基于博弈论的UAV辅助MEC系统中飞行路径及任务卸载优化研究

韦熳熠，王高才，温一虎

引用本文

韦熳熠，王高才，温一虎. 基于博弈论的UAV辅助MEC系统中飞行路径及任务卸载优化研究[J]. 计算机科学，2026，53(2)：396-405.

WEI Manyi, WANG Gaocai, WEN Yihu. Game Theory-based Optimization of Flight Paths and Task Offloading in UAV-assisted MECSystems [J]. Computer Science, 2026, 53(2): 396-405.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250300088

基于博弈论的 UAV 辅助 MEC 系统中飞行路径及任务卸载优化研究

韦熳熠　王高才　温一虎
广西大学计算机与电子信息学院　南宁 530004
(weimygx@163.com)

摘　要　在传统的移动边缘计算(Mobile Edge Computing,MEC)系统中,MEC 服务器通常部署于固定位置,易受多径效应和非视距路径影响,造成通信阻塞。采用无人机(Unmanned Aerial Vehicle,UAV)辅助 MEC,已经成为一种新的趋势。首先,为使 UAV 能够为区域内所有设备提供服务并延长运行时间与相关网络寿命,提出 UAV 辅助 MEC 中基于终端用户分簇的飞行路径优化方法,使用 K-means 算法将终端用户划分为多个簇,分簇后,基于簇中心将最短飞行路径问题转换为旅行商问题,并采用混沌博弈优化结合 2-Opt 的方法解决该问题。然后,为优化系统能耗与时延,从用户角度出发设计出基于势博弈的任务卸载策略方法,将全局优化问题建模为势博弈模型,并证明其至少存在一个纳什均衡(Nash Equilibrium,NE),且最优 NE 恰好对应全局最优解。实验结果表明,与其他经典算法相比较,所提飞行路径与任务卸载优化算法能在有效最小化系统的能耗与时延加权和的同时,确保服务覆盖整个区域。

关键词:势博弈;混沌博弈;移动边缘计算;无人机;任务卸载
中图分类号　TP301

Game Theory-based Optimization of Flight Paths and Task Offloading in UAV-assisted MEC Systems

WEI Manyi,WANG Gaocai and WEN Yihu
School of Computer and Electronic Information,Guangxi University,Nanning 530004,China

Abstract　In traditional MEC systems,fixed MEC server deployments are susceptible to communication blockages due to multipath and non-line-of-sight(NLOS) effects. UAV-assisted MEC has emerged as a solution. This paper proposes a UAV-assisted MEC system with a clustering-based flight path optimization method to extend the UAV's operational time and the network lifetime. K-means clustering partitions users into clusters,and the shortest UAV flight path problem among cluster centers is formulated as a TSP and solved using Chaos Game Optimization(CGO) combined with 2-Opt. A potential game-based task offloading strategy is then designed to optimize system energy consumption and latency from the users' perspective. The global optimization problem is modeled as a potential game with at least one Nash Equilibrium(NE),which corresponds to the global optimal solution. Experimental results show that the proposed methods effectively minimize system energy consumption and latency while ensuring complete service coverage.

Keywords　Potential game,Chaos game,Mobile edge computing,Unmanned Aerial Vehicle,Task offloading

1　引言

随着 5G 网络的普及,用户终端数量激增,进而导致终端生成的任务数据量急剧上升。例如,自动驾驶、智能医疗和增强现实(Augmented Reality,AR)等应用产生的大量数据需要实时处理[1]。如果将数据传输至云端进行处理,不仅会显著增加云服务器的负载,还会导致用户延迟的增加。移动边缘计算(MEC)这项新兴技术应运而生。MEC 通过将服务器部署在网络边缘,使用户能将任务卸载至 MEC 服务器进行处理,从而大幅降低能耗和时延,提高系统的可靠性与整体网络效率[2]。

在传统的 MEC 网络中,MEC 服务器通常放置于地面,存在明显的局限性,易因多径效应和非视距路径造成信号阻塞,严重影响通信质量[3]。相较之下,无人机(UAV)搭载 MEC 服务器可建立优质空对地视距链路,其三维机动性不仅能够弥补地面部署的固有缺陷,还可根据用户分布动态调整服务节点位置,显著提升网络覆盖灵活性[4-5]。这种具备自主飞行能力与实时路径规划特性的空中平台兼具部署成本低、

到稿日期:2025-03-17　返修日期:2025-06-06
基金项目:国家自然科学基金(62062007);广西自然科学基金(2025GXNSFAA069236)
This work was supported by the National Natural Science Foundation of China(62062007) and Natural Science Foundation of Guangxi,China (2025GXNSFAA069236).
通信作者:王高才(wanggcgx@163.com)

响应速度快等优势,可作为高效无线中继或智能基站,有效增强蜂窝网络的连接稳定性并扩大覆盖范围[6]。

博弈论可以解决多个决策主体间存在利益冲突的问题,即各利益相关者或参与者如何根据自身的能力、掌握的信息以及相互之间的策略影响来做出最有利于自己的决策[7-8]。在这个过程中,决策主体之间相互制约,形成每个决策者相互对抗的局面,任何一个决策者都不被允许选择只有自身收益的决策,而是根据其他决策者的信息选择当前最利的情况,并寻求最均衡的解决方案[9]。在 MEC 场景中,不同用户之间的决策会相互影响,博弈论能够帮助分析和预测这些用户如何在竞争和合作中找到最优策略,以实现各自的目标。

本文主要研究 UAV 辅助 MEC 系统中 UAV 的路径优化和用户设备的计算任务卸载至 UAV 进行计算的问题。其中,UAV 可以集成 MEC 服务器处理用户设备的卸载任务,UAV 与用户设备之间建立的通信链路可视为视径路径。本文主要贡献如下:

1)为使 UAV 能够为区域内所有设备提供通信和计算服务,并延长其生存时间与相关网络寿命,提出一种基于 K-means 聚类和混沌博弈的路径优化方法。该方法摒弃传统的对物理区域进行划分的方法,通过 K-means 将用户设备划分为多个簇,分簇后,基于簇中心将路径规划问题转换为旅行商问题(Traveling Salesman Problem,TSP),并创新性地结合混沌博弈优化与 2-Opt 算法求解最短飞行路径。该方法在延长 UAV 生存时间的同时,确保了区域内所有设备均获得稳定服务。

2)为实时优化系统能耗与时延,提出一种基于势博弈的任务卸载策略方法,将全局优化问题拆解为势博弈问题,并证明至少存在一个纳什均衡(NE),且最优 NE 恰好对应全局最优解。该方法相较于其他卸载策略算法,在优化性能和计算时延上表现良好。

2 相关研究

传统的将 MEC 服务器部署于固定地面的方法存在着一定的局限性:由于多径反射和非视距传播等环境因素的影响,信号在传输过程中容易受到干扰而衰减,导致通信质量下降[10]。采用 UAV 辅助 MEC 正逐渐成为一种创新的解决方案,通过动态调整位置和优化信号路径,其能显著提升系统性能和服务质量。

在 UAV 辅助 MEC 系统中,UAV 通过与多个用户设备建立连接,接收并处理其卸载的时延敏感任务,同时按预设优化路径移动,以提供高效服务[11-12]。现有研究主要集中在任务卸载策略和 UAV 飞行轨迹的优化上。例如:文献[11]将 MEC 与 UAV 结合,设计了一种联合优化算法,以最小化系统能耗和时延加权和为目标,通过差分进化算法优化任务卸载策略,并利用 OAC 算法优化 UAV 路径,从而提升系统性能;文献[12]提出一种新的变种群大小遗传轨迹规划算法,用于规划 UAV 的飞行路径,以缩短用户设备与基站之间的距离,降低设备能耗;文献[13]则研究 UAV 辅助 NOMA-MEC 系统的联合轨迹优化与资源分配问题,提出一种基于 LOTRS 的低复杂度在线优化方案,以最小化系统开销。

