XXXX

基于分层博弈与混合参数化控制的云资源调度研究

摘　要：面对云计算环境的大规模并发性、高度动态性以及多目标冲突（如完工时间、能耗与资源利用率之间的权衡）的严峻挑战，传统的启发式算法难以适应负载潮汐，而单智能体深度强化学习（DRL）在处理大规模状态空间时往往陷入“维度灾难”。本文提出了一种创新HGT-MARL-CS-PSO框架，旨在通过分层博弈机制解决云资源调度的复杂性。首先，本文将调度问题形式化为一个两级斯塔克伯格博弈模型：高层管理智能体作为领导者，根据全局负载特征制定宏观战略目标；低层执行智能体作为跟随者，在战略约束下进行非合作纳什博弈，以平衡局部资源竞争与全局协作。其次，为了克服离散动作空间爆炸的问题，本文设计了一种混合参数化控制机制，智能体并不直接输出任务映射，而是输出连续的权重参数向量，动态驱动底层的布谷鸟搜索-粒子群调度器进行微观寻优。理论分析证明了该框架在斯塔克伯格均衡点的收敛性。Google Borg Trace真实数据集的大规模实验表明，HGT-MARL-CS-PSO在完工时间、能耗效率及资源利用率等关键指标上显著优于现有的DRL基线及主流元启发式算法，展现了卓越的泛化能力与鲁棒性。

关键词：云资源调度；多智能体强化学习；斯塔克伯格博弈；参数化控制

中图分类号：TP393

文献标志码：A

doi: 10.11959/j.issn.1000−0801.

Research on Cloud Resource Scheduling Based on Hierarchical Game Theory and Hybrid Parameterized Control

Abstract: Cloud resource scheduling in large-scale data centers faces severe challenges characterized by high concurrency, dynamic workload volatility, and conflicting optimization objectives (e.g., tradeoffs between makespan, energy consumption, and resource utilization). Traditional heuristics often lack adaptability, while single-agent Deep Reinforcement Learning (DRL) approaches suffer from the "curse of dimensionality" when scaling to large state spaces. To address these limitations, this paper proposes HGT-MARL-CS-PSO, a novel framework that orchestrates scheduling through a hierarchical game-theoretic approach. First, we formulate the scheduling problem as a Two-Level Stackelberg Game: a high-level Manager Agent (M-Agent) acts as the Leader, defining global strategic goals based on macro-workload patterns, while low-level Executor Agents (E-Agents) act as Followers, engaging in a non-cooperative Nash game to balance local resource competition with global cooperation. Second, to mitigate the explosion of discrete action spaces, we introduce a Hybrid Parameterized Control Mechanism. Instead of generating task mappings directly, the RL agents output continuous weight parameter vectors to dynamically configure a subordinate Cuckoo Search-Particle Swarm Optimization (CS-PSO) scheduler for precise micro-execution. Theoretical analysis validates the convergence of the proposed framework towards a Stackelberg Equilibrium. Extensive experiments using real-world Google Borg Traces demonstrate that HGT-MARL-CS-PSO significantly outperforms state-of-the-art DRL baselines and meta-heuristics in terms of makespan reduction, energy efficiency, and resource utilization, prov-

收稿日期：2026−04−07；修回日期：XXXX−XX−XX

ing its superior generalization and robustness in complex environments.

Key words: Cloud Resource Scheduling, Multi-Agent Reinforcement Learning (MARL), Stackelberg Game, Parameterized Control

1 引言

随着云计算范式的不断演进以及大模型训练、边缘计算等新兴计算密集型业务的爆发，云数据中心的规模正在经历前所未有的急剧扩张。在这一背景下，云计算资源配置已被工业界与学术界广泛视为第五大公用事业基础设施[1]。作为云操作系统的核心组件，云资源调度负责将海量并发的异构任务映射至底层物理节点。然而，由于该过程需要在完工时间（Makespan）、系统能耗、资源利用率等多重相互冲突的目标之间寻求帕累托最优 [2]，云资源调度被严格证明为一类高度复杂的 NP-难（NP-hard）组合优化问题[3]。如何在保障服务质量（QoS）的前提下，实现数据中心级别的能效动态整合，已成为当前亟待解决的关键挑战。

早期针对云资源调度的研究主要聚焦于静态启发式规则，如先来先服务（FCFS）与最小化完成时间（Min-Min）等策略[4]。此类方法依靠贪心机制在极短时间内完成任务分配，计算开销极低。但其“短视”的决策本质使其在面对云环境高度动态的潮汐负载时，无法感知全局资源状态，极易陷入局部次优并导致部分高性能节点过载。为了突破单一目标的局限并求解多目标帕累托前沿，以遗传算法（GA）和粒子群优化（PSO）为代表的元启发式算法被广泛引入[5]。尽管此类算法具备优异的无梯度全局寻优能力，但其需要针对每一批次任务在极高维度的解空间中进行成百上千次的离线种群迭代。这种庞大的计算开销会产生极高的调度延迟，难以满足现代云环境对毫秒级实时决策的严苛要求。

深度强化学习（DRL）通过与环境的持续交互学习最优策略，为实现在线动态调度提供了革命性的新范式[6]。DRL 能够自适应地捕捉负载波动的时空特征，并进行端到端的毫秒级推断。随着数据中心异构节点数量的激增，单智能体 DRL 在输出离散的任务-虚拟机映射矩阵时，会遭遇动作空间呈指数级膨胀的“维度灾难”（Curse of Dimensionality），导致神经网络极难收敛[7]。为了缓解单节点的计算压力，多智能体强化学习（MARL）架构被提出，通过将全局调度任务拆解并进行分布式执行[8]。尽管 MARL 显著降低了状态动作空间的维度，但现有的分布式框架多建立在“完全合作”或“完全独立”的系统假设之上，忽略了各个子调度器在竞争底层有限资源时的利益冲突，极易在高并发场景下引发严重的策略震荡与资源盲目抢占[9]。

2025 年，大语言模型（LLM）等生成式人工智能技术开始被引入云调度领域，以增强调度策略的语义理解与成本感知能力[10]。多智能体 LLM 系统的相关前沿研究进一步揭示了，在层级化资源分配过程中，建立一套严密的数学机制以平衡智能体之间的“协作与竞争”关系至关重要[11]。然而，如何将博弈论的制衡机制与深度强化学习的高效推断能力进行无缝融合，目前仍缺乏系统性的理论框架支撑。

