基于深度强化学习的空地协同群智感知任务分配优化方法

刘晓亮[1,2]，蒋伟进[3]，王 刚[4†]

(1. 中南大学 商学院，长沙 410083；2. 湘江实验室，长沙 410083；
3. 湖南工商大学 计算机学院，长沙 410205；4. 湖南工商大学 管理科学与工程学院，长沙 410205)

摘 要: 空地协同移动群智感知 (Air-Ground Collaborative Mobile Crowd Sensing, AG-MCS) 融合无人机高机动覆盖能力与地面用户细粒度感知优势, 可为复杂城市感知任务提供协同感知方式. 然而, 任务动态到达、执行者异构性强及多维时空约束耦合等特性, 使现有任务分配方法难以在大规模场景下兼顾系统效用与服务质量. 为此, 本文提出一种基于深度强化学习的空地协同任务分配方法 (DRL-CTA). 该方法构建包含异构执行者能力与多重约束的协同感知环境, 将动态任务分配问题建模为马尔可夫决策过程; 进一步设计基于 Actor-Critic 架构的深度策略网络, 引入多头注意力机制刻画任务与执行者间的时空关联, 并结合优先经验回放机制提升训练效率. 在不同任务规模与动态到达率场景下, DRL-CTA 在任务完成率与执行者利用率上优于多种基线方法, 在平均响应时间上优于多数基线方法, 并在大规模高负载场景下保持领先. 实验结果表明, DRL-CTA 能够改善异构执行者协同与动态任务分配性能, 为低空经济场景下的智慧城市感知任务提供可扩展的优化方法.

关键词: 移动群智感知；空地协同；任务分配；深度强化学习；多头注意力机制；优先经验回放

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2026.0452

引用格式: 刘晓亮, 蒋伟进, 王刚. 基于深度强化学习的空地协同群智感知任务分配优化方法 [J]. 控制与决策, xxxx, x(x): xxxx-xxxx.

Deep reinforcement learning-based optimization method for air-ground collaborative crowd sensing task allocation

LIU Xiao-liang[1,2]，*JIANG Wei-jin*[3]，*WANG Gang*[4†]

(1. Business School，Central South University，Changsha 410083，China；2. Xiangjiang Laboratory，Changsha 410083，China；3. School of Computer Science，Hunan University of Technology and Business，Changsha 410205，China；4. School of Management Science and Engineering，Hunan University of Technology and Business，Changsha 410205，China)

Abstract: Air-Ground Collaborative Mobile Crowd Sensing (AG-MCS) combines the high-mobility coverage of Unmanned Aerial Vehicles (UAVs) with the fine-grained sensing capability of ground users, providing a collaborative sensing approach for complex urban tasks. However, dynamic task arrivals, executor heterogeneity, and coupled spatio-temporal constraints make it difficult for existing task allocation methods to balance system utility and service quality in large-scale scenarios. To address this problem, this paper proposes a Deep Reinforcement Learning-based Collaborative Task Allocation method (DRL-CTA). The method constructs a collaborative sensing environment with heterogeneous executor capabilities and multiple constraints, and formulates dynamic task allocation as a Markov Decision Process (MDP). It further designs an Actor-Critic policy network that uses Multi-Head Attention to model spatio-temporal correlations between tasks and executors, together with Prioritized Experience Replay to improve training efficiency. Experiments under different task scales and dynamic arrival rates show that DRL-CTA outperforms several baseline methods in task completion rate and executor utilization, and outperforms most baseline methods in average response time, maintaining its advantage in large-scale high-load scenarios. The results indicate that DRL-CTA improves heterogeneous executor coordination and dynamic task allocation, providing a scalable optimization method for smart city sensing in low-altitude economy applications.

Keywords: mobile crowd sensing；air-ground collaboration；task allocation；deep reinforcement learning；multi-head attention；prioritized experience replay

收稿日期: 2026-05-15；**录用日期:** 2026-11-02.
基金项目: 国家自然科学基金项目 (52408364)；长沙市自然科学基金项目 (kq2502069).
责任编委: 许斌.
†通信作者. E-mail: 3045@hutb.edu.cn.

0 引 言

随着低空经济发展和无人机技术进步, 空地协同感知系统逐渐成为智慧城市建设的重要组成部分[1,2]. 无人机 (UAV) 凭借三维移动、广覆盖和快速响应优势, 已应用于灾害监测、交通管理、环境监测等领域[3]; 移动群智感知 (MCS) 利用地面用户的便携设备构建大规模感知网络, 具有成本低、精度高、部署灵活等特点[4]. 如何协调这两类异构执行者、实现感知资源的优化配置, 是智慧城市领域的重要问题[5,6].

本文将无人机与地面用户协同参与的移动群智感知范式称为空地协同移动群智感知 (Air-Ground Collaborative Mobile Crowd Sensing, AG-MCS). AG-MCS 通过空中平台的快速覆盖与机动性优势, 结合地面参与者的细粒度、低成本感知能力, 实现对复杂城市任务的协同数据采集与在线服务保障. 其整体工作流程与分层架构如图 1 所示.

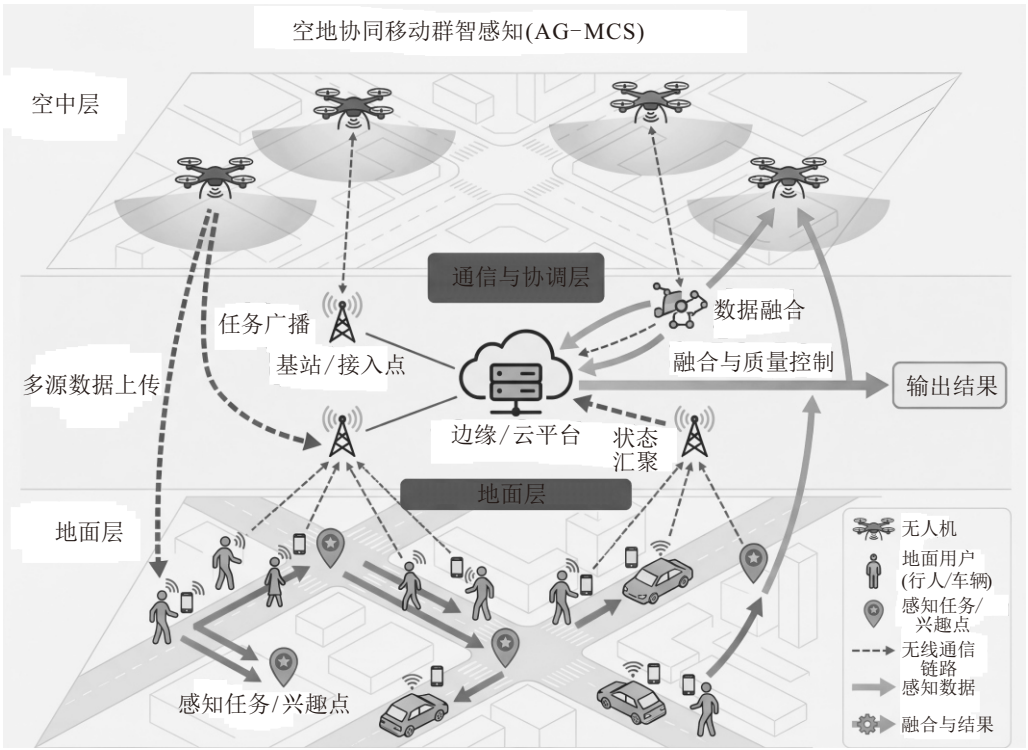


图 1 空地协同移动群智感知 (AG-MCS) 模式示意图

近年来, 异构多执行者系统协同优化受到持续关注[7,8]. 传统优化方法 (整数规划、贪心、启发式) 在小规模问题上表现良好, 但面对动态多约束耦合问题存在计算复杂度高、实时性差、适应性弱等局限; 尤其在异构执行者场景中, 移动能力、感知精度、能耗特性差异明显, 传统方法难以获得全局较优策略[9].

深度强化学习 (DRL) 为解决复杂决策问题提供新范式[10,11], 已成功应用于路径规划、资源调度和协同控制等任务[12], 但应用于空地协同任务分配仍面临大规模状态-动作空间、异构执行者协调、实时性等挑战[11].