然而,现有研究在关注 UAV 飞行路径的同时,极少关注 UAV 服务范围是否覆盖全部用户设备,仅有少部分研究尝试解决这一问题。例如:文献[14]为让大面积区域内所有终端设备都有被服务的机会,将区域划分为若干子区域,UAV 按固定路线在子区域间巡逻,以覆盖更广泛的终端设备;文献[15]基于最优传输理论视角对整个区域进行划分,使 UAV 在各区域中心悬停服务,确保 UAV 覆盖区域范围内的所有设备;文献[16]则采用 K-means 算法对终端设备进行分簇,UAV 在每个簇中心悬停以提供服务,确保 UAV 服务能覆盖全部终端设备。上述算法虽考虑到 UAV 服务覆盖范围的问题,但规划的 UAV 飞行路径普遍较长,严重缩短了 UAV 的生存时间。

UAV 辅助 MEC 系统的主要性能指标包括时延、能耗和成本等[11-12]。由于终端设备对时延和能耗极为敏感,相关研究多集中于优化这两方面。文献[17-19]以最小化系统时延为目标,分别采用遗传算法、联合优化算法和基于深度强化学习的调度策略进行优化。文献[20-21]则以最小化系统能耗为目标,提出基于多目标优化模型和负载均衡的资源分配策略。最小化时延和最小化能耗都有其各自的应用场景,平衡时延和能耗的需求比单一最小化时延或者最小化能耗更具意义[22]。文献[23-25]进一步平衡时延和能耗的优化需求,采用元启发式算法(如二元灰狼优化器和种群多样性-二元粒子群优化算法)进行联合优化,以提升用户的服务质量。然而,多数研究在优化系统能耗和时延时,未充分考虑用户设备的能耗和时延,可能会因系统性能的优化而牺牲用户体验。

任务卸载策略是优化 MEC 网络性能的关键。合理的卸载策略不仅能够优化系统性能,还能提升用户体验。随着网络环境复杂性的提升和设备数量的增加,传统卸载策略算法已难以满足大规模网络的需求。对此,改进的传统算法、深度强化学习和博弈论等先进算法被广泛用于优化任务卸载策略。例如,文献[26-29]采用深度强化学习构建决策模型,优化卸载策略,以适应动态无线信道条件,相较于传统卸载策略算法,其极大地提升了系统性能。然而,这些方法通常需要大量资源进行预训练,对训练数据的需求也较高。在资源受限和实时性要求高的情况下,改进的传统卸载策略算法或博弈算法成为有效的替代方案。例如,文献[17]和文献[30]采用改进的遗传算法来优化卸载策略,通过二进制编码、优化选择函数及考虑任务优先级约束等,在多服务器场景下确定最佳服务器,从而降低任务延迟与时延。文献[24]和文献[31]采用改进的粒子群算法,以优化系统性能为目标,通过迭代更新粒子速度与位置,不断探索解空间,在多维空间中求解最优值,即最佳卸载策略。文献[14,32-33]基于博弈论提出不同的博弈模型,用于优化系统性能。文献[14]提出一个混合博弈模型,通过混合策略博弈和子模博弈分别确定用户设备的移动概率和卸载数据量,从而得到最优卸载策略。文献[32]以优化系统性能为目标构建全局优化问题,在博弈框架下将全局问题解耦为局部优化问题,从而博弈得到最佳卸载策略。文献[33]则提出一种基于 Stackelberg 博弈的用户卸载策略,通过维持 MEC 服务器与移动用户之间的优势博弈关系,调

整各自的策略，直至各自的效用最大化。在复杂的动态场景中，结合激励机制的博弈算法表现更为优异，文献[34]和文献[35]提出一种基于 Stackelberg 博弈的激励机制用于资源分配，实现了系统效用最大化，相较于传统的博弈算法更加高效。

综上所述，当前 UAV 辅助 MEC 研究中存在着以下几个问题：

1)多数研究中，UAV 沿着飞行路径提供服务时未考虑区域内的所有用户设备，可能导致部分用户设备无法接收到服务。部分研究虽考虑到了 UAV 服务范围覆盖全部终端设备的情况，但飞行路径随之增加，UAV 续航时间大幅缩短，系统持续服务的能力受到影响。

2)从用户服务视角看，多数研究在以优化系统能耗和时延为目标时，未充分考虑用户设备的性能差异，这可能会因考虑系统性能而牺牲用户设备的服务体验。

3)传统卸载策略算法难以适应大规模网络需求，而先进算法(如深度强化学习)虽能提升系统性能，但对资源和训练数据需求较高，在资源受限和实时性要求高的情况下难以应用。

3 系统模型

系统模型如图 1 所示。模型中有 M 台用户设备，用集合表示为 $M=\{1,2,3,\cdots,M\}$，它们位于 Ω 区域内的任意位置，第 m 个用户设备的位置信息用 $S_m=(x_m,y_m)$ 表示。系统内，基站可以将用户设备位置、UAV 位置、数据信息等实时信息进一步分发给各用户设备和 UAV，UAV 为可通信范围内的用户设备提供服务。

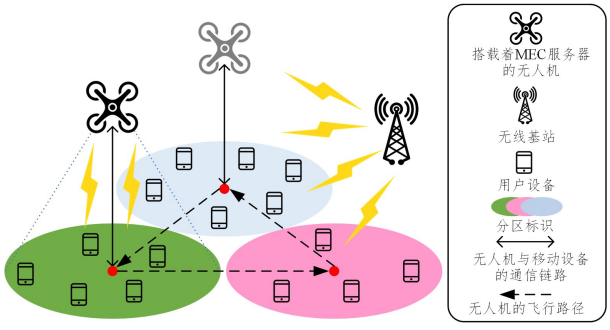


图 1 系统模型

Fig. 1 System model

系统中每个时隙 t 可以分割为两个阶段：第一阶段是 UAV 的飞行时间 t_{fly}，第二阶段是 UAV 提供服务的时间 t_{svc}。用户设备在每个时隙内都有任务要处理，任务大小集合为 $L=\{L_1,L_2,\cdots,L_m\}$。UAV 在时隙 t 的位置表示为 $N(t)=\{X(t),Y(t),H(t)\}$，其中 H 为 UAV 的飞行高度，UAV 在每个时隙内为通信范围内的用户设备提供通信和计算服务。考虑到 UAV 搭载的 MEC 服务器的负载压力与用户设备的任务大小，将用户设备的卸载方法分为完全卸载和部分卸载，以在保证用户设备计算需求的同时避免 MEC 服务器过载。

UAV 的服务位置会显著影响网络性能和覆盖范围，容易导致系统中部分用户设备接收到质量不佳的服务或无法接收到服务。为此，本文采用 K-means 对用户设备进行分簇，通过设定每个时隙内 UAV 在簇中心映射的三维坐标为用户设备提供服务，能确保所有的用户设备都接收到 UAV 所提供的优质服务。

采用 K-means 对用户设备进行分簇的表示如下：给定一组用户设备，算法将用户设备划分为 $u_1,u_2,\cdots,u_P$ 共 P 簇，每个簇之间是互斥的集体穷举集合，即 $u_a\cap u_b=\varnothing, a\neq b$ 且 $u_1\cup u_2\cup\cdots\cup u_P=U$。通过最小化用户设备与其集质心距离的平方偏差，可以降低 UAV 与用户设备之间的路径损耗和发射功率。簇的大小设置在 UAV 的服务覆盖范围内。$s_p=(x_p,y_p)$ 为簇 u_p 的簇中心，$g_p=(x_p,y_p,h)$ 为基于簇中心映射的 UAV 固定服务点的三维位置，其中 $p\in P$，则簇中心与用户设备之间的关联可以通过求解式(1)得到。

$$\min_{\{u_1,\cdots,u_P\}}\sum_{p=1}^{P}\sum_{m\in u_p}\|S_m-s_p\|^2 \tag{1}$$

在动态场景下，用户设备为保持与 UAV 的通信服务质量，会主动以较低的受限速率 v_m 向 UAV 服务点趋近移动。基于高斯-马尔可夫移动模型分析[36]，用户设备在 UAV 单次飞行周期内的最大位移量满足 $\Delta d_{\max,m}=v_m\cdot T_{\text{fly}}\leqslant 0.3d_p$($T_{\text{fly}}$ 为 UAV 单次飞行周期，d_p 为簇 p 的覆盖半径)，其活动范围始终位于初始分簇的容限内，系统可在单次飞行周期内保持分簇拓扑稳定，无须实时更新。为平衡系统动态适应性与资源效率，UAV 采用周期性路径更新策略：仅在完成全服务点遍历后，基于最新用户设备位置分布重新执行 K-means 分簇并优化飞行路径。