为了从底层机制上化解全局能效协作与局部吞吐量竞争的天然冲突，本文创新性地引入斯塔克伯格（Stackelberg）主从博弈模型[12]，提出一种基于分层博弈与混合参数化控制的云资源调度框架（HGT-MARL-CS-PSO）。本文的主要创新与学术贡献如下：

提出两级斯塔克伯格博弈调度模型，突破传统 MARL 的扁平化拓扑，将全局系统抽象为制

定宏观战略的领导者（Leader），将底层分布式调度单元抽象为进行非合作纳什博弈的跟随者（Followers）。通过严格的数学约束机制平抑多智能体间的盲目资源抢占，实现了全局能效与局部性能的帕累托最优折衷。

设计连续参数化控制降维机制，摒弃 RL 直接输出高维离散映射矩阵的传统方式，智能体转而输出低维、连续的适应度权重向量，动态驱动底层的改进型 CS-PSO 算法进行微观寻优。该机制将离散组合优化问题转化为连续参数寻优问题，从根本上突破了超大规模并发调度的维度瓶颈。

理论验证与真实负载评估，从理论层面证明了本框架在斯塔克伯格均衡点的收敛性，并基于 Google Borg Trace 真实集群数据集进行了大规模实证分析，验证了该模型在极端异构负载下兼具低调度延迟与高鲁棒性。

2　相关工作

云数据中心资源调度的演进路径，深刻反映了计算机科学在应对极大规模组合优化问题时的范式变迁。现有研究脉络主要沿传统与元启发式算法、智能学习与大模型协同驱动，以及混合参数化架构三个核心维度展开。

2.1　传统启发式与元启发式调度

早期的云调度研究主要依赖静态启发式规则，如 FCFS、Min-Min 和 Max-Min[4] 。这些方法侧重于解决任务与资源的初步映射问题，创新点在于通过简单的贪婪策略降低了计算复杂度，但其不足在于缺乏对云环境高度动态性的感知，难以处理负载潮汐带来的资源碎片化。

在此之后，GA、ACO 和 PSO 等元启发式算法被引入以解决多目标优化。这些工作的创新在于利用生物启发式搜索在复杂的解空间中寻找帕累托最优解 。然而，从问题解决的角度看，这类算法本质上是离线式优化，面临收敛速度慢和容易陷入局部最优的瓶颈，无法满足大规模数据中心对实时响应的需求[5]。

2.2　强化学习与大模型协同调度

为打破离线迭代的算力桎梏，基于马尔可夫决策过程（MDP）的序列化动态决策方法成为当前的研究焦点。深度强化学习（DRL）凭借其强大的非线性拟合能力，开启了端到端在线资源管理的先河[6]。但针对包含海量任务与虚拟机的异构映射场景，单智能体方法在评估状态-动作价值时，极易因动作维度的高阶非线性组合而导致神经网络梯度爆炸或陷入停滞[7]。为应对此类“维度灾难”，多智能体强化学习（MARL）架构被引入以解耦决策过程[8]。在最新的研究进展中，自适应协同机制已被视为突破动态资源分配瓶颈的核心方向[13]。特别是进入 2025 年以后，学界开始探索利用大语言模型（LLM）的常识推理能力来引导强化学习过程。前沿研究提出了 LLM 引导的 SARSA 算法，通过知识注入显著提高了探索效率并降低了在线推断的偏差[14]。正如博弈论在云资源分配领域的专项调研所明确指出的[15]，无论是传统的 MARL 还是新兴的大模型多智能体系统，在处理大规模异构环境下的多目标利益冲突时，仍普遍缺乏严密的数学制衡机制。当系统同时面临降低全局能效与最大化单节点吞吐量的矛盾时，松散的协作假设往往导致智能体陷入局部资源的零和博弈，最终使整个集群收敛于低效的次优纳什均衡

2.3　混合架构与参数化控制

为了兼顾强化学习的优势与元启发式算法的能力，将 DRL 与元启发式算法深度集成的混合架构（Hybrid Architecture）在近年来引起了学术界的广泛关注。此类方法的核心思想是利用 RL 动态感知环境状态，并据此在线调整底层启发式算法的超参数[16]。在此基础上，前沿的算法优化研究进一步确立了“参数化控制”（Parameterized Heuristics）框架[17]。该框架通过让强化学

习智能体仅输出连续的优化目标权重（如时间权重、能耗权重），成功将决策空间从呈指数级增长的离散任务分配矩阵，降维至低维且固定的连续参数空间。这种降维机制不仅有效规避了组合动作爆炸的风险，极大加速了模型的训练收敛速度，同时还保留了底层元启发式算法的精细化寻优精度。然而，纵观目前的文献，大多参数化控制研究仅局限于单一的控制流或简单的单层优化场景，鲜有研究探讨如何将此种高效的降维控制机制与分层斯塔克伯格博弈（Stackelberg Game）理论进行深度融合。本文所提出的框架正是立足于这一理论空白，旨在为解决极端异构负载及多重利益冲突下的超大规模云资源映射问题，提供一种兼具理论严谨性与工程可行性的全新解决机制。

3 模型构建

本章首先对云数据中心的底层基础设施、任务负载特征及能耗模型进行数学抽象，随后定义多目标优化的关键指标，最后将该调度问题形式化为一个双层斯塔克伯格博弈过程。

3.1 系统与动态任务模型

本文将云数据中心建模为由 L 台异构物理机（Physical Machines, PMs）组成的集群 $H=\{h_1,h_2,...,h_L\}$。每台物理机 h_l 的计算能力由其指令执行速度 C_l 表征，其承载的虚拟机（VMs）集合记为 $\mathcal{V}_l$，集群内虚拟机总数严格约束为 M。

对于到达调度中心的动态批次任务集 $T=t_1,t_2,\ldots,t_N$，任务 t_i 的多维属性由元组 $\langle A_i,L_i\rangle$ 定义，分别表示到达时间与指令长度。定义二元决策变量 $x_{i,j}\in 0,1$，当 $x_{i,j}=1$ 时表示任务 t_i 被分配至虚拟机 v_j。考虑到任务在并发映射时的排队效应，任务 t_i 的完工时间（Finish Time, FT）必须将其在目标节点上的等待时间纳入统筹。其严格定义如下：

$$FT_i=A_i+\sum_{j=1}^{M}x_{i,j}\cdot(T_{i,j}^{wait}+T_{i,j}^{exec})$$

其中，$T_{i,j}^{exec}=L_i/C_j$ 为任务在虚拟机 v_j 上的预期执行时间，$T_{i,j}^{wait}$ 为受限于该节点当前队列长度所产生的实际排队等待时间。当前批次任务的全局最大完工跨度（Makespan）定义为：$Mspan=\max\limits_{t_i\in T}\{FT_i\}$。