低空无人机与地面用户协同的任务分配问题兼具异构执行者协同、时空耦合约束、多目标权衡、动态适应与组合复杂度高等特征[5,13,14,15].

本研究从理论建模、算法设计和应用验证三个层面展开. 理论层面, 将异构多智能体协同优化与 DRL 理论结合, 建立基于 MDP 的统一数学框架[16]; 算法层面, 设计融合多头注意力与优先经验回放的 Actor-Critic 算法[17,18]; 应用层面, 为智慧城市应急管理、智能交通监控、环境监测等场景提供任务分配方案[1,19].

1 文献综述

1.1 移动群智感知任务分配研究

移动群智感知 (Mobile Crowd Sensing, MCS) 利用移动设备用户的感知能力完成大规模感知任务, 任务分配是其中核心环节. 在系统效用优化方面, PSAllocator 框架考虑感知能力约束, 通过多任务分配优化系统效用[20], 但难以处理动态环境和大规模问题; iCrowd 框架采用 k-depth 覆盖度量和双重优化目标, 在机会主义模式下实现近似最优分配, 但计算复杂度较高[21].

在混合模式与质量感知方面, Wang 等提出的 HyTasker 框架将机会主义模式和参与式模式相结合, 通过两阶段混合优化提高任务完成率[22], 但未考

虑异构执行者协同场景. Wei 等研究了数据质量感知的任务分配, 在预算约束下优化质量覆盖, 但主要面向单一类型执行者[23]. 蒋伟进等针对移动群智感知中的任务质量需求, 研究了多任务分发场景下的参与者选择问题[24]; 杨潇等进一步面向异构群智感知场景, 采用 PPO 方法处理多目标任务指派问题[25]. 在时空约束与路径规划方面, 已有研究针对时间约束下的任务分配问题进行了探索, 从时限约束方面提升任务可达性[26]; 刘娅汐等将无人机时空众包资源分配引入工业场景, 体现了无人机辅助群智感知任务分配的应用扩展[27].

1.2　深度强化学习在任务分配中的应用

深度强化学习 (Deep Reinforcement Learning, DRL) 结合了深度学习的表示能力和强化学习的决策能力, 在复杂优化问题中表现出色[10]. 近年来, DRL 在任务分配问题中的应用逐渐增多. 相关研究已将多机器人任务分配与导航过程联合建模为端到端强化学习问题, 并将多无人机集群任务指派建模为多智能体深度强化学习问题[19,28]. 与此同时, 在无人机辅助移动群智感知场景中, 数据采集与资源配置的协同优化也逐渐受到关注[6]. 面向移动群智感知中的隐私约束, 杨明川等构建了考虑隐私保护的深度强化学习任务分配模型[29]; 在紧急任务场景中, 张华卿等研究了多智能体批次任务分配问题, 为动态任务到达和多执行者协同提供了近期参考[30]. Actor-Critic 架构作为 DRL 的重要分支, 通过策略网络和价值网络的双网络结构实现策略梯度与价值评估的结合, 在连续控制和离散决策问题中均取得了良好效果[12].

注意力机制方面, Transformer 架构[17] 通过多头注意力捕获长程依赖, 为处理大规模状态-动作空间提供有效工具. 经验回放通过重用历史经验提高样本效率[10]; 优先经验回放[18] 进一步引入优先级机制, 优先学习 TD 误差较大的重要经验, 加速收敛.

现有研究存在以下不足: (1) 异构执行者协同研究不足: 现有方法主要面向单一类型执行者, 缺乏针对无人机与地面用户协同场景的专门研究; (2) 实时性要求难以满足: 传统优化算法计算复杂度高, 难以满足实时任务分配的时效性要求; (3) 动态适应能力弱: 现有方法难以适应动态变化的环境, 无法实时调整分配策略; (4) 大规模问题求解困难: 随着问题规模增大, 传统算法的求解效率急剧下降.

2　问题描述与建模

2.1　问题背景与挑战

低空无人机与地面用户协同任务分配的核心难点在于协调两类能力差异明显的执行者, 并在多重约束下完成在线决策. 无人机具备三维机动能力和广域覆盖优势, 但受续航限制; 地面用户具备精细感知和持续驻守能力, 但活动范围有限. 这种异构性使统一分配策略难以直接套用, 也要求模型同时考虑协同收益、个体能力和任务约束[31].

任务时空窗口、完成率、响应时间与能耗成本相互耦合, 且任务需求与执行者状态持续变化, 要求算法兼顾快速响应与计算开销控制.

2.2　系统模型定义

2.2.1　执行者模型

考虑异构执行者集合$\mathcal{W}=\mathcal{W}_U\cup\mathcal{W}_G$, 其中$\mathcal{W}_U=\{w_1^u,\ldots,w_{|\mathcal{W}_U|}^u\}$表示无人机集合, $\mathcal{W}_G=\{w_1^g,\ldots,w_{|\mathcal{W}_G|}^g\}$表示地面用户集合. 每个执行者$w_i\in\mathcal{W}$定义为五元组:

$$w_i=\langle p_i(t),v_i,S_i,r_i,b_i(t)\rangle. \tag{1}$$

式中, $p_i(t)$为时刻t的二维位置, v_i为移动速度, S_i为传感器类型集合, $r_i\in[0,1]$为可靠性, $b_i(t)\in[0,1]$为剩余电量 (仅无人机有效, 地面用户记为 1).

2.2.2　任务模型

任务集合定义为$\mathcal{T}=\{t_1,\ldots,t_{|\mathcal{T}|}\}$, 每个任务$t_j\in\mathcal{T}$描述为六元组:

$$t_j=\langle l_j,e_j,d_j,S_j^{req},r_j^{req},R_j\rangle. \tag{2}$$

式中, $l_j=(x_j,y_j)$为目标位置, e_j、d_j为最早开始与最晚完成时间, S_j^{req}为所需传感器集合, $r_j^{req}\in[0,1]$为最低可靠性要求, R_j为基础奖励.

2.3　马尔可夫决策过程建模

为将任务分配问题转化为可计算形式, 本文采用马尔可夫决策过程 (MDP) 进行建模. MDP 由五元组$\langle\mathcal{S},\mathcal{A},P,r,\gamma\rangle$定义.

2.3.1　状态空间

状态空间$\mathcal{S}$包含任务状态、执行者状态和环境状态的综合信息, 即$s_t=\langle T_t,W_t,E_t\rangle\in\mathcal{S}$.

设系统共有K类传感器, 任务需求集合S_j^{req}与执行者传感器集合S_i分别映射为多热向量$\mathbf{q}_j,\mathbf{u}_i\in\{0,1\}^K$(第$k$位为 1 表示包含第$k$类传感器).

1) 任务状态矩阵$T_t\in\mathbb{R}^{|\mathcal{T}|\times d_t}$. 其第$j$行编码任务$t_j$的动态特征:

$$T_t[j]=[\frac{x_j}{L},\frac{y_j}{L},\frac{d_j-t}{d_{\max}},\frac{R_j}{R_{\max}},r_j^{req},\frac{t-e_j}{d_j-e_j},\mathbf{q}_j,\mathrm{flag}_j]. \tag{3}$$

式中, L为区域边长, $d_{\max}$、$R_{\max}$为最大截止时间与最大奖励, t为当前时间; 第六维表示时间紧迫性; $\mathrm{flag}_j=(\mathrm{is_pending}_j,\mathrm{is_completed}_j,\mathrm{is_failed}_j)$ 为

任务状态标志三元组, 故$d_t = 6 + K + 3$.

2) 执行者状态矩阵$W_t \in \mathbb{R}^{|\mathcal{W}|\times d_w}$. 其第$i$行编码执行者$w_i$的实时状态:

$$W_t[i] = [\frac{x_i(t)}{L}, \frac{y_i(t)}{L}, \frac{v_i}{v_{\max}}, r_i, b_i(t), \text{type}_i, \mathbf{u}_i]. \quad (4)$$

式中, $v_{\max}$为最大速度, $\text{type}_i \in \{0,1\}$为执行者类型标识 (0 表示无人机, 1 表示地面用户), 因此$d_w = 6 + K$.