3.1 通信模型

假设时隙 t 内，用户设备 m 生成的计算任务需要卸载到 UAV 上进行处理，故系统需要给用户设备分配上传和下载的信道和传输带宽。在这一过程中，UAV 与用户设备之间的信道增益是一个重要的参数，可以表示为：

$$h_{\text{up}}(m,t)=\beta_0 d^{-2}(k,t)=\frac{\beta_0}{H^2+\|N(t)-S_m\|^2} \tag{2}$$

其中，β_0 表示通信距离为 1 米时的信号增益，$d(k,t)$ 表示时隙 t 内 m 与 UAV 在三维坐标系中的欧几里得距离。假设 UAV 在飞行范围 Ω 内，UAV 和 m 的通信范围定义为：

$$N(t)-S_m\leqslant d_{\max} \tag{3}$$

其中，$d_{\max}$ 为 UAV 和 m 之间的最大距离。

获取信道增益后，系统可以基于无线链路的传输条件计算数据的传输速率。数据传输速率的计算式为：

$$r_{\text{up}}(m,t)=B_0\log_2\left(1+\frac{Ph_{\text{up}}(m,t)}{\sigma_0^2}\right) \tag{4}$$

其中，σ_0^2 是加性白噪声功率，B_0 表示传输带宽，P 表示用户设备的信号发射功率。

设置 UAV 与 m 在时隙 t 内的通信策略，即 m 可以选择是否将任务卸载至 UAV：

$$c(m,t)=\begin{cases}1, & \text{offload}\\ 0, & \text{local}\end{cases} \tag{5}$$

m 在时隙 t 内选择将任务卸载至 UAV 的策略分为部分卸载和完全卸载：

$$a(m,t)=\begin{cases}1, & \text{total offloading}\\ 0, & \text{partial offloading}\end{cases} \tag{6}$$

3.2 计算模型

3.2.1 全部卸载

假设时隙 t 内,用户设备 m 选择将任务全部卸载至 UAV,则总的卸载延迟为:

$$T_{\text{to},\text{up}}(m,t)=\frac{L_m}{r_{\text{up}}(m,t)} \tag{7}$$

数据卸载能耗表示为:

$$E_{\text{to},\text{up}}(m,t)=PT_{\text{to},\text{up}}(m,t) \tag{8}$$

数据卸载到 UAV 后,在服务器上处理数据的时间定义为:

$$T_{\text{to},n}(m,t)=\frac{L_mC_n}{f_n} \tag{9}$$

其中,C_n 为 UAV(服务器)处理一位数据所需的 CPU 周期,f_n 为 UAV 的计算资源。此时,UAV 的计算能耗为:

$$E_{\text{to},n}(m,t)=K_n(f_n)^3T_{\text{to},n}(m,t) \tag{10}$$

其中,K_n 为服务器 CPU 的有效电容系数。

因此,m 采用全部卸载策略时所需要的总延迟和能耗为:

$$T_{\text{to}}(m,t)=T_{\text{to},\text{up}}(m,t)+T_{\text{to},n}(m,t) \tag{11}$$

$$E_{\text{to}}(m,t)=E_{\text{to},\text{up}}(m,t)+E_{\text{to},n}(m,t) \tag{12}$$

3.2.2 部分卸载

假设时隙 t 内,用户设备 m 按照卸载比例 η 将任务部分卸载至 UAV,即本地处理 $(1-\eta)L_m$ 个任务,则本地处理时延和能耗为:

$$T_{\text{pa},m}(m,t)=\frac{(1-\eta)L_mC_m}{f_m} \tag{13}$$

$$E_{\text{pa},m}(k,t)=K_m(f_m)^3T_{\text{pa},m}(m,t) \tag{14}$$

其中,C_m 为 m 处理一位数据所需要的 CPU 周期数,f_m 和 K_m 分别为 m 的计算资源和 CPU 的有效电容因数。剩余的 ηL_m 个任务卸载至 UAV 上进行计算。参考全部卸载策略计算公式,任务卸载的时间和能耗可以表示为:

$$T_{\text{pa},\text{up}}(m,t)=\frac{\eta L_m}{r_{\text{up}}(m,t)} \tag{15}$$

$$E_{\text{pa},\text{up}}(m,t)=PT_{\text{pa},\text{up}}(m,t) \tag{16}$$

服务器处理任务的时延和能耗为:

$$T_{\text{pa},n}(m,t)=\frac{\eta L_mC_n}{f_n} \tag{17}$$

$$E_{\text{pa},n}(m,t)=K_n(f_n)^3T_{\text{pa},n}(m,t) \tag{18}$$

因此,当卸载策略为部分卸载时,m 在时隙 t 内所需要的总能耗和时延为:

$$T_{\text{pa}}(m,t)=\max\{T_{\text{pa},m}(m,t),T_{\text{pa},\text{up}}(m,t)+T_{\text{pa},n}(m,t)\} \tag{19}$$

$$E_{\text{pa}}(m,t)=E_{\text{pa},m}(m,t)+E_{\text{pa},\text{up}}(m,t)+E_{\text{pa},n}(m,t) \tag{20}$$

为了使部分卸载的延迟最小,可以调整卸载比 η,得到 $\min(T_{\text{pa}}(m,t))$。每个用户设备在任意时隙内的任务量是恒定的,则 $T_{\text{pa},m}$ 和 $T_{\text{pa},\text{up}}+T_{\text{pa},n}$ 彼此成反比。根据数学增减函数的特殊性,当 $T_{\text{pa},m}=T_{\text{pa},\text{up}}+T_{\text{pa},n}$ 时,可得到最小值。因此,卸载比可由式(21)求得。

$$\frac{(1-\eta)L_mC_m}{f_m}=\frac{\eta L_m}{r_{\text{up}}(m,t)}+\frac{\eta L_mC_n}{f_n}$$

$$\Rightarrow\eta=\frac{C_mf_nr_{\text{up}}}{(C_nf_m+C_mf_n)r_{\text{up}}(m,t)+f_mf_n} \tag{21}$$

3.2.3 本地计算

假设时隙 t 内,用户设备 m 选择在本地计算所有任务,则所需要的总能耗和时延为:

$$T_{\text{to}}(m,t)=\frac{L_mC_m}{f_m} \tag{22}$$

$$E_{\text{to}}(m,t)=K_m(f_m)^3T_{\text{to}} \tag{23}$$

3.3 问题定义

考虑最小化系统的能耗与时延的加权和,用户设备 m 在时隙 t 处的延时和能耗可以表示为:

$$\begin{aligned}T(m,t)=&(1-c(m,t))T_{\text{to}}(m,t)+\\&c(m,t)[a(m,t)T_{\text{to}}(m,t)+\\&(1-a(m,t))T_{\text{pa}}(m,t)]\end{aligned} \tag{24}$$

$$\begin{aligned}E(m,t)=&(1-c(m,t))E_{\text{to}}(m,t)+\\&c(m,t)[a(m,t)E_{\text{to}}(m,t)+\\&(1-a(m,t))E_{\text{pa}}(m,t)]\end{aligned} \tag{25}$$