3.2 动态能耗评估模型

数据中心的总能耗最终由物理服务器（Host）产生。假设数据中心包含 L 台物理机，表示为集合 $H=\{h_1,h_2,\ldots,h_L\}$。每台物理机上承载了多个虚拟机（VM），虚拟机 v_j 对物理机 h_l 的功耗贡献取决于其对该物理机资源的占用比例。根据已有研究，虚拟机的功耗与其CPU利用率呈现非线性正相关关系[18]。本文采用广泛使用的线性能耗模型来估算虚拟机 v_j 在 t 时刻的瞬时功率。

$$P_j(t)=P_j^{idle}+\left(P_j^{peak}-P_j^{idle}\right)\cdot U_j(t)$$

其中，P_j^{idle} 和 P_j^{peak} 分别表示虚拟机处于空闲状态和满载状态（100%利用率）时的功率，$U_j(t)\in[0,1]$ 是 t 时刻的CPU利用率。整个云数据中心在调度周期 $[0,T_{total}]$ 内的总能耗 E_{total} 定义为所有虚拟机能耗的积分之和：

$$E_{total}=\sum_{j=1}^{M}\int_0^{T_{total}}P_j(t)dt$$

3.3 多目标优化与斯塔克伯格博弈形式化

云资源调度的核心在于打破完工时间、能耗与资源利用率之间的固有冲突。为此，本文构建如下加权多目标适应度函数。

$$F(X)=w_{time}\cdot\frac{Mspan}{Mspan^*}+w_{energy}\cdot\frac{E_{total}}{E_{total}^*}-w_{balance}\cdot U_{avg}$$

其中，$Mspan^*$ 与 E_{total}^* 为归一化基准常数，U_{avg} 为系统平均资源利用率，权重向量参数满足 $w_{time}+w_{energy}+w_{balance}=1$。

考虑到云集群具有天然的层级管理拓扑，本

文将上述优化问题形式化为元组 $\Gamma=\langle \mathcal{N},\Sigma,U\rangle$ 所定义的两级斯塔克伯格博弈。

领导者（M-Agent，全局管理者）：在宏观层面观测系统整体负载压力与能耗状态，主动切换全局战略 $a_L\in$ {Energy, Perf, Balance}，以最大化全局长期综合效用。

跟随者（E-Agents，局部调度器）：在 M-Agent 的战略约束下，多个 E-Agents 针对局部计算资源展开非合作纳什博弈。其输出不再是高维离散的映射矩阵 X，而是连续的权重向量 $w_{k,t}=[w_{time},w_{energy},w_{balance}]^T$，动态驱动底层启发式算法进行微观寻优。为消除冗余计算，其局部竞争效用函数简化为线性差值机制 $R_{local}(s_k)=U_{k,t}^{cpu}-U_{sys,t}$。

4　HGT-MARL-CS-PSO 框架设计

本章将详细阐述 HGT-MARL-CS-PSO 框架的内部机制与实现细节。为了有效应对大规模云数据中心中“全局优化目标与局部资源竞争”的矛盾，以及强化学习在大规模离散动作空间中面临的“维度灾难”问题，本文提出了一种分层博弈与混合驱动相结合的系统架构。

4.1　总体架构

HGT-MARL-CS-PSO 框架旨在将复杂的云资源调度问题解耦为一个“宏观决策—微观博弈—混合执行”的闭环控制系统。如图 1 总体架构所示，该系统由三个逻辑层级组成，通过自上而下的控制流和自下而上的反馈流紧密耦合。

HGT-MARL-CS-PSO 的运行流程遵循一个严格的时间步机制，如图 2 架构时序图所示。在每个调度周期开始时，环境反馈当前的全局状态和局部状态；M-Agent 首先根据全局状态选择最优战略并广播给所有 E-Agents；随后，E-Agents 结合局部状态和接收到的战略，通过博弈策略网络输出优化参数；最后，CS-PSO 调度器加载这些参数，执行启发式搜索并输出具体的调度方案。环境执行该方案后产生新的性能指标（完工时间、能耗等），这些指标被转化为奖励信号，分别用于更新 M-Agent 和 E-Agents 的策略网络，推动系统向斯塔克伯格均衡点收敛。

4.2　宏观战略层

在 HGT-MARL-CS-PSO 框架的双层斯塔克伯格博弈模型中，M-Agent 被设计为系统的“大脑”，扮演着领导者的核心角色。不同于关注具体任务分配细节的底层执行者，M-Agent 的首要职责是进行宏观调控。它通过长周期的观测，识别云数据中心工作负载的潮汐特征，并动态引导整个系统在不同的优化目标之间进行战略切换，从而解决由于缺乏全局视野而导致的局部最优问题。其马尔可夫决策过程 MDP 的元组 $\langle \mathcal{S}_M,\mathcal{A}_M,\mathcal{R}_M\rangle$ 定义如下：

