang V. Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems [J]. *IEEE Transactions on Mobile Computing*, 2022, 21 (6): 1985-1997.
- [48] Tan L, Kuang Z, Zhao L, et al. Energy-Efficient Joint Task Offloading and Resource Allocation in OFDMA-based Collaborative Edge Computing [J]. *IEEE Transactions on Wireless Communications*. 2022, 21 (3): 1960-1972.
- [49] Qian Y, Hu L, Chen J, et al. Privacy-aware service placement for mobile edge computing via federated learning [J]. *Information Sciences*, 2019, 505: 562-570.
- [50] Min M, Xiao L, Chen Y, et al. Learning-based computation offloading for IoT devices with energy harvesting [J]. *IEEE Transactions on Vehicular Technology*, 2019, 68 (2): 1930-1941.
- [51] Liu Y, Yang C, Jiang L, et al. Intelligent edge computing for IOT-based energy management in smart cities [J]. *IEEE Network*, 2019, 33 (2): 111-117.
- [52] Yan J, Bi S, Zhang Y. Optimal offloading and resource allocation in mobile edge computing with inter user task dependency [J]. *IEEE Transactions on Wireless Communications*, 2020, 19 (1): 235-250.
- [53] Li B, Niu L, Huang X, et al. Mobility Prediction-Based Computation Offloading Handoff Strategy for Vehicular Edge Computing [J]. *Journal of Electronics & Information Technology*, 2020, 42 (11): 2664-2670.
- [54] 国家发展和改革委员会, 中央网信办, 工业和信息化部, 等. 全国一体化大数据中心协同创新体系算力枢纽实施方案 [R], 2021.
- [55] Yao A C. Protocols for secure computations [C] // 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). *IEEE*, 1982: 160-164.