UAV 沿着预设的飞行路径匀速飞行前进,时隙开始时获得飞行机动,在两个服务点之间匀速飞行,到达下一个服务点后停留,取得飞行机动后继续前进。因此,只需要考虑 UAV 与用户设备之间的通信和计算的能耗和时延。假设 UAV 在每个时隙结束时获取一个前进动作,所有用户设备重新生成它们的任务。问题可以被表示为:

$$\begin{aligned}&\text{P1}:\min_{A,N}\sum_{t=1}^{T}[\omega(\sum_{m=1}^{M}E(m,t))+(1-\omega)((\sum_{m=1}^{M}T(m,t))]\\&\text{s.t. C1}:N(t),U_m\in\Omega,\forall m\in M\\&\quad\text{C2}:c(m,t)\in\{0,1\}\\&\quad\text{C3}:a(m,t)\in\{0,1\}\\&\quad\text{C4}:\bigcup_{p\in P}g_p\subseteq\{N(1),N(2),\cdots,N(T)\}\\&\quad\text{C5}:N^{\text{start}}\to N^{\text{end}}\end{aligned} \tag{26}$$

4 优化算法

4.1 混沌博弈结合 2-Opt 的 UAV 路径优化算法

采用 K-means 对用户设备进行分簇,并基于簇中心灵活地设置 UAV 服务点位置,使得 UAV 与用户设备之间的通信距离尽可能短,且确保服务覆盖所有用户设备。进一步地,引入 TSP 的求解思路,通过计算 UAV 服务点之间的最短路径,确保 UAV 能够以最短路径飞行,从而减少飞行能耗,延长生命时间。TSP 是一种经典的组合优化问题,目标是找到一条经过所有服务点且路径长度最短的闭合回路[37],避免不必要的迂回和重复飞行,进而提高服务点的覆盖效率,降低系统的整体能耗和时延,延长 UAV 的生存时间。

通过式(1)对区域进行分簇后,基于簇中心计算得出 UAV 的 P 个飞行服务点 $G=\{g_0,g_1,\cdots,g_P\}$,UAV 在服务点处为用户设备提供通信和卸载服务。假设有 $i,j\in p$ 两个 UAV 服务点,d_{ij} 为 i 和 j 之间的距离,决策变量 $x_{ij}\in\{0,1\}$,$x_{ij}=1$ 表示 i 和 j 之间存在着一条路径,u_i 表示 g_i 在路径中被访问的顺序,$1\leqslant u_i\leqslant p-1,\forall i\in\{1,2,\cdots,p\}$,则 UAV 的最短飞行路径问题可表示为 TSP:

$$\begin{aligned}&\text{P2}:\min\sum_{i=0}^{p}\sum_{j=0,j\neq i}^{p}d_{ij}\cdot x_{ij}\\&\text{s.t. C1}:\sum_{j=0,j\neq i}^{p}x_{ij}=1,\forall i\in\{0,1,\cdots,p\}\\&\quad\text{C2}:\sum_{j=0,j\neq i}^{p}x_{ji}=1,\forall i\in\{0,1,\cdots,p\}\\&\quad\text{C3}:u_i-u_j+(p-1)\cdot x_{ij}\leqslant p-2,\forall i\neq j,1\leqslant i,j\leqslant p\end{aligned} \tag{27}$$

针对式(27)所提出的 TSP，混沌博弈优化(Chaos Game Optimization，CGO)算法能够模拟混沌系统的行为，生成初始解或候选解，然后采用博弈的思想对解进行选择和优化，从而逐步逼近最优解，即最短旅行商路径。CGO 算法是一种基于混沌理论的新型元启发式优化算法，由 Talatahari 于 2020 年首次提出[38]。其核心思想是利用混沌系统的随机性和遍历性来生成多样化的解，并通过博弈机制选择最优解并保持解的多样性，具有较强的全局搜索能力，能够有效地解决约束优化问题。

首先，将每个节点 g_i 的混沌种子初始化为均匀分布 U(0,1)，计算初始路径距离：

$$Seed_i \sim \mathrm{U}(0,1), \forall i \in \{0,1,\cdots,p\} \tag{28}$$

混沌种子的初始化是在混沌空间中随机分布各个节点的初始状态，为后续的混沌迭代提供初始条件。使用 Logistic 映射更新混沌种子，其中 r 是混沌参数，通常取值为 4，以确保系统处于混沌状态。混沌种子的更新可表示为：

$$Seed_i^{(t+1)} = r \cdot Seed_i^{(t)} \cdot (1 - Seed_i^{(t)}) \tag{29}$$

Logistic 映射是一种常见的混沌映射，通过迭代更新混沌种子，可以在混沌空间中生成新的混沌值，从而模拟混沌系统的行为，增加找到全局最优解的概率[38]。对式(29)更新后的混沌种子进行排序操作，依据其排序后的结果对节点序列进行重新排列。通过对混沌种子的排序，能够将混沌值有效映射至路径的顺序之上，从而构建出新的路径候选解 π'：

$$\pi' = sort(Seed_i^{(t+1)}), \forall i \in \{0,1,\cdots,p\} \tag{30}$$

生成混沌候选解后，需要通过博弈进行评估并指导优化方向[39]。定义博弈效用函数 Utility(u_i, u_j)和收益函数$\pi_i(s_i, s_{-i})$。效用函数用于量化 g_i 与 g_j 之间的效用，计算式为：

$$Utility(u_i, u_j) = \frac{1}{|u_i - u_j|} \tag{31}$$

式(31)表示节点 g_i 和节点 g_j 之间的效用与它们的访问顺序差的倒数成正比，鼓励节点在路径中相邻访问。

收益函数则用于评估节点 g_i 选择下一个访问节点 g_j 的收益，它不仅考虑节点 g_i 和 g_j 之间的效用，还考虑其之间的距离，计算式为：

$$\pi_i(s_i, s_{-i}) = -d_{ij} + \alpha \cdot Utility(u_i, u_j) \tag{32}$$

其中，α 用于平衡路径距离与节点访问顺序效用，取值为 0.618(采用混沌优化的黄金分割比例)[39]。

根据收益函数，从当前路径 π 和新路径候选 π' 中选择一个路径作为下一代路径 $\pi^{(t+1)}$：

$$\pi^{(t+1)} = Selection(\pi, \pi') \tag{33}$$

通过式(33)获取到的下一代路径 $\pi^{(t+1)}$ 是基于全局搜索所得到的，虽能有效避免陷入局部最优解，但在一定程度上忽略了部分局部搜索结果。引入 2-Opt 算法在局部范围内对路径 $\pi^{(t+1)}$ 进行优化，能进一步提高路径质量。2-Opt 算法是一种局部搜索算法，能够对当前路径进行优化，通过交换路径中的边，可以快速地找到最短路径，常用于 TSP 的优化场景[40-42]。对于每条边(g_i, g_{i+1})和(g_j, g_{j+1})，2-Opt 算法根据式(34)检查交换边是否可以缩短总路径长度，如果 $\Delta<0$，则进行交换，更新路径，否则保持。

$$\Delta = d_{g_i g_j} + d_{g_{i+1} g_{j+1}} - d_{g_{i+1} g_{i+1}} - d_{g_{j+1} g_{j+1}} \tag{34}$$

CGO 算法负责全局探索，为算法提供多样化的路径候选；而 2-Opt 算法则负责局部优化，对当前路径进行改进。这种结合使得算法能够在全局探索和局部优化之间取得平衡，逐步收敛到最优或近似最优解。CGO 结合 2-Opt(Chaos Game Optimization and 2-Opt，CGO-2Opt)的 TSP 路径优化算法如算法 1 所示。