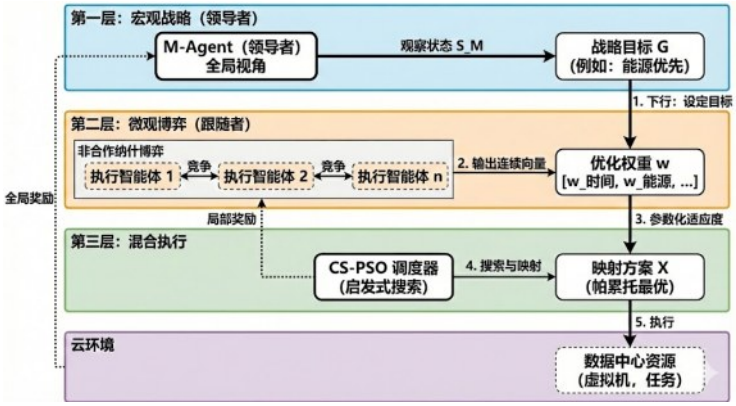

图 1　总体架构

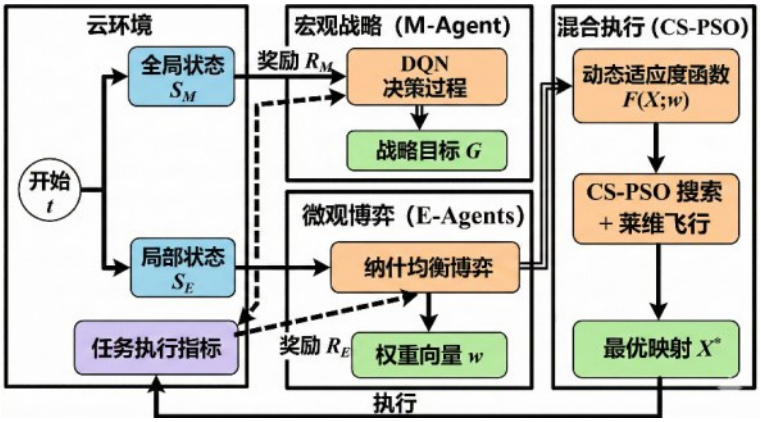

图2　架构时序图

全局状态空间 $\mathcal{S}_M$：为了制定有效的战略，M-Agent 需要感知整个云数据中心的宏观运行“脉搏”。本文将 t 时刻的全局状态空间定义为向量 $s_{M,t}\in\mathbb{R}^4$，该向量高度概括了系统的实时健康状况。

$$s_{M,t}=\{\bar{U}_{sys,t},\sigma_{load,t},\Delta E_t,Q_{len,t}\}$$

式中其中 $\bar{U}_{sys,t}$ 为全网物理机的平均 CPU 利用率，$\sigma_{load,t}$ 衡量节点间的负载方差，ΔE_t 表征当前周期的能耗增长率，$Q_{len,t}$ 为全局未处理任务的积压队列长度。

离散动作空间 $\mathcal{A}_M$：基于对宏观状态的感知，M-Agent 的决策动作 $a_{M,t}$ 并非直接干预底层的任务调度，而是从预定义的战略集合 $\mathcal{A}_M$ 中选择一个全局战略目标。该集合包含三种典型的云管理策略，性能优先策略 (G_{perf})、节能优先策略 (G_{energy}) 和负载均衡策略 $(G_{balance})$。例如在低负载期输出 G_{energy} 以触发能效整合。

全局奖励函数 $\mathcal{R}_M$：为了引导 M-Agent 学习到能够最大化系统长期利益的策略 π_M，本文设计了一个复合奖励函数 $R_{M,t}$。该函数并非单一目标的简单反馈，而是多重性能指标的加权权衡。

$$R_{M,t}=-\left(\alpha\frac{Mspan_t}{Mspan_{max}}+\beta\frac{E_{total,t}}{E_{max}}\right)+\gamma U_{sys,t}-\Phi(Q_{len,t})$$

由三部分组成，第一部分是对低效调度的惩罚，包括过长的完工时间和过高的能耗，这迫使智能体追求帕累托前沿。第二部分是对高资源利用率的奖励，鼓励智能体充分挖掘基础设施潜力。第三部分是基于 SLA 的软约束惩罚，一旦任务积压超过阈值 Q_{th}，将施加巨大的负反馈。通过最大化这一累积奖励，M-Agent 利用深度 Q 网络（DQN）算法不断试错与迭代，最终掌握在复杂动态环境中自适应切换最优战略的能力。

4.3　微观博弈层

在 HGT-MARL-CS-PSO 框架的底层，一组 E-Agents $\mathcal{E}=E_1,\ldots,E_K$ 构成了斯塔克伯格博弈的跟随者群体。每个 E-Agent 独立管理一个特定的资源集群或虚拟机池。与 M-Agent 的全局战略导向不同，E-Agents 的核心职责是在接收到上层下发的战略指令后，通过非合作纳什博弈（Non-Cooperative Nash Game）来优化各自的局部调度策略。这一层级的设计旨在解决大规模分布式环境下的资源争抢与负载失衡问题。

每个 E-Agent，E_k 仅能观测到其管辖范围内的局部环境信息。本文将 t 时刻第 K 个智能体的局部状态定义为向量 $s_{E,k,t}\in\mathbb{R}^3$：

$$s_{E,k,t}=\{u_{k,t}^{cpu},u_{k,t}^{mem},n_{k,t}\}$$

其中，$u_{k,t}^{cpu}$ 和 $u_{k,t}^{mem}$ 分别表示该资源池内所有虚拟机的平均 CPU 和内存利用率，$n_{k,t}$ 表示当前分配给该资源池的待处理子任务数量。这种去中心化的状态观测机制显著降低了单个智能体的输入

维度。

为克服海量任务分配时的“动作空间爆炸”难题，本文提出参数化控制机制。E-Agent 并不输出离散的映射动作，而是输出一个连续的优化权重参数向量 $\mathbf{w}_{k,t}$，用于驱动底层的 CS-PSO 调度器：

$$\mathbf{w}_{k,t}=[w_{time},w_{energy},w_{balance}]^T,\text{s.t.}\sum_i w_i=1,w_i\in[0,1]$$

该向量直接定义了底层启发式算法在当前时间步的搜索方向。例如，$\mathrm{w}_{k,t}=[0.8,0.1,0.1]$时，意味着 E-Agent 指示底层调度器以极高的优先级优化完工时间。这种将离散调度决策转化为连续参数控制的方法，不仅实现了动作空间的降维，还保留了底层元启发式算法在组合优化问题上的搜索优势。实验观测表明，$\mathbf{w}_{k,t}$ 的动态轨迹能够直观映射博弈的收敛过程。当系统趋向斯塔克伯格均衡（SE）时，$\mathbf{w}_{k,t}$ 会从早期的剧烈震荡逐渐锁定至特定区间的稳定吸引子，实现了从全局战略引导到局部资源竞价的逻辑闭环。

为了引导 E-Agents 在博弈中达到纳什均衡，本文设计了一个包含“服从”与“竞争”双重激励的混合奖励函数 $R_{E,k,t}$：

$$R_{E,k,t}=(1-\lambda)\cdot R_{M,t}(a_M)+\lambda\cdot\left(u_{k,t}^{cpu}-\bar{U}_{sys,t}\right)$$

第一项继承自 M-Agent 的全局奖励 $R_{M,t}$，确保局部优化不偏离全局战略；第二项为基于利用率差值的竞争项，激励智能体主动争夺任务，从而在微观层面消除“搭便车”现象。