3) 环境状态向量$E_t \in \mathbb{R}^3$, 编码全局环境信息:

$$E_t = [\frac{t}{d_{\max}}, \frac{|\mathcal{T}_{pending}|}{|\mathcal{T}|}, \lambda_{arrival}]. \quad (5)$$

式中, $|\mathcal{T}_{pending}|$为当前待处理任务数, $\lambda_{arrival}$为任务到达率 (静态取 0). 状态$s_t = \langle T_t, W_t, E_t\rangle$综合任务动态、执行者位置与电池变化, 满足马尔可夫性.

2.3.2 动作空间

动作空间$\mathcal{A}$定义为任务分配决策的集合$a_t = \langle a_t[1], \ldots, a_t[|\mathcal{W}|]\rangle$. 为与网络输出保持一致, 令$a_t[i] \in \{0, 1, \ldots, |\mathcal{T}|\}$表示执行者$w_i$的分配决策: 当$0 \leqslant a_t[i] < |\mathcal{T}|$时表示分配任务$t_{a_t[i]}$, 当$a_t[i] = |\mathcal{T}|$时表示保持空闲. 因此联合动作空间大小为$(|\mathcal{T}| + 1)^{|\mathcal{W}|}$. 由于直接枚举联合动作在大规模场景下不可行, 本文采用按执行者分解的方式生成候选联合动作$u_t \sim \pi_\theta(\cdot|s_t)$, 经动作掩码$\mathbf{M}(s_t)$限定可行支持集后, 再由确定性冲突消解算子$\Psi$输出实际执行动作$\hat{a}_t = \Psi(s_t, u_t)$, 环境转移记为$P_\Psi(s_{t+1}|s_t, u_t)$. 经验回放存储候选动作$u_t$, 环境状态更新使用实际动作$\hat{a}_t$, 二者接口明确, 策略分布$\pi_\theta$不被$\Psi$改变.

2.3.3 奖励函数

为引导算法学习最优策略, 设计如下多目标综合奖励函数:

$$\begin{aligned} r(s_t, a_t, s_{t+1}) = &\sum_{j\in\mathcal{T}_{completed}} (R_j + R_j^{time} + \\ &R_j^{priority}) f_{time}(j) - \lambda_m \sum_{i\in\mathcal{W}} c_i^{move}(a_t[i]) - \\ &\lambda_f \sum_{j\in\mathcal{T}_{failed}} P_j - \lambda_c \sum_{j\in\mathcal{T}_{conflict}} P_j^{conflict}. \end{aligned} \quad (6)$$

其中$\lambda_m, \lambda_f, \lambda_c \geqslant 0$分别为移动成本、失败任务和冲突任务的惩罚权重, 用于避免与折扣因子γ混淆. 对完成任务t_j, 记其完成时刻为$t_{complete}^j$, 时间效率因子和时间奖励定义为

$$\begin{aligned} f_{time}(j) &= \max\{0, 1 - \frac{t_{complete}^j - e_j}{d_j - e_j}\}, \\ R_j^{time} &= \rho_t R_j f_{time}(j). \end{aligned} \quad (7)$$

其中$\rho_t \geqslant 0$为时间奖励系数; 优先级奖励$R_j^{priority} = \rho_p \omega_j R_j$($\omega_j \in [0,1]$为外部优先级, 未设置时取 0). 移动成本按归一化距离计算:

$$c_i^{move}(a_t[i]) = \begin{cases} \|p_i(t) - l_j\|_2 / L, & a_t[i] = j, \\ 0, & a_t[i] = |\mathcal{T}|. \end{cases} \quad (8)$$

其中$0 \leqslant j < |\mathcal{T}|$; 失败惩罚与冲突惩罚均取$P_j = R_j$, 分别对应超时未完成与多执行者竞争同一任务的无效决策.

2.3.4 约束条件

分配方案必须严格满足以下多重约束, 以保障任务的可行性与系统的安全性:

(C1) 时空窗口约束: 执行者w_i到达任务t_j的预计时刻$t_{arrival} = t + \|p_i(t) - l_j\|_2 / v_i$须位于任务有效时间窗口内, 即

$$t_{arrival} \leqslant d_j. \quad (9)$$

确保任务在截止时间前完成.

(C2) 能力匹配约束: 执行者须具备完成任务所需的全部传感器类型, 且可靠性等级不低于任务要求, 即

$$S_j^{req} \subseteq S_i, \ r_i \geqslant r_j^{req}. \quad (10)$$

(C3) 电池容量约束 (仅对无人机有效): 无人机的剩余电量$b_i(t)$随移动能耗持续更新, 并作为状态特征参与网络决策.

(C4) 空间可达性约束: 执行者到达任务的可达性统一由约束 (C1) 的时间窗口判定覆盖; 地面用户与无人机的可达性差异体现为二者移动速度区间不同 (地面用户$2 \sim 8$单位/时间步, 无人机$10 \sim 20$单位/时间步).

上述约束在决策阶段通过动作掩码实现. 对任意执行者-任务对(w_i, t_j), 若其不满足时空窗口 (C1) 或能力匹配 (C2) 条件, 则令对应掩码$M_{ij} = 0$, 使该不可行动作在策略输出中被排除; 仅当满足 (C1)、(C2) 时, $M_{ij} = 1$. (C3)、(C4) 在当前实现中未纳入掩码判定, 其局限性在第 6 节讨论.

2.4 复杂度分析

2.4.1 计算复杂度

任务分配问题的决策空间大小为$|\mathcal{A}| = (|\mathcal{T}| + 1)^{|\mathcal{W}|}$. 随着任务数$|\mathcal{T}|$和执行者数$|\mathcal{W}|$的增长, 解空间呈指数级扩展, 传统精确算法难以在多项式时间内求解.

2.4.2 理论性质

定理 1 低空无人机与地面用户协同的任务分配问题属于 NP-hard 问题.

证明 将经典的多目标多约束指派问题 (Generalized Assignment Problem, GAP) 多项式时间

归约为本文问题 (即GAP $\leq_p$ CTA) 来证明. 给定任意 GAP 实例, 可在多项式时间内构建对应的协同任务分配实例: 执行者对应 GAP 中的"代理", 任务对应"作业", 执行者资源对应容量限制. 由于 GAP 已被证明是 NP-hard 问题[15], 若本文问题存在多项式时间算法, 则该算法亦可求解 GAP, 故本文研究问题亦为 NP-hard.

推论 1　除非 P=NP, 否则不存在多项式时间的精确算法能够求解所有规模的任务分配实例.

2.5　强化学习求解框架

鉴于问题的 NP-hard 性质和动态环境特征, 本文采用深度强化学习方法求解. 通过参数化价值函数和策略函数, 利用神经网络逼近最优策略. 价值函数$V^{\pi}(s_t)$和动作价值函数$Q^{\pi}(s_t,a_t)$定义如下:

$$V^{\pi}(s_t)=\mathbb{E}_{\pi}[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\mid s_t] \tag{11}$$

$$Q^{\pi}(s_t,a_t)=\mathbb{E}_{\pi}[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\mid s_t,a_t] \tag{12}$$

式中$\gamma\in[0,1]$为折扣因子. 通过最大化期望累积回报, 学习最优策略$\pi^*=\arg\max_{\pi}V^{\pi}(s)$, 实现任务分配的智能化决策.

3　DRL-CTA 算法设计

3.1　算法总体框架

DRL-CTA 算法采用 Actor-Critic 架构, 通过策略网络 (Actor) 和价值网络 (Critic) 的协同学习, 实现任务分配策略的优化. 算法框架如算法 1 所示.