算法 1 CGO-2Opt 的 TSP 路径优化算法

输入：G，D=[d_{ij}]，π，max_iter

输出：π

1. 根据式(28)初始化每个节点 g_i 的混沌种子
2. 定义博弈收益函数和效用函数
3. WHILE iter < max_iter：
4. 根据式(29)更新混沌种子并生成新路径候选 π'
5. 通过式(32)评估并选择最优策略 s_i 更新路径 π'
6. 使用 Opt 算法进行路径优化
7. 若达到最大迭代次数 max_iter 或路径不再变化，END WHILE
8. 输出最短路径 $\pi=\{g_0, g_1, \cdots, g_p\}$

4.2 基于势博弈的卸载策略优化算法

为进一步降低系统的能耗和时延，考虑从系统中用户设备的卸载策略出发，对式(26)进行优化。假设在时隙 t 内，区域内的用户设备已划分为 P 簇，此时 UAV 服务点坐标为 g_p，$p \in P$，可通信范围内的用户设备集合为 $V \in M$，用户设备所需要处理的任务大小为 $L=\{L_1, L_2, \cdots, L_v\}$，用户设备可以选择全部卸载、部分卸载或不卸载任务到 UAV 上。在当前场景下，可以将式(26)的目标优化函数转换为博弈目标函数，对其进行博弈求解。博弈过程中，将用户设备作为决策主体，卸载策略作为博弈策略，各个决策主体之间考虑自身的能耗与延迟以及博弈目标函数，从而选择最有利于自己的决策(策略)，最小化系统能耗与延迟的加权和。

博弈模型主要依赖于策略空间、效用函数、最佳反应函数和目标函数等核心设计。首先，策略空间定义每个终端设备在特定时隙内可以选择的卸载策略集合，如完全卸载、部分卸载或不卸载。这些策略决定了设备在不同服务场景下进行任务处理的方式。

任意用户设备 v 通过 $c(v,t)$和 $a(v,t)$的组合形成策略组合，共有 3 种策略组合，即本地计算 s_1、部分卸载 s_2 和完全卸载 s_3，表示为 $S_v=\{0,1,2\}$，$\forall v \in V$，则所有用户设备的策略集合为 $S=S_1 \times S_2 \times \cdots \times S_v$。需要最小化目标函数为：

$$\Phi(s) = \sum_{v \in V} (\omega E_v(s) + (1-\omega) T_v(s)) \tag{35}$$

效用函数(即收益函数)用于量化每个设备在选择特定卸载策略后的系统表现，通常与能耗、时延等性能指标相关联。效用函数反映设备在采取某一策略后的“满意度”或“收益”，使设备能够评估不同策略的优劣。任意用户设备 v 的收益函数 R_v 可以表示为：

$$R_v = \omega E_v(s) + (1-\omega) T_v(s) \tag{36}$$

对于 v 而言，其目标是最小化其收益 minimize R_v。因此，可以形成博弈模型：

$$G = (V, \{S_i \parallel i \in V\}, \{R_i \parallel i \in V\}) \tag{37}$$

最佳反应函数描述每个设备在给定其他设备策略的情况下，如何选择自身的最优卸载策略，以最大化其效用或最小化

其能耗和时延。对于任意用户设备 v,在其他用户设备的策略固定时,其最佳反应函数为:

$$BR_v(s_{-v})=\arg\max_{s_v\in S_v} R_v(s_v,s_{-v}) \tag{38}$$

其中,s_{-v}表示其他用户设备的策略组合。因此,可以将收益函数写为:

$$R_v(s_v,s_{-v})=-(\omega E(s_v,s_{-v})+(1-\omega)T(s_v,s_{-v})) \tag{39}$$

通过迭代优化博弈模型求解目标函数,能够得到博弈的均衡解,从而实现整个系统性能的优化与提升。在达到 NE 的条件下,对于任意用户设备 v 都有:

$$R_v(BR_v(s_{-v}),s_{-v})\geqslant R_v(s_v,s_{-v}),\forall s_v\in S_v \tag{40}$$

在每一轮迭代中,所有用户设备根据当前策略更新自己的策略:

$$s_v^{k+1}=BR_v(s_{-v}^{(k)}) \tag{41}$$

其中,k 为迭代轮数。

策略更新继续进行,直到所有设备的策略不再改变,即收敛(达到 NE):

$$|s_v^{k+1}-s_v^k|<\varepsilon,\forall v(\text{收敛}) \tag{42}$$

使用博弈算法得到的解是基于 NE 条件下的每个用户设备的策略。在 NE 中,每个设备的策略都是在考虑到其他设备的策略后所做出的最优选择,即没有设备能够通过单方面改变自己的策略来提高其效用。需要注意的是,NE 并不一定代表全局最优解或最小值,它仅仅是基于特定的条件,每个设备都无法通过改变自身策略来进一步改善性能的状态,即 NE 体现的是博弈中的稳定性,而非最小化目标函数的最优值。但当式(35)为全局潜在函数时,则存在着至少一个 NE,并且最优解是其中一个 NE。

当某个用户设备改变其策略时,系统的全局潜在函数变化和该用户设备的效用函数变化一致,符合这个条件即为潜在函数[43]。因此,只需要证明对任意用户设备 v,效用函数的变化率 ΔR_v 和潜在函数的变化 $\Delta\Phi$ 是一致的即可。

假设式(35)为全局潜在函数,当用户设备 v 改变策略时,效用函数的变化为:

$$\Delta R_v=R_v(s_v',s_{-v})-R_v(s_v,s_{-v}) \tag{43}$$

代入式(39)的定义可得:

$$\Delta R_v=(1-\omega)\Delta E_v+\omega\Delta T_v$$

$$\Delta E_v=E_v(s_v',s_{-v})-E_v(s_v,s_{-v}) \tag{44}$$

$$\Delta T_v=T_v(s_v',s_{-v})-T_v(s_v,s_{-v})$$

其中,ΔE_v 和 ΔT_v 分别表示用户设备 v 的能耗与时延变化。潜在博弈函数的变化 $\Delta\Phi$ 来源于所有用户设备的能耗与时延的变化:

$$\Delta\Phi(s)=\sum_{v\in V}((1-\omega)\Delta E_v+\omega\Delta T_v) \tag{45}$$

由于只有用户设备 v 根据当前的情况改变策略,其他用户设备的 ΔE_v 和 ΔT_v 为 0,因此潜在函数变化只取决于用户设备 v 的变化,即:

$$\Delta\Phi(s)=(1-\omega)\Delta E_v+\omega\Delta T_v \tag{46}$$

由式(44)和式(46)可得:$\Delta\Phi(s)=\Delta R_v$ (47)

即式(35)为全局潜在函数,而潜在函数的存在是定义势博弈(Potential Game,PG)的关键,因此式(37)的博弈模型为 PG 模型[44]。在 PG 中,参与者的效用函数与潜在函数之间存在一一对应的关系,即当一个参与者改变其策略时,其效用函数的增减与潜在函数的增减保持一致。由式(47)可知,当用户设备 v 更新策略至 s_v^{k+1} 时,势函数变化满足:

$$\Phi(s^{k+1})-\Phi(s^k)=R(s_v^{k+1},s_{-v}^{k+1})-R(s_v^k,s_{-v}^k)\leqslant 0 \tag{48}$$

因 $\Delta E_v\geqslant 0$ 且 $\Delta T_v>0$,故 $\Delta\Phi(s)>0$,结合式(48)可知势函数非递增且有下界。基于这一性质,PG 在 NE 下至少存在一个解,并且该解对应的策略配置是全局最优的。因此,通过求解 PG 的 NE,能够确定最优的卸载策略,从而有效优化系统的能耗与时延。在此过程中,NE 不仅提供博弈的稳定解,也保证策略选择的最优性。基于 PG 的卸载策略算法如算法 2 所示。