鉴于 E-Agents 输出的是连续动作向量，且环境具有非平稳性，本文采用多智能体深度确定性策略梯度（MADDPG）算法进行训练。MADDPG 采用“集中式训练，分布式执行”架构：在训练阶段，每个 E-Agent 的评估网络能利用其他所有智能体的动作信息来评估状态价值，从而稳定梯度的更新；在执行阶段，执行网络仅根据局部状态 $s_{E,k,t}$ 独立输出权重参数 $\mathbf{w}_{k,t}$。通过这种方式，E-Agent 能够有效地学习到在复杂博弈环境下的最优联合策略。

4.4　混合执行层

单纯依赖强化学习输出离散动作易导致不收敛。为此，本文设计了基于改进 CS-PSO 算法的混合执行层。其核心创新在于参数化动态适应度函数：传统算法权重固定，而本框架的适应度函数 $F(X)$ 由 E-Agent 实时输出的权重向量 $\mathbf{w}_{k,t}=[w_{time},w_{energy},w_{balance}]^T$ 动态定义。对于任意映射矩阵 X，其寻优目标为 $MinimizeF(X;w_{k,t})$。在搜索机制上，本框架结合了 PSO 的快速局部开发与布谷鸟搜索的全局探索。粒子在标准位置更新的基础上，定期引入服从莱维分布的随机步长以跳出局部最优。最终，算法通过 SPV 规则将连续的粒子位置向量无缝离散化为任务-虚拟机的具体映射方案。

4.5　算法流程与复杂度分析

结合前述的分层博弈架构与参数化控制机制，HGT-MARL-CS-PSO 算法通过高层战略引导与底层微观寻优的耦合，实现对云资源的动态精密调度。其整体执行逻辑如算法 1 所示。

算法 1HGT-MARL-CS-PSO 动态调度执行流程

输入: 动态任务流 T，物理机集群 H，网络参数 $\theta_M,\phi_M,\theta_E,\phi_E$，最大回合数 E_{max}。

1: 初始化经验回放池 D_M,D_E 及所有目标（Target）网络参数；

2: for e = 1 to E_{max} do

3: 重置云仿真环境，获取初始全局状态 $S_{M,0}$；

4: for 每个调度步 $t=1$ to T_{step} do

5: M-Agent 观测 $S_{M,t}$，根据策略 π_M 生成宏观战略 $a_{L,t}\in\{G_{energy},G_{perf},G_{balance}\}$；

6: for 每个 E-Agent $E_k(k=1\ldots K)$ 并行执行 do

7: 观测局部状态 $s_{E,k,t}$ 并接收战略约束 $a_{L,t}$；

8: Actor 网络输出连续参数化权重 $\mathbf{w}_{k,t}=$

$[w_{time}, w_{energy}, w_{balance}]^T$；

9: // 微观寻优阶段

10: 以 $F(X; w_{k,t})$ 为目标函数，初始化 CS-PSO 种群 P；

11: 依据式(11)-(13)更新粒子速度与位置，并引入 莱维飞行扰动；

12: 通过 SPV 规则离散化生成局部最优调度子矩阵 X_k^*；

13: end for

14: 系统执行全局联合分配矩阵 $X_t^* = \bigcup_{k=1}^{K} X_k^*$；

15: 计算各智能体混合奖励 $R_{E,k,t}$ 与全局长期回报 $R_{M,t}$；

16: 将状态转移元组存入对应经验池 D_E 与 D_M；

17: if 满足更新条件 then

18: 从 D_E 采样 Mini-batch，依据式(9)-(10)更新 MADDPG 策略参数；

19: end if

20: end for

21: 定期执行 M-Agent 的网络更新及所有 Target 网络的软更新；

22: end for

23: return 最优映射矩阵 X^*；

HGT-MARL-CS-PSO 的计算复杂度主要由神经网络推理和 CS-PSO 搜索组成，M-Agent 和 E-Agents 的推理基于简单的前馈神经网络。设网络层数为 L，最大神经元数为 H，则单次推理复杂度为 $O(L \cdot H^2)$。由于 H 通常较小（如 64 或 128），这部分开销在毫秒级，记为 $O(1)$。CS-PSO 的复杂度取决于粒子数 P、最大迭代次数 I_{max} 以及适应度评估的成本。适应度评估需遍历所有任务 N 和虚拟机 M，复杂度为 $O(N \cdot M)$。因此，CS-PSO 的总复杂度为 $O(I_{max} \cdot P \cdot N \cdot M)$。

综上，单个时间步的总时间复杂度为：

$$T_{total} \approx O(K \cdot L \cdot H^2) + O(I_{max} \cdot P \cdot N \cdot M) \approx O(N \cdot M)$$

由于 I_{max} 和 P 为常数，算法的复杂度与任务规模呈在固定资源规模 M 时对任务数 N 的线性相关。相比于传统组合优化算法（如分支定界法）的指数级复杂度 $O(M^N)$，HGT-MARL-CS-PSO 在处理大规模调度问题时具有显著的效率优势，能够满足云数据中心实时调度的响应需求。

4.6 理论收敛性分析

为了验证系统的稳定性，本节简要论证 HGT-MARL-CS-PSO 框架的收敛性。首先，在底层微观博弈中，由于执行智能体 E_k 的效用函数 $U_{F,k}$ 在连续动作空间上满足准凹性，根据卡库塔尼不动点定理，E-Agents 之间必然存在纳什均衡点 $W^*(a_L)$。其次，高层 M-Agent 将跟随者的平衡响应视为环境状态转移的一部分，由于其采用的 DQN 算法在满足贝尔曼收缩映射条件下具有稳健的随机逼近特性，管理策略能够收敛至最优战略 a_L^*。综上，当高层战略与底层响应达成一致时，系统收敛至稳定的斯塔克伯格均衡点，确保了动态调度过程中的策略鲁棒性。

5 实验仿真与分析

5.1 实验设置

本文构建了包含 200 台异构物理机（4 种规格，涵盖 2~16 核 CPU 与 4~32 GB 内存）及 1000 台虚拟机的云仿真环境，并与 PyTorch 框架实现端到端集成。系统能耗采用基于 CPU 利用率的线性动态功耗模型进行精确计算（硬件及功耗参数详见表 1 实验仿真环境详细参数配置表）。实验负载采用 Google Borg Cluster Trace (2019) [20]真实数据集。对原始日志剔除异常记录后，随机截取含 5000 至 10000 个任务的连续时间窗口，并将任务资源需求归一化映射至仿真维度，以全面检验算法应对强异构与高并发潮汐负载的鲁棒性。