Algorithm 1　DRL-CTA算法框架

Require: 环境$\mathcal{E}$, 训练轮数M, 探索参数ϵ
Ensure: 最优策略网络π^*

初始化Actor网络π_θ和Critic网络Q_ϕ
初始化目标网络$\pi_{\theta'}$和$Q_{\phi'}$($\theta'\leftarrow\theta,\phi'\leftarrow\phi$)
初始化经验回放缓冲区$\mathcal{D}$
for episode $m=1$ to M **do**
　初始化状态s_0
　for step $t=0$ to T **do**
　　计算动作掩码$\mathbf{M}(s_t)$, 依策略生成候选动作$u_t\sim\pi_\theta(\cdot|s_t)$
　　冲突消解: 实际动作$\hat{a}_t=\Psi(s_t,u_t)$
　　执行$\hat{a}_t$, 观察奖励r_t与下一状态s_{t+1}
　　存储经验(s_t,u_t,r_t,s_{t+1})到$\mathcal{D}$
　　if $|\mathcal{D}|>$ batch_size **then**
　　　从$\mathcal{D}$中采样批次经验
　　　更新Critic: $\phi\leftarrow\phi-\alpha_\phi\nabla_\phi L_Q$
　　　更新Actor: $\theta\leftarrow\theta-\alpha_\theta\nabla_\theta L_{\text{actor}}$
　　　软更新目标网络: $\theta'\leftarrow\tau\theta+(1-\tau)\theta'$, $\phi'\leftarrow\tau\phi+(1-\tau)\phi'$
　　end if
　　$s_t\leftarrow s_{t+1}$
　end for
end for
return π_θ

3.2　网络架构设计

3.2.1　Actor 网络 (策略网络)

Actor 网络$\pi_\theta(s_t)$负责根据当前状态输出动作概率分布. 网络结构包括:

任务编码器: 将任务状态矩阵T_t编码为任务特征向量:

$$h_t^{task}=\mathrm{ReLU}(W_2\cdot\mathrm{ReLU}(W_1\cdot T_t^T+b_1)+b_2). \tag{13}$$

其中$T_t\in\mathbb{R}^{|\mathcal{T}|\times d_t}$为任务状态矩阵, $W_1\in\mathbb{R}^{d_h\times d_t}$, $W_2\in\mathbb{R}^{d_h\times d_h}$为权重矩阵, b_1, b_2为偏置向量, d_h为隐藏层维度.

执行者编码器: 将执行者状态矩阵W_t编码为执行者特征向量:

$$h_t^{agent}=\mathrm{ReLU}(W_4\cdot\mathrm{ReLU}(W_3\cdot W_t^T+b_3)+b_4). \tag{14}$$

其中$W_t\in\mathbb{R}^{|\mathcal{W}|\times d_w}$为执行者状态矩阵, $W_3\in\mathbb{R}^{d_h\times d_w}$, $W_4\in\mathbb{R}^{d_h\times d_h}$为权重矩阵, b_3, b_4为偏置向量.

多头注意力机制: 通过多头注意力机制学习任务和执行者之间的匹配关系[17]:

$$\mathrm{Attention}(Q,K,V)=\mathrm{Softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{15}$$

其中Q, K, V分别为查询、键和值矩阵, d_k为键的维度.

对于多头注意力, 定义H个注意力头:

$$\mathrm{MultiHead}(Q,K,V)=\mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_H)W^O. \tag{16}$$

其中$\mathrm{head}_i=\mathrm{Attention}(QW_i^Q,KW_i^K,VW_i^V)$, W_i^Q, W_i^K, W_i^V和W^O为可学习的权重矩阵.

将任务特征矩阵$h_t^{task}\in\mathbb{R}^{|\mathcal{T}|\times d_h}$和执行者特征矩阵$h_t^{agent}\in\mathbb{R}^{|\mathcal{W}|\times d_h}$分别通过多头注意力机制处理, 得到增强的任务特征$\tilde{h}_t^{task}$和执行者特征$\tilde{h}_t^{agent}$.

多尺度特征融合: 为进一步捕获不同粒度的局部结构, 对注意力增强后的特征施加多尺度融合模块 (MultiScaleFusion), 采用三路并行一维卷积 (卷积核 1、3、5) 提取不同感受野特征, 拼接后经线性投影压缩至原始维度, 并以残差连接与层归一化保持训练稳定性:

$$\mathrm{MSF}(h)=\mathrm{LN}(h+\mathrm{Proj}(\mathrm{Concat}(f_1,f_3,f_5))). \quad (17)$$

其中$f_k(h)$为卷积核大小为k的一维卷积. 该设计使网络同时感知单个实体精细特征 ($k=1$) 与相邻实体上下文关联 ($k=3,5$), 增强任务-执行者空间分布的表征能力.

匹配分数计算: 计算每个执行者-任务对的匹配分数. 对于执行者w_i和任务t_j, 匹配分数通过点积计算:

$$s_{ij}=\tilde{h}_t^{agent}[i]\cdot\tilde{h}_t^{task}[j]. \quad (18)$$

其中$\tilde{h}_t^{agent}[i]$表示第i个执行者的特征向量, $\tilde{h}_t^{task}[j]$表示第j个任务的特征向量. 匹配分数矩阵$S\in\mathbb{R}^{|\mathcal{W}|\times|\mathcal{T}|}$表示所有执行者-任务对的匹配程度.

动作输出层: 对于每个执行者w_i, 动作空间包括选择任务$t_j(j\in\{0,1,\ldots,|\mathcal{T}|-1\})$或保持空闲($a_i=|\mathcal{T}|$). 动作概率分布通过 Softmax 函数计算:

$$\pi_\theta(a_i=j|s_t)=\frac{\exp(s_{ij})\cdot M_{ij}}{\sum_{k=0}^{|\mathcal{T}|}\exp(s_{ik})\cdot M_{ik}}. \quad (19)$$

其中$M_{ij}\in\{0,1\}$为动作掩码, 由约束 (C1)、(C2) 决定 ((C3)、(C4) 未纳入掩码, 见约束条件小节); 空闲动作固定$M_{i,|\mathcal{T}|}=1$. 空闲动作的分数$s_{i,|\mathcal{T}|}=\bar{s}_i-m_{idle}$, 其中$\bar{s}_i$为执行者$w_i$匹配分数的均值, m_{idle}为可学习空闲边际参数 (初始值 5.0), 自适应调节选择任务与保持空闲的倾向. 最终的动作概率分布为:

$$\pi_\theta(a_t|s_t)=\prod_{i=1}^{|\mathcal{W}|}\pi_\theta(a_i|s_t). \quad (20)$$

式中乘积分解表示网络先为各执行者生成候选动作$u_t\sim\pi_\theta(\cdot|s_t)$, 避免显式枚举联合动作空间; 随后经冲突消解算子$\Psi(\cdot)$得到实际动作$\hat{a}_t$(见第 4.4 节).

3.2.2 Critic 网络 (价值网络)

Critic 网络$Q_\phi(s_t,a_t)$负责评估状态-动作对的价值. 网络结构包括:

状态编码器: 将状态s_t编码为状态特征:

$$h_s=\mathrm{ReLU}(W_7\cdot\mathrm{Flatten}([T_t;W_t])+b_7). \quad (21)$$

动作编码器: 将动作a_t编码为动作特征:

$$h_a=\mathrm{ReLU}(W_8\cdot\mathrm{OneHot}(a_t)+b_8). \quad (22)$$

价值评估层: 融合状态和动作特征, 输出价值估计:

$$Q_\phi(s_t,a_t)=W_9\cdot\mathrm{ReLU}(W_{10}\cdot[h_s;h_a]+b_{10})+b_9. \quad (23)$$

3.3 训练策略

3.3.1 损失函数

Critic 网络的损失函数采用 TD 误差的均方误差:

$$L_Q=\mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(Q_\phi(s,a)-y)^2]. \quad (24)$$

其中目标值y定义为:

$$y=r+\gamma\mathbb{E}_{a'\sim\pi_{\theta'}(\cdot|s')}[Q_{\phi'}(s',a')]=r+\gamma\sum_{a'}\pi_{\theta'}(a'|s')Q_{\phi'}(s',a'). \quad (25)$$

其中$\pi_{\theta'}(a'|s')$表示目标策略网络在状态s'下选择动作a'的概率. 由于联合动作空间随执行者数指数增长, 精确求和不可行, 本文采用K次蒙特卡洛采样近似策略期望:

$$y=r+\gamma(1-d)\cdot\frac{1}{K}\sum_{k=1}^{K}Q_{\phi'}(s',\tilde{a}'^{(k)}). \quad (26)$$

式中$\tilde{a}'^{(k)}\sim\pi_{\theta'}(\cdot|s')$, 取$K=4$(计算效率与稳定性的工程权衡, 近似误差见局限性).