算法 2 基于 PG 的卸载策略算法

输入:s,ω,ε,max_iter

输出:s^*

1. 初始化每个用户设备策略集合 $s_v^{(0)}$
2. WHILE iter < max_iter:
3. 每个用户设备 v 根据式(39)计算收益函数
4. 根据收益函数更新策略组合 s_v^{k+1}
5. 根据式(42)判断是否收敛
6. 若达到最大迭代次数 max_iter 或所有策略不再变化,END WHILE
7. 输出最终的 NE 策略集合 s^*
8. 从 NE 策略集合中选择最优 NE 解

5 仿真与结果分析

仿真实验基于一个 100 m×100 m 的矩形区域,UAV 在区域范围内为用户设备提供计算及通信服务,用户设备坐标随机生成。仿真参数的数值参考文献[11],具体如表 1 所列。

表 1 仿真参数

Table 1 Simulation parameters

参数说明	值
用户设备个数	200
用户设备生成的数据量 L	[1,10]Mbits
UAV 飞行高度 H	100 m
UAV 服务点个数	4
用户设备和 UAV 的移动通信范围	[100 m,100 m]
单位距离信道增益 β_0	−50 dB
用户设备的信道带宽 B_0	10 MHz
用户设备的传输功率 P	0.5 W
UAV 的噪声功率 σ^2	−70/(dBm/Hz)
用户设备的 CPU 电容系数 K_m	10^{-27}
UAV 的 CPU 电容系数 K_n	10^{-28}
在本地计算每比特所需要的 CPU 周期	800/(cycles/bit)
在 UAV 计算每比特所需要的 CPU 周期	1 000/(cycles/bit)
用户设备的计算资源 f_m	1 GHz
UAV 的计算资源 f_n	3 GHz
能耗和延迟的权重 ω	0.75

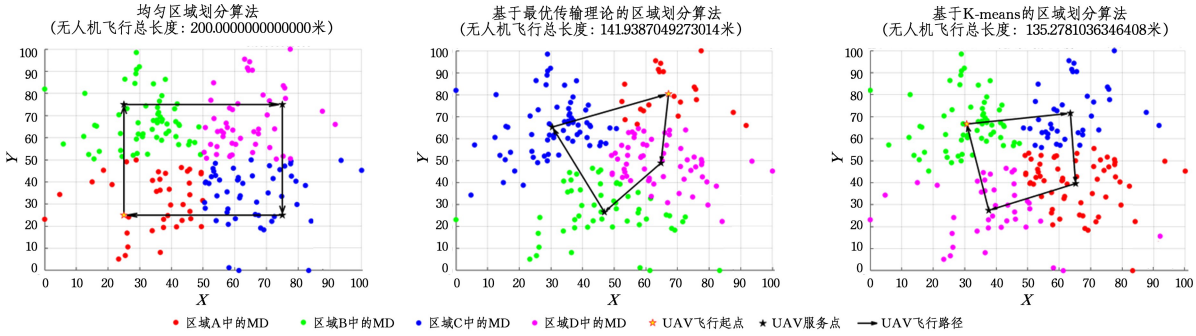
为验证基于 K-means 分簇的区域划分算法的优势,将其与其他两种划分区域算法进行对比分析,UAV 的飞行路径规划均采用本文的最短路径优化算法。具体对比算法如下。

1)均匀区域划分算法[14]:该算法将服务区域按照面积均等划分为若干子区域,UAV 在每个子区域的中心为用户设备提供计算和通信服务。

2)基于最优传输理论的区域划分算法[13]:该算法以优化系统性能为目标,基于最优传输理论对服务区域进行划分,

UAV 在每个子区域的中心为用户设备提供服务。

图 2 展示了在用户设备分布相同的情况下，不同算法的分区结果以及 UAV 的飞行路径。可以看出，3 种算法划分的区域存在显著差异，且 UAV 的飞行路径也各不相同。均匀区域划分算法未考虑用户设备的实际位置需求，导致划分后的 UAV 飞行路径较长，部分用户设备与 UAV 服务点的距离较远。基于最优传输理论的区域划分算法虽然考虑到系统性能，但 UAV 服务点与用户设备的距离以及飞行路径仍长于 K-means 分簇算法。相比之下，K-means 分簇算法充分考虑到了 UAV 与用户设备之间的距离，使得 UAV 服务点与用户设备的距离最短，且服务点相对集中，实现了最短的飞行路径。



(a)均匀区域划分算法 (b)基于最优传输理论的区域划分算法 (c)基于 K-means 的区域划分算法

图 2 不同分区算法分区与路径的对比

Fig. 2 Comparison of clustering and path planning under different partitioning algorithms

然而，飞行路径的长短仅能反映 UAV 的飞行能耗，而不能直接证明系统的整体能耗是否最优。图 3 进一步展示了在相同卸载策略算法(基于 PG 的卸载策略算法)下，不同分区算法的系统能耗与时延加权和。实验结果显示，当终端设备数量较少时，3 种区域划分算法的表现差异较小；随着终端设备数量的增加，基于最优传输理论的算法略优于均匀区域划分算法，而 K-means 分簇算法显著优于对比算法。通过对比可以发现，K-means 分簇算法不仅能够实现更短的飞行路径，还能在保持较低能耗的同时，有效降低系统时延。

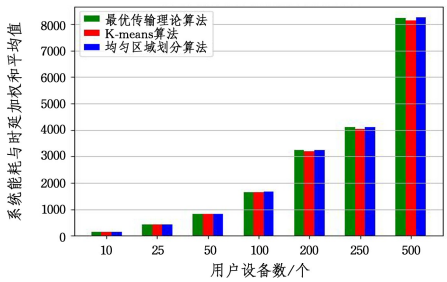


图 3 不同划分算法下系统的能耗与时延加权和

Fig. 3 Energy consumption and latency weighted sum under different partitioning algorithms

综上所述，基于 K-means 分簇的区域划分算法在区域划分中表现出显著的优势，能够在优化 UAV 飞行路径的同时，兼顾能耗和时延的平衡，为系统性能的提升提供有力支持。

为验证 CGO-2Opt 算法在解决 TSP 上的优势，将其与在 TSP 上应用较为广泛的算法进行对比分析。具体对比算法如下。

1)遗传算法结合 2-Opt(Genetic Algorithm and 2-Opt，GA-2Opt)[40]：遗传算法通过模拟进化，不断优化最短路径；同时结合 2-Opt 算法对路径进行局部优化，以进一步缩短路径长度。

2)蚁群优化结合 2-Opt(Ant Colony Optimization and 2-Opt，ACO-2Opt)算法[41]：蚁群优化算法通过模拟蚂蚁释放信息素和选择路径的行为，最终找到最优的旅行商路径；同时结合 2-Opt 算法对路径进行局部优化，以进一步缩短路径长度。

3)最近邻算法结合 2-Opt(Nearest Neighbor Algorithm and 2-Opt，NNA-2Opt)[42]：最近邻算法通过贪心策略快速构建初始策略，不断选择最近的未访问节点作为下一个节点，直至结束；同时结合 2-Opt 算法对路径进行局部优化，以进一步缩短路径长度。

图 4 和图 5 分别展示了不同算法在 TSP 中平均最短路径及平均算法运行时间的对比情况。由图 4 可知，节点个数较少时，CGO-2Opt 算法、ACO-2Opt 算法和 GA-2Opt 计算出的解结果较 NN-2Opt 表现更优异，随着节点个数的增加，CGO-2Opt 和 ACO-2Opt 表现依旧良好，GA-2Opt 解的质量逐渐不佳，而 NN-2Opt 可能因陷入局部最优，解的质量最低。由图 5 可知，随着节点个数的增加，GA-2Opt 和 ACO-2Opt 算法的运行时间快速增加，CGO-2Opt 的运行时间增长较为缓慢，NN-2Opt 的运行时间基本保持较低水平。综上所述，较其他算法，随着节点数量的增加，CGO-2Opt 算法能够有效解决 TSP，计算出最短飞行路径，同时在算法运行时间上也表现出色。