本文选取三类共 8 种算法作为对比基准：传统启发式（FCFS、Min-Min）、元启发式（GA、

表 1　实验仿真环境详细参数配置表

资源层级	性能参数项	具体配置与取值范围
物理主机 (Host)	节点总数	200 台
	CPU 核心规格	Type 1: 2 核; Type 2: 4 核; Type 3: 8 核; Type 4: 16 核
	内存容量 (RAM)	4GB / 8GB / 16GB / 32GB
	功耗模型	P^{idle}: 100W~200W; P^{peak} : 350W~500W
虚拟机 (VM)	实例总数	1000 台
	处理能力 (MIPS)	500 ~ 2500 MIPS (异构分布)
	资源配额	CPU 份额 (1~4 vCPUs), 内存 (512MB~4GB)
工作负载 (Task)	数据来源	Google Borg Cluster Trace (2019)
	任务特征	计算长度: 1,000 ~ 50,000 MI; 内存需求: 128MB ~ 2GB

ACO）以及深度强化学习基线（DQN-CS-PSO、MARL-DQN、MARL-QLearning）。为验证架构设计的必要性，设置 HMARL（无博弈层）与 GT-MARL（无宏观引导）两组消融变体。

在参数配置上，RL 模块基于 PyTorch 实现，学习率设为 10^{-4}，折扣因子 0.99，经验池容量 10^5，博弈竞争系数 0.4。底层 CS-PSO 调度器配置粒子数 $P=50$，最大迭代次数 $I=100$，并引入步长因子 0.01 的莱维飞行机制。预实验表明，$P=50$ 能有效平衡全局寻优质量与实时响应延迟（$P>50$ 时解的质量呈边际效应递减），且在高层 RL 动态权重引导下，底层通常在 85 次迭代内达到适应度平台，设定 $I=100$ 足以为极端负载寻优预留演化空间。

需要特别说明的是，由于深度强化学习（DRL）的策略探索和元启发式算法（CS-PSO）的种群初始化均包含随机机制，为了消除单次运行的偶然性，本文所有的评估实验均在不同的随机种子和不同的动态负载时间窗口下独立重复运行了 10 次。后文中所汇报的完工时间、资源利用率和总能耗等核心性能指标，除非特别标注，均指代这 10 次独立重复实验的平均值。实验主要关注三个核心性能指标：完工时间(Makespan)用于衡量系统吞吐效率，总能耗(Total Energy)用于评估绿色计算能力，而平均资源利用率(Resource Utilization)则反映了资源分配的紧凑程度与负载均衡水平。

5.2　总体性能对比

为了验证 HGT-MARL-CS-PSO 框架在解决多目标优化冲突方面的有效性，我们将该算法在完工时间、资源利用率及综合性能三个维度上与传统启发式算法（FCFS,Min-Min）、元启发式算法（GA,ACO）及标准深度强化学习方法进行了详细对比。

5.2.1　完工时间效率分析

完工时间是衡量云系统处理真实并发任务流效率的核心指标。如图 3 不同调度算法的总完工时间对比所示，HGT-MARL-CS-PSO 的总完工时间为 112.05s，较 FCFS 和 Min-Min 分别提升了 39.5% 和 30.0% 的处理效率 。与具备全局搜索能力的 GA (145.91s) 和 ACO (140.23s)相比，本文方法依然缩短了约 20% 的时间。这证明了“RL 决策参数+元启发式执行”机制能比传统进化算法更快锁定异构负载下的最优映射方案。

5.2.2　资源利用率分析

在资源利用率方面如图 4 不同调度算法的平均资源利用率对比，本文框架达到了 88.17% 的高水平，显著优于单智能体 DQN-CS-PSO (78.92%)，体现了多智能体协同在消除资源碎片方面的优势 。在能耗控制上如图 5 不同调度算法的能耗对比所示，由于 M-Agent 能够根据负载潮汐动态切换战略，系统总能耗仅为 0.55kWh，较

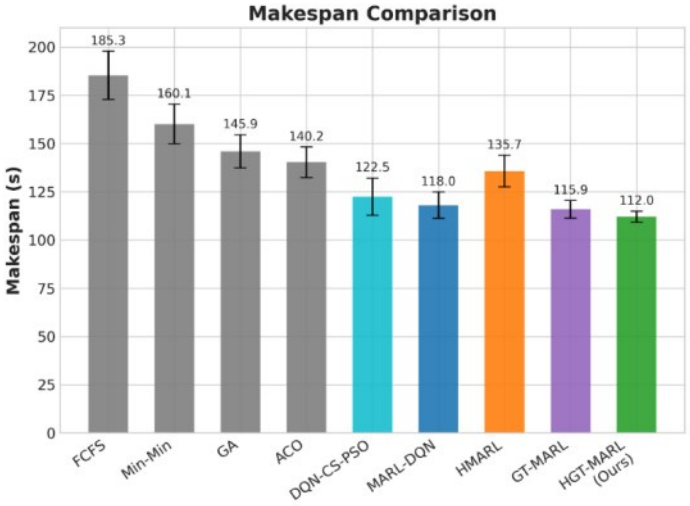

图3　不同调度算法的总完工时间对比

FCFS 节能 56%，较表现较好的 MARL-DQN 进一步降低了 9.8% 的能耗。

5.3　消融实验与机制分析

本文通过对比完整框架与 HMARL（无博弈）、GT-MARL（无分层）及单智能体 DQN-CS-PSO 基线，从多个维度分析 HGT 架构的必要性。

5.3.1　博弈机制与分层结构的协同贡献

实验数据表明，移除博弈机制（HMARL）会导致完工时间增至 135.67s，资源利用率从 88.17% 骤降至 68.90%。这揭示了纳什博弈中“局部竞争项”对消除资源碎片的关键驱动作用，缺乏竞争机制会导致智能体陷入协作失效。对比仅含博弈的 GT-MARL 发现，虽然其局部利用率较高，但在缺乏 M-Agent 宏观战略引导下，系统无法实现多目标的帕累托最优平衡。HGT-MARL 在 GT-MARL 基础上进一步提升了 3.61% 的利用率并降低了 5.2% 的能耗，证明了分层引导在全局优化中的核心价值。