Actor 网络通过可微的软动作代理目标更新. 环境执行离散候选动作u_t, Actor 训练时以动作概率矩阵的展平作为可微代理输入 Critic, 避免对离散采样求梯度:

$$L_{\mathrm{actor}}=-\mathbb{E}_{s\sim\mathcal{D}}[Q_\phi(s,\mathrm{flatten}(\mathbf{M}(s)\odot\pi_\theta(\cdot|s)))]. \quad (27)$$

式中, $\mathbf{M}(s)$为动作掩码, $\odot$为逐元素乘积, $\mathrm{flatten}(\cdot)$将$|\mathcal{W}|\times(|\mathcal{T}|+1)$的联合动作概率矩阵展平为向量后输入 Critic, 保留执行者-任务分配信息. 该软动作代理属工程近似 (局限性见结论部分).

3.3.2 优先经验回放

采用优先经验回放 (Prioritized Experience Replay) 提高学习效率[18]. 定义经验(s,a,r,s')的优先级为:

$$p_i=|\delta_i|+\epsilon. \quad (28)$$

其中$\delta_i=Q_\phi(s,a)-y$为 TD 误差, ϵ为小常数避免优先级为 0.

$$P(i)=\frac{p_i^\alpha}{\sum_j p_j^\alpha}. \quad (29)$$

其中$\alpha\in[0,1]$控制优先级的强度.

$$w_i=(\frac{1}{N}\cdot\frac{1}{P(i)})^\beta. \quad (30)$$

其中$\beta\in[0,1]$控制重要性采样的强度, N为经验回放缓冲区大小.

$$L_Q=\mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[w_i\cdot(Q_\phi(s,a)-y)^2]. \quad (31)$$

3.3.3 软更新机制

采用软更新 (Soft Update) 机制更新目标网络, 保证训练稳定性:

$$\theta'\leftarrow\tau\theta+(1-\tau)\theta', \quad (32)$$

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi'. \tag{33}$$

其中$\tau \ll 1$为软更新系数 (通常取 0.005).

3.4 任务冲突处理机制

针对多智能体分布式决策中的任务竞争冲突, 本文设计基于先到先得 (FCFS) 原则的冲突消解机制: 当多个执行者选择同一任务时, 按预计到达时间$t^i_{arrival}$(由当前位置$p_i(t)$到任务l_j的欧氏距离计算) 优先分配给最早到达者, 其余执行者重规划或保持待机. 该机制减少资源碰撞与死锁, 并通过隐式冲突惩罚引导分散化空间分布, 提升协作效率与任务覆盖率.

3.5 动态维度适配机制

为应对任务与执行者数量动态波动导致的状态-动作空间维度变化, DRL-CTA 提出基于动态缓存的维度适配机制: 维护可复用的神经网络模块缓存池, 实体规模变更时优先复用匹配当前维度的编码器实例, 无匹配则按需构建并纳入缓存. 该"按需构建、优先复用"策略降低动态变化带来的计算开销, 有助于跨规模平滑迁移与训练稳定.

3.6 理论分析

3.6.1 收敛性分析

DRL-CTA 的收敛性可基于随机逼近理论分析. 假设状态空间$\mathcal{S}$与动作空间$\mathcal{A}$有限、奖励函数有界, 学习率序列满足 Robbins-Monro 条件 ($\sum \alpha_t = \infty$, $\sum \alpha_t^2 < \infty$)、探索策略保证状态-动作对访问遍历性, 则 Critic 通过最小化 TD 误差逼近真实价值函数$V^\pi(s)$, Actor 沿策略梯度方向更新参数θ, 为收敛到局部较优策略提供理论条件.

3.6.2 复杂度分析

算法单步推理的时间复杂度为$O(|\mathcal{T}|d_h + |\mathcal{W}|d_h + H|\mathcal{W}||\mathcal{T}|d_k)$($d_h$为隐藏层维度, H为注意力头数, 二者通常为常数), 与问题规模呈多项式关系; 空间复杂度主要由网络参数$O(d_h^2)$与经验回放缓冲区$O(N(d_s + d_a))$构成. 相较传统组合优化算法的指数级复杂度, DRL-CTA 具备处理大规模问题的计算可行性.

4 实验结果与分析

4.1 实验设置

4.1.1 数据集

为评估 DRL-CTA 性能, 本文构建基于 Python 的空地协同移动群智感知仿真环境, 区域为1000×1000的二维连续空间; 执行者初始水平位置服从均匀分布. 静态实验按任务规模同步调整执行者配置 (50/100 任务对应 2 无人机+3 地面用户, 200 任务对应 5+5, 300 任务对应 5+10), 动态实验采用 10 无人机+20 地面用户. 无人机速度10 ∼ 20单位/时间步、初始电量0.6 ∼ 1.0; 地面用户速度2 ∼ 8单位/时间步、无能耗限制.

任务奖励在$[50, 200]$内随机采样, 可靠性需求$[0.7, 0.95]$, 传感器需求随机 1 至 2 种; 发布时间在$[0, 20]$时间步内, 截止时间按简单/中等/困难/过载难度动态设定.

4.1.2 对比算法

本文选取五种代表性基准算法进行对比: Random 作为随机下界; Greedy 按任务奖励优先进行贪心指派; PSAllocator [20] 综合感知能力、可靠性和剩余时间构造优先级; HyTasker [22] 结合机会主义与参与式模式进行两阶段优化; GA 通过选择、交叉与变异搜索分配方案.

4.1.3 评估指标

采用以下四项指标评估算法性能. 任务完成率 (CR) 衡量成功分配并执行的任务比例:

$$\mathrm{CR} = \frac{|\mathcal{T}_{completed}|}{|\mathcal{T}|} \times 100\%. \tag{34}$$

平均响应时间 (ART) 衡量从任务可被执行起到实际完成的端到端延迟, 反映系统对任务需求的响应速度:

$$\mathrm{ART} = \frac{1}{|\mathcal{T}_{completed}|} \sum_{j \in \mathcal{T}_{completed}} (t^j_{complete} - t^j_{arrive}). \tag{35}$$

其中$t^j_{complete}$为任务j完成时刻 (执行者到达任务位置的时刻), $t^j_{arrive} = e_j$为最早可执行时间.

执行者利用率 (WU) 衡量执行者资源的有效利用程度:

$$\mathrm{WU} = \frac{\sum_{w \in \mathcal{W}} t^{busy}_w}{\sum_{w \in \mathcal{W}} t^{total}_w} \times 100\%. \tag{36}$$

计算时间衡量算法运行的时间开销, 用于评估实时性.

4.1.4 参数设置

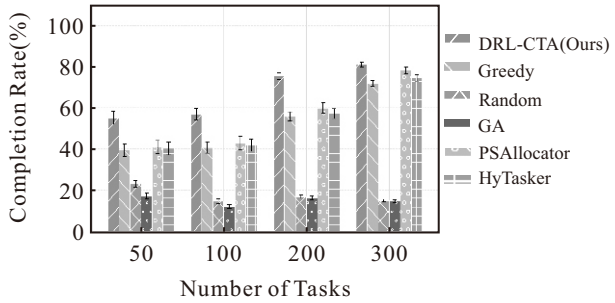
DRL-CTA 关键超参数配置如下: Actor 与 Critic 均采用 Adam 优化器, 学习率3×10^{-4}; 折扣因子$\gamma = 0.99$; 软更新系数$\tau = 0.005$; 经验回放缓冲区容量 10000, 批量大小 32; 训练共 1000 个 episode, 探索率ϵ指数衰减 ($\epsilon_t = \max(0.01, 1.0 \cdot 0.995^t)$).

4.2 性能评估

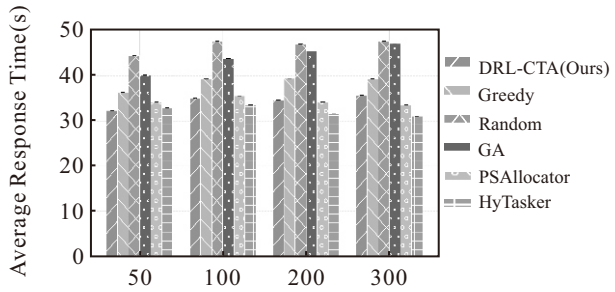
4.2.1 任务完成率对比

图 2 展示了不同算法在不同任务规模下的性能

对比.



(a) Task Completion Rate



(b) Average Response Time

图 2 不同任务规模下的算法完成率 (a) 与响应时间 (b) 对比

表 1 展示了不同算法在不同任务规模下的任务完成率 (平均值±标准差).