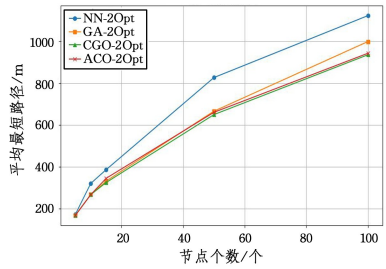


图 4 不同 TSP 算法的平均最短路径

Fig. 4 Average shortest path of different TSP algorithms

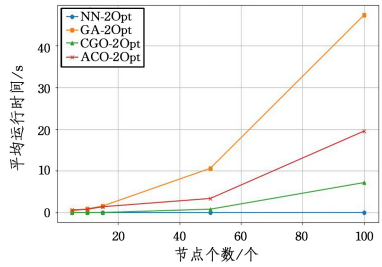


图 5 不同 TSP 算法的平均运行时间

Fig. 5 Average runtime of different TSP algorithms

图 6 展示了基于 PG 的卸载策略算法的收敛性能。实验结果显示，该算法在 6 次迭代内成功收敛。算法的快速收敛证明了其有效性，也进一步说明了在当前博弈框架下，系统能在有限的迭代次数下快速达到纳什均衡。

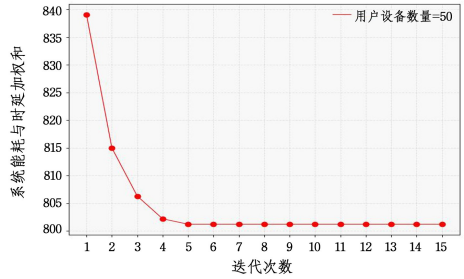


图 6 算法的收敛性

Fig. 6 Convergence of the proposed algorithm

为验证所提出的基于 PG 的卸载策略算法在该场景下的优势，将其与目前主流的卸载策略算法进行比较。

1）差分进化（Differential Evolution，DE）算法：采用改进差分进化算法优化终端设备的卸载策略[11]，以实现能耗和延迟加权和的最小化。

2）种群多样性-二元粒子群优化（Population Diversity Particle Swarm Optimization，PDPSO）算法[24]：以系统的能耗与延迟加权和作为优化目标，结合环境因素，采用 PDPSO 算法进行优化，以获得最佳的卸载策略。

3）遗传算法（Genetic Algorithm，GA）：采用基于改进遗传算法的任务决策算法为用户选择适合的卸载策略以及目标边缘服务器[18]，从而使系统的能耗与延迟加权和最小。

4）双深度 Q 网络（Double Deep Q-Network，Double DQN）算法[28]：通过深度强化学习优化任务卸载策略，利用双神经网络避免过估计，动态调整卸载决策，以最小化系统的时延和能耗加权和。

图 7 和图 8 分别展示了随着用户设备数量的增加，不同卸载策略算法计算出的系统能耗与延迟加权和平均值及其平均运行时间。可以看出，基于 PG 的卸载策略算法在系统能耗与延迟加权和平均值以及平均运行时间方面均表现出色，特别是在用户设备数量和节点个数较多的情况下，其优势更为明显。相比之下，随着终端设备的增多，Double DQN 算法系统能耗与延迟加权和平均值与 PG 算法相当，但运行时间显著长于 PG 算法。DE 和 PDPSO 算法在终端设备数量较少时的表现接近于 PG 算法；但随着用户设备数量的增加，它们的系统能耗与延迟加权和平均值以及平均运行时间显著增加。GA 在所有情况下的能耗与延迟加权和都较高，且平均运行时间也较长，表现相对较差。综上所述，随着用户设备数量的增加，PG 算法能够有效最小化系统能耗和时延，其在计算效率和计算时间方面具备优势。

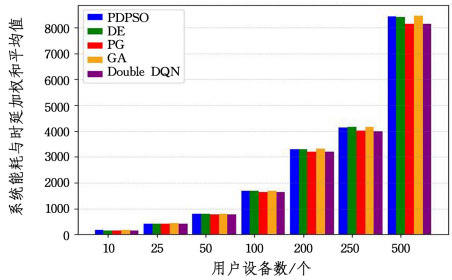


图 7 不同卸载策略算法下系统的能耗与延迟加权和

Fig. 7 Energy consumption and latency weighted sum under various offloading strategies algorithms

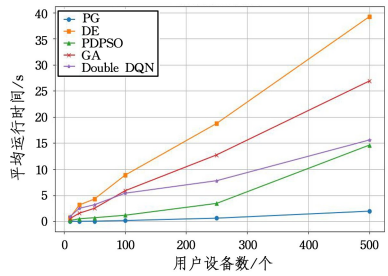


图 8 不同卸载策略算法的平均运行时间

Fig. 8 Average runtime of different offloading strategiesalgorithms

结束语 本文主要研究 UAV 辅助 MEC 系统中 UAV 的路径优化和用户设备计算任务卸载至 UAV 进行计算的问题。首先，为让所有终端设备都能够接收到优质服务，使用 K-means 算法将区域内的终端设备划分为多个簇，基于簇中心使用混沌博弈结合 2-Opt 的算法规划出 UAV 的飞行路径，在 UAV 服务范围覆盖全区域的同时延长 UAV 的生命时间。然后，将系统的能耗与时延和优化问题建模成势博弈模型，卸载策略作为博弈模型的策略空间，终端设备的能耗与时延加权和作为博弈模型效用函数，通过迭代得到 NE，并证明模型中至少存在一个 NE 对应着全局最优解。最后，通过仿真实验验证了算法的有效性与实用性。

尽管本研究的优化框架已取得显著性能提升，但是其扩展性仍受限于一些现实约束，未来研究将重点从以下方向展开：

1）考虑任务动态性（如突发任务生成与优先级机制）对资源分配的影响，探索实时任务调度与资源预留策略；

2）考虑 UAV 飞行高度动态调整对通信质量的影响，研究三维路径规划中的能耗与信道质量平衡机制；

3）优化分布式策略协调过程，考虑设备间的隐私保护需求与通信开销约束；

4）进一步将单 UAV 辅助扩展至多 UAV 协作场景，研究集群协同服务机制与空域资源优化分配问题。

参 考 文 献

[1] TALEB T, SAMDANIS K, MADA B, et al. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681.

[2] QU Y, DAI H, WANG H, et al. Service Provisioning for UAV-Enabled Mobile Edge Computing[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(11): 3287-3305.

[3] MACH P, BECVAR Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

[4] BAKTAYAN A A, THABIT Z A, AHMED A I. A Systematic Mapping Study of UAV-Enabled Mobile Edge Computing for Task Offloading[J]. IEEE Access, 2024, 12: 101936-101970.

[5] 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, 2021, 39(1): 131-141.

[6] MKIRAMWENI M E, YANG C, LI J, et al. A Survey of Game Theory in Unmanned Aerial Vehicles Communications[J]. IEEE Communications Surveys & Tutorials, 2019, 21(4): 3386-3416.

[7] CHI C, WANG Y, TONG X, et al. Game Theory in Internet of Things: A Survey[J]. IEEE Internet of Things Journal, 2022, 9(14): 12125-12146.

[8] LIN X, LIU A, HAN C, et al. LEO Satellite and UAVs Assisted Mobile Edge Computing for Tactical Ad-Hoc Network: A Game Theory Approach[J]. IEEE Internet of Things Journal, 2023, 10(23): 20560-20573.

[9] WANG M, ZHANG L, GAO P, et al. Stackelberg-Game-Based Intelligent Offloading Incentive Mechanism for a Multi-UAV-Assisted Mobile-Edge Computing System[J]. IEEE Internet of Things Journal, 2023, 10(17): 15679-15689.

[10] MAO Y, YOU C, ZHANG J, et al. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.

[11] XIANG K, HE Y. UAV-Assisted MEC System Considering UAV Trajectory and Task Offloading Strategy[C]// ICC 2023—IEEE International Conference on Communications. IEEE, 2023: 4677-4682.