5.3.2　分布式架构对维度灾难的抑制

对比单智能体 DQN-CS-PSO 发现，其完工时

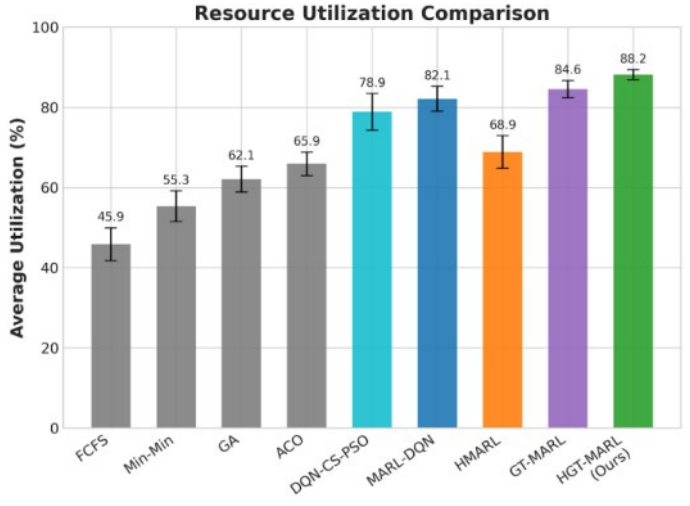

图4　不同调度算法的平均资源利用率对比

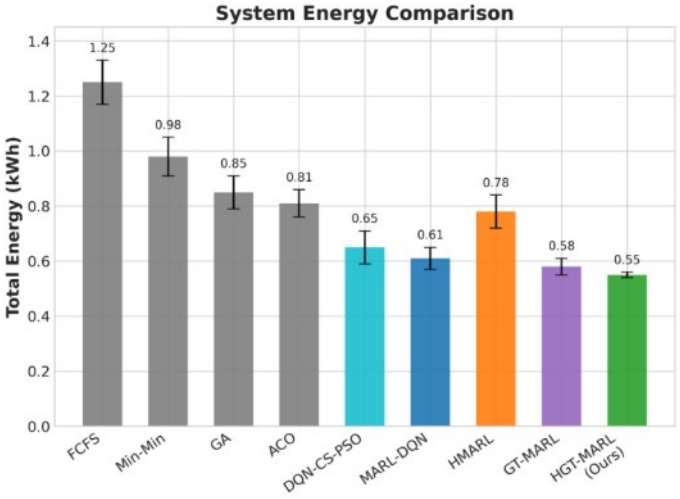

图5　不同调度算法的能耗对比

间较本文方法慢9.3%，能耗高出18.2%。这一差距源于大规模云环境下状态-动作空间的指数级膨胀（维度灾难），导致集中式架构难以在毫秒级时间内锁定最优决策。HGT-MARL通过“分而治之”策略，将全局复杂决策解耦为M-Agent的宏观战略与E-Agents的微观寻优，显著降低了单个智能体的学习难度，实现了对传统单智能体方法的全面超越。

5.4　算法收敛性与稳定性分析

为验证混合参数化机制对缓解“维度灾难”的贡献，各框架的全局奖励演化轨迹如图6基于实际算法运行的全局奖励收敛曲线对比所示（实线为滑动平均，浅色为原始数据）。

单智能体DQN-CS-PSO（蓝色曲线）的收敛过程极其缓慢，且在整个150个训练回合内表现出极大的不稳定性，未能达到理想的奖励峰值。

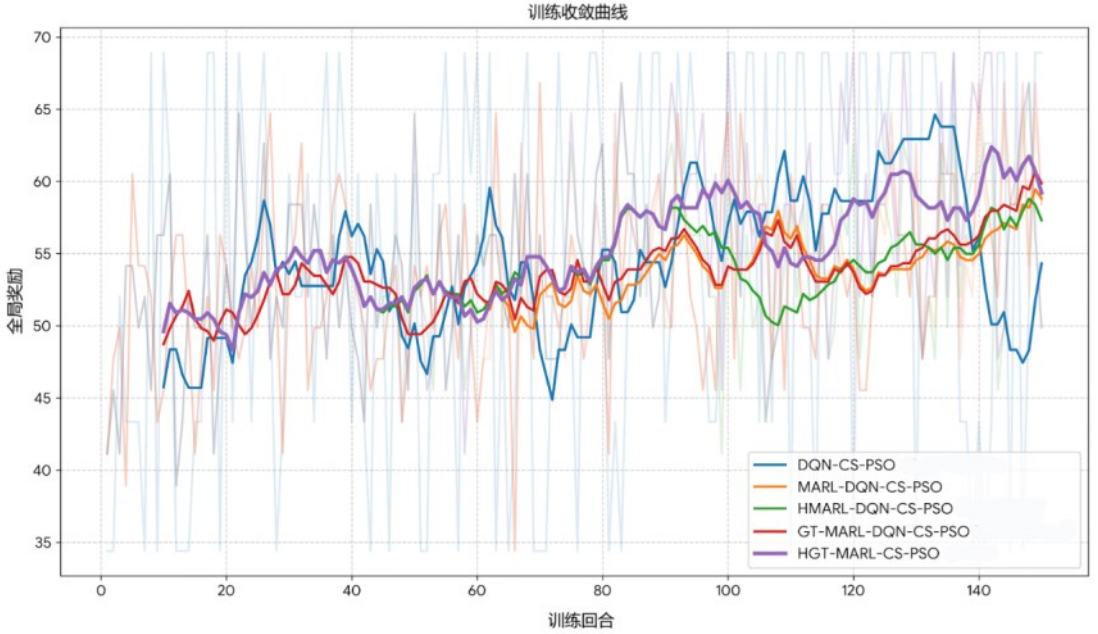

图6　基于实际算法运行的全局奖励收敛曲线对比

这深刻反映了“维度灾难”的负面影响：随着云数据中心虚拟机和任务规模的增加，单智能体面临着呈指数级爆炸的离散动作空间，导致其在庞大的状态-动作空间中进行盲目的随机探索，难以快速锁定高回报区域。 相反，HGT-MARL-CS-PSO（紫色加粗曲线） 展现出了显著的收敛速度优势。在训练的前40个回合内，其奖励值迅速攀升，并率先进入稳定收敛期，最终收敛于最高奖励水平。这一飞跃性的提升归功于本文提出的混合参数化控制机制：执行智能体不再直接输出高维的离散任务映射矩阵，而是输出仅含低维连续超参数的权重向量。这种从指数级离散空间到低维连续空间的降维转换，极大地缩小了强化学习的探索范围，使得模型能够更快地学习到最优策略。