表 1 不同算法的任务完成率对比

任务数量	DRL-CTA	Greedy	PSAllocator	HyTasker
50	55.1%±15.9%	39.4%±14.9%	40.9%±16.1%	40.3%±16.1%
100	57.0%±14.1%	40.7%±14.1%	42.9%±15.9%	41.8%±14.8%
200	75.4%±8.2%	55.8%±11.2%	59.9%±12.7%	57.1%±12.0%
300	80.9%±5.5%	71.8%±7.0%	78.2%±8.1%	74.5%±7.6%

DRL-CTA 在各任务规模下任务完成率均优于全部五种基线方法.

实验结果表明, DRL-CTA 在不同任务规模下均取得了更高的任务完成率. 在 300 任务的大规模场景下完成率达到 80.9%, 相比 Greedy 算法的 71.8% 提升约 9.1 个百分点. 结合图 2(a) 的趋势曲线可以观察到, 随着任务规模从 50 增加到 200, DRL-CTA 的完成率从 55.1% 提升到 75.4%, 且在 200 任务节点, 相比 Greedy 算法的 55.8%, DRL-CTA 的优势扩大至 19.6 个百分点. 这表明在资源受限和高负载的复杂场景下, DRL-CTA 通过全局价值评估能够减少局部最优带来的性能损失.

值得注意的是, PSAllocator 在 300 任务场景下完成率达 78.2%, 与 DRL-CTA 差距为 2.7 个百分点, 小于 200 任务场景. 300 任务场景包含 15 个执行者, 任务/执行者比与 200 任务场景相同, 均为 20. PSAllocator 与 DRL-CTA 的差距缩小可能与任务实例及算法适配性有关, 具体原因仍需进一步验证.

4.2.2 平均响应时间对比

响应时间对比结果已合并至图 2(b) 展示. 表 2 展示了不同算法在不同执行者配置下的平均响应时间 (单位: 秒, 平均值±标准差).

表 2 不同算法的平均响应时间对比

执行者配置	DRL-CTA	Greedy	PSAllocator	HyTasker
2UAV+3G	35.11±3.59	39.31±3.91	35.62±3.15	33.60±3.29
5UAV+10G	35.77±1.51	39.37±1.47	33.61±1.58	31.10±1.58
5UAV+5G	34.68±1.72	39.48±2.27	34.22±2.02	31.58±1.91

DRL-CTA 平均响应时间低于 Greedy: 200 任务 (5UAV+5G) 下 34.68s vs 39.48s, 300 任务 (5UAV+10G) 下 35.77s vs 39.37s. HyTasker 响应时间略低, 因其采用激进就近分配、以较低完成率换更快响应; DRL-CTA 则统一优化响应时间与完成率, 300 任务下完成率较 HyTasker 高 6.4 个百分点、200 任务下高 18.3 个百分点, 体现综合优势.

4.2.3 执行者利用率对比

DRL-CTA 在高负载场景 (200/300 任务) 下能够更充分地调动执行者资源, 并在静态与动态任务场景中均保持较高的任务完成率与较低的响应时间, 体现其全局优化决策能力.

表 3 展示了执行者利用率对比结果 (平均值±标准差).

表 3 不同算法的执行者利用率对比

场景	DRL-CTA	Greedy	PSAllocator	HyTasker
50Tasks, 2UAV+3G	6.1%±2.4%	4.2%±1.8%	4.4%±2.4%	4.2%±1.8%
100Tasks, 2UAV+3G	11.9%±3.6%	8.3%±3.0%	8.7%±3.3%	8.5%±3.1%
200Tasks, 5UAV+5G	16.1%±2.9%	11.4%±2.5%	12.2%±2.9%	11.6%±2.5%
300Tasks, 5UAV+10G	17.3%±2.2%	15.0%±2.2%	16.0%±1.9%	15.3%±1.9%

注: 场景配置分别为: 50/100任务(2UAV+3G), 200任务(5UAV+5G), 300任务(5UAV+10G). 所有场景均为Overloaded难度.

4.2.4 可扩展性分析

为验证 DRL-CTA 在规模变化场景下的适应能力, 实验对比了不同任务规模 (50 至 300 个任务) 及相应执行者配置下各算法的性能演变趋势. 在该设置下, DRL-CTA 在 300 个任务场景中仍能保持较高任务完成率与较低响应时间, 说明注意力机制有助于处理变长任务-执行者匹配关系.

为进一步区分资源配置效应与算法本身的可扩展性, 本文补充了两组受控对照实验, 复用 5 个独立训练种子的 full 模型, 通过动态维度适配机制迁移评估不同规模场景, 结果如表 4 所示.

表 4　DRL-CTA 可扩展性实验

配置	DRL-CTA	Greedy	PSAllocator
固定执行者数量(*4 UAV + 6 GU*), 任务数递增			
50 tasks	0.739	0.577	0.598
100 tasks	0.757	0.606	0.641
150 tasks	0.758	0.606	0.652
200 tasks	0.748	0.600	0.655
固定任务/执行者比例(=*10*), 系统规模递增			
50T/2U3G	0.562	0.399	0.417
100T/4U6G	0.757	0.606	0.641
200T/8U12G	0.874	0.770	0.813
300T/12U18G	0.913	0.849	0.888

在固定执行者数量 (4 架无人机、6 名地面用户)、任务数从 50 递增至 200 的设置下, DRL-CTA 完成率稳定在 0.739-0.758, 相比 Greedy 和 PSAllocator 保持约 9-16 个百分点的领先, 且未随任务数增加而明显下降, 表明该架构在固定资源条件下具备较好的抗负载能力. 在固定任务/执行者比例 (=10)、系统规模从 50 任务 (2 无人机+3 地面用户) 扩展至 300 任务 (12 无人机+18 地面用户) 的设置下, DRL-CTA 完成率从 0.562 提升至 0.913, 全规模范围内均领先两个基线算法, 验证了 DRL-CTA 在资源比例不变时具备良好的规模扩展性.

4.3　消融实验

为验证 DRL-CTA 算法中各关键组件的有效性, 设计了消融实验对比以下变体: (1) 无注意力机制 (No-Attn): 移除多头注意力模块, 仅使用全连接层处理状态特征; (2) 无优先经验回放 (No-PER): 采用均匀随机采样替代优先经验回放; (3) 无软更新 (No-Soft): 直接复制参数更新目标网络; (4) 完整算法 (Full). 消融实验采用高负载配置 (100 个任务、4 架无人机和 6 名地面用户, 任务/执行者比为 10.0), 并采用 5 个随机种子 (0、42、123、456、789) 各 100 个评估 episode 的结果进行统计分析.

表 5 展示了消融实验结果 (平均值±标准差).

结果表明, 各消融变体的任务完成率均低于完整模型 (74.6%), 其中移除软更新机制 (No-Soft) 对完成率的降幅最大 (完成率下降至 64.4%, 降幅 13.8%, Cohen's $d_z = 0.99$, 大效应量, $p = 0.092$), 提示软更新在稳定目标网络估计、抑制训练震荡方面可能存在重要作用; 移除注意力机制 (No-Attn) 次之 (完成率下降至 67.8%, 降幅 9.1%, $d_z = 0.73$, 中等效应量, $p = 0.179$), 提示多头注意力在捕获任务-执行者时空关联特征方面可能存在作用; 移除优先经验回放 (No-PER) 的影响最小 (完成率下降至 72.6%, 降幅 2.8%, $d_z = 0.26$, $p = 0.586$). 软更新与注意力机制的效应量达中等以上, 但因样本量$n = 5$有限, 均未达到$p < 0.05$显著水平. 整体上, 完整模型在完成率与利用率上最优, 响应时间与 No-PER 基本相当; 软更新与注意力机制显示较大效应量, PER 的边际贡献有限.

表 5　消融实验结果

配置	任务完成率	平均响应时间	执行者利用率
无注意力机制	67.8%±9.2%	32.62±2.16s	7.3%±1.3%
无优先经验回放	72.6%±7.6%	31.78±0.74s	8.0%±1.0%
无软更新	64.4%±10.8%	32.31±1.05s	6.9%±1.5%
完整DRL-CTA	74.6%±0.7%	31.97±0.22s	8.3%±0.1%

为进一步考察 DRL-CTA 与最强基线 PSAllocator 在不同任务难度下的表现差异, 表 6 给出了两者在 easy/medium/hard/overloaded 四种难度下的分层完成率与种子级配对统计结果.