[12] ASIM M, MASHWANI W K, BELHAOUARI S B, et al. A Novel Genetic Trajectory Planning Algorithm With Variable Population Size for Multi-UAV-Assisted Mobile Edge Computing System[J]. IEEE Access, 2021, 9: 125569-125579.

[13] LU Y, ZHOU H, WANG H, et al. Online Trajectory Optimization and Resource Allocation in UAV-Assisted NOMA-MEC Systems[C]// 2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS). IEEE, 2024: 1-2.

[14] ZHU Y, LIU S W, CHEN Q, et al. Task Offloading in UAV-assisted Mobile Edge Computing Based on Mixed-Strategy Game [J]. Aero Weaponry, 2024, 31(4): 112-120.

[15] WANG D, TIAN J, ZHANG H, et al. Task Offloading and Trajectory Scheduling for UAV-Enabled MEC Networks: An Optimal Transport Theory Perspective[J]. IEEE Wireless Communications Letters, 2022, 11(1): 150-154.

[16] EI N N, ALSENWI M, TUN Y K, et al. Energy-Efficient Resource Allocation in Multi-UAV-Assisted Two-Stage Edge Computing for Beyond 5G Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 16421-16432.

[17] LI Q Y, PENG B, LI Q, et al. A Latency-Optimal Task Offloading Scheme Using Genetic Algorithm for DAG Applications in Edge Computing[C]// 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, 2023: 344-348.

[18] SONG H, GU B, SON K, et al. Joint Optimization of Edge Computing Server Deployment and User Offloading Associations in Wireless Edge Network via a Genetic Algorithm[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(4): 2535-2548.

[19] DENG Y, PENG A. A DRL-Based Intelligent Task Offloading and Caching Strategy for IIoT in MEC[C]// 2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE). IEEE, 2024: 1347-1351.

[20] ALMELU S, VEENADHARI S. Task Offloading Strategy using Double Q-Learning based Optimization in MEC[C]// 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET). IEEE, 2022: 1-5.

[21] HU N, XIANG M, HUANG C H, et al. An Efficient Computing Task Offloading Strategy Based on Energy Consumption and Load Balancing Degree[C]// 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2022: 860-866.

[22] WANG Z H, WANG G C. Multi-objective Optimization of D2D Collaborative MEC Based on Improved NSGA-III[J]. Computer Science, 2024, 51(3): 280-288.

[23] FANG J, YE Z, SONG S. Research on Task Offloading Strategy Based on Priority Chemical Reaction Algorithm in Edge-Cloud Scenario[C]// 2022 11th International Conference on Communications, Circuits and Systems (ICCCAS). IEEE, 2022: 307-312.

[24] GAN Y, HE Y. Trajectory Optimization and Computing Offloading Strategy in UAV-Assisted MEC System[C]// 2021 Computing, Communications and IoT Applications (ComComAp). IEEE, 2021: 132-137.

[25] SUM F, WANG G J, ZHOU C, et al. A heuristic task offloading approach with delay and energy constraints for edge-cloud collaboration[J]. Journal of Jilin University (Engineering and Technology Edition), 2025, 55(5): 1648-1663.

[26] WANG Z, ZHU Q. Partial Task Offloading Strategy Based on Deep Reinforcement Learning[C]// 2020 IEEE 6th International Conference on Computer and Communications (ICCC). IEEE, 2020: 1516-1521.

[27] ZHANG P, GAN P, CHANG L, et al. DPRL: Task Offloading Strategy Based on Differential Privacy and Reinforcement Learning in Edge Computing[J]. IEEE Access, 2022, 10: 54002-54011.

[28] MING Z,GOU Q,YU H,et al. Deep Reinforcement Learning-Based Task Offloading Over In-Network Computing and Multi-Access Edge Computing[C]// 2023 International Conference on Networking and Network Applications(NaNA). IEEE,2023:281-286.

[29] LI X C,SUN H Y,XU C. Network resource allocation method of aerospace edge computing based on deep Q network algorithm [J]. Journal of Jilin University(Engineering and Technology Edition),2025,55(7):2418-2424.

[30] HE J. Optimization of Edge Delay Sensitive Task Scheduling Based on Genetic Algorithm[C]// 2022 International Conference on Algorithms,Data Mining,and Information Technology (ADMIT). IEEE,2022:155-159.

[31] RAJWAR D,KUMAR D. A PSO based Computation Offloading Model in Edge Computing[C]// 2024 IEEE International Conference on Contemporary Computing and Communications (InC4). IEEE,2024:1-6.

[32] DU Z,WANG X,WANG S,et al. Toward UAV Communication Network Self-Organization:A Game-Theoretic Distributed Control Approach[C]// 2022 IEEE International Conference on Unmanned Systems(ICUS). IEEE,2022:121-127.

[33] SUN Z,WANG Y,CHEN G,et al. StackelbergGame-Based Task Offloading Strategy for Multi-Users[C]// 2021 International Conference on Electronic Information Engineering and Computer Science(EIECS). IEEE,2021:674-677.

[34] XU Y,XIAO M,WU J,et al. A Personalized Privacy Preserving Mechanism for Crowdsourced Federated Learning[J]. IEEE Transactions on Mobile Computing,2024,23(2):1568-1585.

[35] XU Y,XIAO M,ZHU Y J,et al. AoI-Guaranteed Incentive Mechanism for Mobile Crowdsensing With Freshness Concerns [J]. IEEE Transactions on Mobile Computing,2024,23(5):4107-4125.

[36] SUN H,ZHOU Y,ZHANG H,et al. Joint Optimization of Caching,Computing,and Trajectory Planning in Aerial Mobile Edge Computing Networks:An MADDPG Approach[J]. IEEE Internet of Things Journal,2024,11(24):40996-41007.

[37] TAKASHIMA Y,NAKAMURA Y. Theoretical and Experimental Analysis of Traveling Salesman Walk Problem[C]// 2021 IEEE Asia Pacific Conference on Circuit and Systems (APCCAS). IEEE,2021:241-244.

[38] TALATAHARI S,AZIZI M. Chaos Game Optimization:a novel metaheuristic algorithm[J]. Artificial Intelligence Review,2021,54:917-1004.

[39] WANG Y,WANG X. Analysis of Lyapunov Chaos in Game Dynamics[C]// 2022 International Conference on Machine Learning,Cloud Computing and Intelligent Mining(MLCCIM). IEEE,2022:470-477.

[40] THONGPIEM J,PUNKONG N,RATANAVILISAGUL C,et al. HybridGenetic Algorithm and Ant Colony Algorithm for Solving Travelling Salesman Problem[C]// 2024 IEEE 9th International Conference on Computational Intelligence and Applications(ICCIA). IEEE,2024:69-73.

[41] ZHANG Y,WANG C,LI H,et al. An Improved 2-Opt and ACO Hybrid Algorithm for TSP[C]// 2018 Eighth International Conference on Instrumentation & Measurement,Computer,Communication and Control(IMCCC). IEEE,2018:547-552.

[42] NURAIMAN D,ILAHI F,DEWI Y,et al. A New Hybrid Method Based on Nearest Neighbor Algorithm and 2-Opt Algorithm for Traveling Salesman Problem[C]// 2018 4th International Conference on Wireless and Telematics(ICWT). IEEE,2018:1-4.

[43] WANG T,YOU C. Adaptive Uplink Scheduling and UAV Association in UAV-Assisted OFDMA Cellular Networks: A Game-Theoretical Approach[J]. IEEE Access,2024,12:63504-63514.

WEI Manyi, born in 2000, postgraduate. Her main research interests include computer network and mobile edge computing.

WANG Gaocai, born in 1976, Ph.D, professor, doctoral supervisor. His main research interests include computer network, performance evaluation and network security.

(责任编辑:柯颖)