除了收敛速度，策略的稳定性也是大规模云调度系统的关键。缺乏博弈机制的HMARL-DQN-CS-PSO（绿色曲线）和标准MARL-DQN-CS-PSO（橙色曲线）虽然由于多智能体的分布式架构缓解了部分维度问题，但其训练曲线在后期表现出明显的震荡（High Variance）。这是因为在缺乏有效竞争协调机制的传统MARL中，多个智能体同时更新策略会导致环境表现出强烈的非平稳性，智能体之间极易陷入资源争抢或策略循环。 相比之下，HGT-MARL-CS-PSO的训练曲线在到达顶峰后表现得最为平滑且稳健。这证明了斯塔克伯格博弈模型在多智能体系统中起到了关键作用，M-Agent 的宏观战略引导配合 E-Agents 的纳什博弈竞争，使得系统能够有效克服多智能体间的策略冲突与“搭便车”效应，最终平稳收敛至全局最优的斯塔克伯格均衡点。

6 结束语

针对大规模云数据中心资源调度中存在的“多目标冲突”与“离散动作空间爆炸”两大难题，本文提出了一种基于分层博弈论的多智能体强化学习框架——HGT-MARL-CS-PSO。该框架创新性地将复杂的调度问题解耦为宏观战略制定、微观资源博弈与底层启发式执行三个紧密耦合的层级，通过斯塔克伯格博弈模型实现了全局优化目标与局部资源竞争的动态平衡。我们在架构设计上突破了传统多智能体强化学习（MARL）的局限，提出了一种“RL 决策参数+CS-PSO执行搜索”的混合驱动机制。该机制通过让智能体输出连续的权重向量而非离散的映射动作，成功将动作空间从随任务数量指数级增长的维度降至常数级，有效规避了大规模调度场景下的维度灾难问题。

在机制层面，本文引入了基于非合作纳什均衡的微观博弈机制，不仅解决了多智能体协作中的激励难题，更通过在奖励函数中引入局部竞争项，迫使E-Agents主动争夺任务以消除资源碎片。在基于Google Borg Trace真实数据集的大规模实验中，HGT-MARL-CS-PSO展现了卓越的综合性能。与经典的遗传算法(GA)和Min-Min算法相比，本文方法分别降低了23.2%和30.0%的完工时间；同时，资源利用率高达88.17%，显著优于同类分层算法HMARL(68.90%)。此外，该框架将总能耗控制在0.55kWh的最低水平，相比于基准算法实现了显著的绿色节能效果，验证了其在解决大规模异构云资源调度问题上的鲁棒性与优越性。

尽管HGT-MARL-CS-PSO在当前实验环境下表现优异，但面对云计算技术的快速演进，仍有若干值得深入探索的研究方向。首先，我们将致力于将该框架从单一数据中心扩展至“云-边-端”协同的复杂环境。在该演进过程中，需重点解决两个核心问题，极端异构性下的跨域博弈协调和网络不确定性下的策略鲁棒性，边缘设备与云端服务器在计算能力、带宽延迟及能源约束上存在量级差异 。如何在维持斯塔克伯格均衡的同时，建立一套能够跨越物理域且具备动态隐私保护的

层级博弈激励模型，是实现低延迟任务调度的关键。边缘侧网络拓扑具有高动态性，且链路带宽波动显著 。研究如何增强MARL 智能体对网络异常及节点移动性的感知能力，并通过引入分布回归或鲁棒博弈机制来保障调度方案在链路受损时的可靠性，是未来迈向工程化应用的重要一步 。

诸葛斌（1976- ），男，博士，浙江工商大学信息与电子工程学院教授，主要研究方向为网络和通信技术、互联网技术和网络安全。

参考文献：

[1] Buyya R, Yeo C S, Venugopal S, et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility [J]. Future Generation Computer Systems, 2009, 25(6): 599-616.

[2] Kumar M, Sharma S C, Goel A, et al. A comprehensive survey for scheduling techniques in cloud computing [J]. Journal of Network and Computer Applications, 2019, 143: 1-33.

[3] Ullman J D. NP-complete scheduling problems [J]. Journal of Computer and System Sciences, 1975, 10(3): 384-393.

[4] Etminani K, Naghibzadeh M. A Min-Min Max-Min selective algorithm for grid task scheduling [C]//2007 3rd IEEE/IFIP International Conference on Central Asia on Internet. IEEE, 2007: 1-7.

[5] Jena U K, Das P K, Kabat M R. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment [J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(6): 2332-2342.

[6] Mao H, Alizadeh M, Menache I, et al. Resource management with deep reinforcement learning [C]//Proceedings of the 15th ACM Workshop on Hot Topics in Networks. 2016: 50-56.

[7] 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.

[8] Cheng M, Li J, Nazarian S, et al. Multi-agent reinforcement learning for task scheduling in large-scale cloud data centers [J]. Future Generation Computer Systems, 2023, 145: 112-125.

[9] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [C]//Advances in Neural Information Processing Systems. 2017: 6379-6390.

[10] Krishnamurthy B, Shiva S G. Scalable Resource Provisioning Framework for Fog Computing Using LLM-Guided Q-Learning Approach [J]. Algorithms, 2025, 18(4): 230.

[11] Wu Y, et al. Task Scheduling in Geo-Distributed Computing: A Survey [J]. arXiv preprint arXiv:2501.15504, 2025.

[12] Wang T, Liang Y, Jia Y, et al. Stackelberg game based resource allocation in edge computing with renewable energy [J]. IEEE Internet of Things Journal, 2020, 7(8): 7306-7318.

[13] Hady M A, et al. Multi-Agent Reinforcement Learning for Resources Allocation Optimization: A Survey [J]. arXiv preprint arXiv:2504.21048, 2025.

[14] Krishnamurthy B, Shiva S G. Large Language Model-Guided SARSA Algorithm for Dynamic Task Scheduling in Cloud Computing [J]. Mathematics, 2025, 13(6): 926.

[15] Moura J, Hutchison D. Game Theory for Multi-Access Edge Computing: Survey, Use Cases, and Future Trends [J]. IEEE Communications Surveys & Tutorials, 2019, 21(1): 260-288.

[16] Liu N, Li Z, Xu J, et al. A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning [C]// 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 372-382.

[17] Houssein E H, Gad A G, et al. Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends [J]. Swarm and Evolutionary Computation, 2021, 62: 100841.

[18] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers [J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[19] Yang X S, Deb S. Cuckoo search via Lévy flights [C]//2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE, 2009: 210-214.

[20] Tirmazi M Y, Barker A, Deng N, et al. Borg: the next generation [C]//Proceedings of the Fifteenth European Conference on Computer Systems. 2020: 1-14.