表 6　不同任务难度下的分层完成率与配对统计

难度	DRL-CTA	PSAllocator	paired-t p	Cohen's d_z
easy	0.946	0.931	0.058	1.18
medium	0.936	0.919	0.051	1.23
hard	0.905	0.877	0.009	2.15
overloaded	0.757	0.641	< 0.001	9.59

结果表明, 随着难度提升, DRL-CTA 相对 PSAllocator 的完成率优势逐步扩大. 配对t检验显示, easy 和 medium 难度下的差异均未达到$p < 0.05$显著水平 (p分别为 0.058 和 0.051, d_z分别为 1.18 和 1.23); hard 和 overloaded 难度下的差异达到显著水平 ($p = 0.009$, $d_z = 2.15$; $p < 0.001$, $d_z = 9.59$). Wilcoxon 符号秩检验在四种难度下均未达到$p < 0.05$显著水平 (p分别为 0.125、0.125、0.0625 和 0.0625). 鉴于种子数$n = 5$有限, 上述统计结果应视为探索性证据.

4.4　动态环境实验

为验证 DRL-CTA 在动态任务到达场景下的适应性, 设计了动态环境实验. 与静态环境不同, 动态环境中任务按一定概率持续到达, 模拟真实应用场景中任务需求的动态变化.

4.4.1　实验设置

为模拟真实城市感知中任务动态到达的特性, 构建了基于伯努利到达过程的动态仿真环境. 实验设定初始时刻存在 50 个基础任务, 随后新任务按伯努利过程持续到达: 每个时间步以概率λ生成一个新任务, 到达率λ分别设置为低 (0.1)、中 (0.2) 和高 (0.4) 三个等级, 对应不同的任务负载压力. 该离散时间到达模型是连续时间泊松过程的标准近似, 在时

间步长足够小时两者趋于一致[32]. 最大任务积压阈值根据到达率动态调整, 以模拟系统容量限制. 执行者配置保持为 10 架无人机与 20 名地面用户.

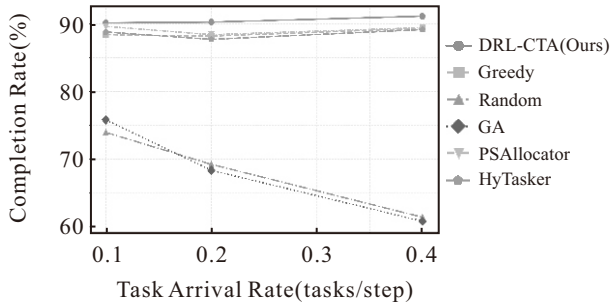
适应性得分定义为响应时间稳定性的度量:

$$\text{Adaptation Score} = \frac{1}{1+\sigma_{response}}. \tag{37}$$

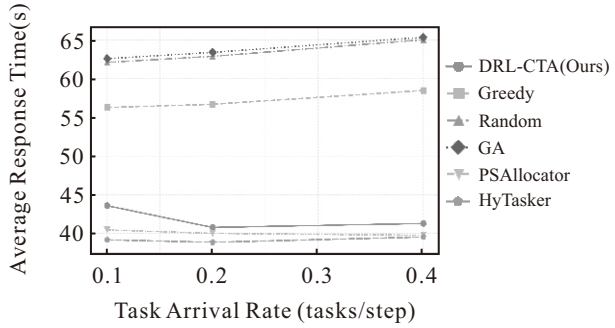
其中$\sigma_{response}$为响应时间的标准差. 该指标反映算法在动态环境中保持稳定性能的能力.

4.4.2 实验结果

图 3 展示了不同任务到达率下各算法的任务完成率与平均响应时间对比.



(a) Completion Rate vs Task Arrival Rate



(b) Response Time vs Task Arrival Rate

图 3 动态环境下不同任务到达率的算法完成率与响应时间对比

表 7 展示了不同任务到达率下各算法的性能对比.

由表 7 和图 3 可知, DRL-CTA 在动态环境中保持较高的鲁棒性. 随着任务到达率从低 (0.1) 增加到高 (0.4), Random 和 GA 算法的完成率明显下降: Random 从 73.9% 降至 61.4%, GA 从 75.8% 降至 60.8%, 表明基线算法难以应对高并发负载. 相比之下, DRL-CTA 在三种到达率下的完成率均稳定在 90% 以上, 在本次三种到达率设定下均高于 PSAllocator、HyTasker 和 Greedy. 在平均响应时间方面, DRL-CTA 同样优于传统启发式基线: 其响应时间相比 Greedy 降低约 23%-29%, 相比 Random 和 GA 降低约 30% 以上; PSAllocator 与 HyTasker 的响应时间虽略低于 DRL-CTA, 但二者完成率均低于 DRL-CTA, 其响应时间优势并未转化为完成率优势, 体现了 DRL-CTA 在完成率与响应时间两项指标上的综合优势. 此外, DRL-CTA 的吞吐量随负载提升, 适应性得分与最强基线 PSAllocator、HyTasker 保持相当, 说明其在负载波动下能够维持稳定的服务体验.

表 7 动态环境实验结果

到达率配置	算法	完成率	响应时间(s)	吞吐量	适应性得分
低到达率	DRL-CTA	90.1%	43.6	0.178	0.034
低到达率	Greedy	88.4%	56.3	0.174	0.030
低到达率	Random	73.9%	62.1	0.148	0.032
低到达率	GA	75.8%	62.6	0.150	0.032
低到达率	PSAllocator	89.6%	40.4	0.177	0.035
低到达率	HyTasker	88.8%	39.1	0.175	0.034
中到达率	DRL-CTA	90.3%	40.7	0.269	0.035
中到达率	Greedy	88.2%	56.7	0.262	0.030
中到达率	Random	69.2%	62.9	0.205	0.032
中到达率	GA	68.3%	63.4	0.205	0.032
中到达率	PSAllocator	88.4%	40.0	0.263	0.036
中到达率	HyTasker	87.7%	38.8	0.261	0.036
高到达率	DRL-CTA	91.1%	41.3	0.454	0.036
高到达率	Greedy	89.3%	58.5	0.445	0.030
高到达率	Random	61.4%	65.1	0.308	0.032
高到达率	GA	60.8%	65.4	0.304	0.031
高到达率	PSAllocator	89.4%	39.7	0.446	0.037
高到达率	HyTasker	89.1%	39.5	0.444	0.037

4.5 统计显著性分析

为分析实验结果, 本文采用配对t检验 (正态假设) 与 Wilcoxon 符号秩检验 (非参数) 评估各算法性能差异, 并以 Cohen's d效应量 ($|d|>0.8$为大效应、$0.5<|d|\leqslant 0.8$为中效应、$|d|\leqslant 0.5$为小效应) 量化差异的实际意义.

对比实验: 主对比实验采用代表性训练种子 (seed_0) 的 100 个评估 episode, DRL-CTA 在各任务规模下任务完成率均优于全部五种基线方法, 具体数值见表 1; 种子级配对统计检验见消融实验 (表 5) 与难度分层分析 (表 6).

消融实验: 以 5 个独立训练种子的种子级配对检验 ($n=5$) 评估各组件贡献. 软更新机制的效应量最大 (完成率下降 13.8%, Cohen's $d_z=0.99$, 大效应, $p=0.092$); 多头注意力机制次之 (下降 9.1%, $d_z=0.73$, 中等效应, $p=0.179$); 优先经验回放的贡献最小 (下降 2.8%, $d_z=0.26$, $p=0.586$). 软更新与注意力机制的效应量达中等以上, 但因样本量$n=5$有限, 均未达到$p<0.05$显著水平.

5 结 论

本文针对低空经济背景下的群智感知空地协同任务分配问题, 提出了一种基于深度强化学习的空

地协同任务分配方法 DRL-CTA. 该方法将异构执行者 (无人机与地面用户) 的协同分配问题建模为带有多重时空约束的马尔可夫决策过程, 并设计了融合多头注意力机制与优先经验回放的 Actor-Critic 求解框架. 通过端到端学习, DRL-CTA 能够从高维状态空间中提取任务与执行者之间的关联特征, 缓解传统启发式算法在动态环境和大规模问题下的局限性.

实验结果表明在 300 任务场景下, DRL-CTA 完成率达 80.9%、利用率 17.3%; 动态环境下三种到达率的完成率均保持最高且稳定; 消融实验显示软更新效应量最大、注意力次之、优先经验回放贡献最小.

本研究的理论贡献体现在三个方面: (1) 构建了空地协同移动群智感知任务分配数学模型, 统一描述异构执行者的能力差异与时空耦合约束, 为相关研究提供理论框架; (2) 证明了该问题的 NP-hard 性质, 并基于随机逼近理论讨论了 DRL-CTA 算法满足收敛所需的条件, 补充了深度强化学习在异构协同分配领域的理论分析; (3) 提出了一种基于注意力机制的可变维度状态表征方法, 用于处理传统强化学习难以适应的动态变化动作空间, 为大规模多智能体系统的协同控制提供参考.

5.1 局限性与未来工作

尽管 DRL-CTA 取得较好性能, 但仍存在局限. 首先, 实验基于仿真环境并假设状态可完整观测, 而真实低空场景中通信中断或定位误差会造成状态缺失, 后续可采用 POMDP 刻画观测缺失并评估历史观测的作用. 此外, 电池容量约束 (C3) 与空间可达性约束 (C4) 未纳入动作掩码硬性判定, 属于工程近似, 后续可显式加入电量安全阈值与地面用户任务高度限制. 其次, 尚未处理执行者失效后的任务重分配, 后续可加入局部重规划机制, 在检测到节点失效后将受影响任务快速转移.

参考文献 (References)

[1] Zhang W, Li L Z, Zhang N, et al. Air-ground integrated mobile edge networks: A survey[J]. IEEE Access, 2020, 8: 125998-126018.

[2] 郭戈, 刘上. 空地跨域无人系统低空协同立体监测与控制[J]. 控制与决策, 2026, 41(6): 1509-1517.
(Guo G, Liu S. Cooperative control of air-ground cross-domain unmanned systems for low-altitude three-dimensional monitoring tasks[J]. Control and Decision, 2026, 41(6): 1509-1517.)

[3] Li X, Tan J W, Liu A F, et al. A novel UAV-enabled data collection scheme for intelligent transportation system through UAV speed control[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(4): 2100-2110.

[4] Guo B, Wang Z, Yu Z W, et al. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm[J]. ACM Computing Surveys, 2015, 48(1): 1-31.

[5] Menouar H, Guvenc I, Akkaya K, et al. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges[J]. IEEE Communications Magazine, 2017, 55(3): 22-28.

[6] Fu L W, Zhao Z W, Min G Y, et al. Energy-efficient 3-D data collection for Multi-UAV assisted mobile crowdsensing[J]. IEEE Transactions on Computers, 2023, 72(7): 2025-2038.

[7] Wang Y T, Damani M, Wang P, et al. Distributed reinforcement learning for robot teams: A review[J]. Current Robotics Reports, 2022, 3(4): 239-257.

[8] Hao H, Xu C Q, Zhang W, et al. Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority[J]. IEEE Transactions on Mobile Computing, 2024, 23(9): 8649-8663.

[9] Khamis A, Hussein A, Elmogy A. Multi-robot task allocation: A review of the state-of-the-art[M]. Cooperative Robots and Sensor Networks. Cham: Springer International Publishing, 2015: 31-51.

[10] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[11] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[12] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[PP/OL]. 2017, arXiv: 1706.02275.

[13] Solomon M M. Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations Research, 1987, 35(2): 254-265.

[14] Deb K. Multi-objective optimization using evolutionary algorithms[M]. Chichester: Wiley, 2001.

[15] Ausiello G, Crescenzi P, Gambosi G, et al. Complexity and approximation: Combinatorial optimization problems and their approximability properties[M]. Berlin: Springer, 1999.

[16] Baxter L A, Puterman M L. Markov decision processes: Discrete stochastic dynamic programming[J]. Technometrics, 1995, 37(3): 353.

[17] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems. Long Beach: Curran Associates, 2017: 5998-6008.

[18] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[C]. Proceedings of the 4th International Conference on Learning Representations. San Juan, 2016: 256-265.

[19] Liu B, Wang S L, Li Q H, et al. Task assignment of

UAV swarms based on deep reinforcement learning[J]. Drones, 2023, 7(5): 297.

[20] Wang J T, Wang Y S, Zhang D Q, et al. PSAllocator: Multi-task allocation for participatory sensing with sensing capability constraints[C]. Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing. New York: ACM, 2017: 1139-1151.

[21] Xiong H Y, Zhang D Q, Chen G L, et al. iCrowd: Near-optimal task allocation for piggyback crowdsensing[J]. IEEE Transactions on Mobile Computing, 2016, 15(8): 2010-2022.

[22] Wang J T, Wang F, Wang Y S, et al. HyTasker: Hybrid task allocation in mobile crowd sensing[J]. IEEE Transactions on Mobile Computing, 2020, 19(3): 598-611.

[23] Wei X H, Wang Y F, Tan J, et al. Data quality aware task allocation with budget constraint in mobile crowdsensing[J]. IEEE Access, 2018, 6: 48010-48020.

[24] 蒋伟进, 陈君鹏, 张婉清, 等. 移动群智感知中基于任务质量的多任务分发参与者选择[J]. 控制与决策, 2022, 37(10): 2667-2676.
(Jiang W J, Chen J P, Zhang W Q, et al. Multitask-oriented participant selection based on task quality in mobile crowd sensing[J]. Control and Decision, 2022, 37(10): 2667-2676.)

[25] 杨潇, 郭一楠, 吉建娇, 等. 异构群智感知 PPO 多目标任务指派方法[J]. 控制理论与应用, 2024, 41(6): 1056-1066.
(Yang X, Guo Y N, Ji J J, et al. PPO multi-objective task allocation method for heterogeneous crowd sensing[J]. Control Theory & Applications, 2024, 41(6): 1056-1066.)

[26] Wei W T, Chen H L, Liu X, et al. Towards time-constrained task allocation in semi-opportunistic mobile crowdsensing[J]. Ad Hoc Networks, 2023, 150: 103282.

[27] 刘娅汐, 李旭龙, 霍佳皓, 等. 面向工业场景的无人机时空众包资源分配[J]. 工程科学学报, 2025, 47(1): 91-100.
(Liu Y X, Li X L, Huo J H, et al. UAV spatiotemporal crowdsourcing resource allocation based on deep reinforcement learning[J]. Chinese Journal of Engineering, 2025, 47(1): 91-100.)

[28] Elfakharany A, Ismail Z H. End-to-end deep reinforcement learning for decentralized task allocation and navigation for a multi-robot system[J]. Applied Sciences, 2021, 11(7): 2895.

[29] 杨明川, 朱敬华, 李元婧, 等. 一种考虑隐私保护的深度强化学习任务分配模型[J]. 计算机研究与发展, 2023, 60(11): 2650-2659.
(Yang M C, Zhu J H, Li Y J, et al. Task allocation model based on deep reinforcement learning considering privacy protection[J]. Journal of Computer Research and Development, 2023, 60(11): 2650-2659.)

[30] 张华卿, 郝明瑞, 姜吉祥, 等. 面向紧急任务的多智能体批次任务分配[J]. 控制理论与应用, 2025, 42(11): 2242-2251.
(Zhang H Q, Hao M R, Jiang J X, et al. Multi-agent batch task allocation for urgent tasks[J]. Control Theory & Applications, 2025, 42(11): 2242-2251.)

[31] 姜斌, 马亚杰, 薛舒心. 无人飞行器集群自主控制: 基于联盟形成博弈的任务分配[J]. 自动化学报, 2025, 51(5): 942-959.
(Jiang B, Ma Y J, Xue S X. Autonomous control of unmanned aerial vehicle swarms: Task allocation based on coalition formation game[J]. Acta Automatica Sinica, 2025, 51(5): 942-959.)

[32] Sutton R S, Barto A G. Reinforcement Learning[M]. The 2nd edition. Cambridge: MIT Press, 2018.

作者简介

刘晓亮, 主要研究方向为移动群智感知, E-mail: v1zone@163.com;

蒋伟进, 主要研究方向为分布式人工智能、复杂系统建模, E-mail: jwjnudt@163.com;

王刚, 通信作者, 主要研究方向为无人机路径规划, E-mail: 3045@hutb.edu.cn.