



XXXX

面向跨区域云-边协同的资源激励型异步任务调度

程冠杰¹, 王睿昊¹, 陈益杉², 赵周行³, 赵新奎¹, 丁智旸⁴

1. 浙江大学软件学院, 浙江 宁波 315000;
2. 江西理工大学信息工程学院, 江西 赣州 341000;
3. 浙江省通信产业服务有限公司, 浙江 杭州 310000;
4. 西北工业大学教育实验学院, 陕西 西安 710129)

摘要: 针对跨区域云-边多数据中心系统 (multi-data center system, MDCS) 中节点异构、流量突发及网络时延不确定导致的协同难题, 提出了一种基于资源激励型 Stackelberg 博弈 (resource incentive Stackelberg game, RISG) 的异步任务调度方法。首先, 利用 M/M/1 排队理论刻画异构节点在高并发下的非线性拥塞效应, 构建融合能耗、时延、传输开销及可靠性风险的综合成本模型, 并推导出数据中心最优响应的闭式解。其次, 设计了基于梯度的异步坐标下降 (asynchronous coordinate descent, ACD) 算法, 支持全局调度器利用陈旧信息进行非阻塞式策略更新, 克服了同步等待的低效性, 并证明了算法的收敛性与均衡的存在唯一性。仿真结果表明, ACD 算法收敛速度较传统异步梯度投影算法提升约 26.9%; 在系统重载场景下, 平均响应时延较贪婪策略降低约 45.17%; 在强网络抖动环境下, 社会总成本波动误差控制在 8% 以内, 验证了该方法的高吞吐量与强鲁棒性。

关键词: 多数据中心系统; 异步任务调度; 主从博弈

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.

Resource-incentivized-based asynchronous task scheduling for cross-regional cloud-edge collaboration

CHENG Guanjie¹, WANG Ruihao¹, CHEN Yishan², ZHAO Zhouxing³, ZHAO Xinkui¹, DING Zhiyang⁴

1. School of Software Technology, Zhejiang University, Ningbo 315000, China

2. School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

3. Zhejiang Communications Industry Service Co., Ltd., Hangzhou 310000, China

4. Honors College, Northwestern Polytechnical University, Xi'an 710129, China

Abstract: To address the collaborative challenges caused by node heterogeneity, traffic bursts, and network latency

收稿日期: XXXX-XX-XX; 修回日期: XXXX-XX-XX

通信作者: 陈益杉, chenys@jxust.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62502446)

Foundation Items: The National Natural Science Foundation of China (No.62502446)



uncertainty in cross-regional cloud-edge multi-data center systems (MDCS), this paper proposes an asynchronous task scheduling method based on a Resource Incentive Stackelberg Game (RISG). First, M/M/1 queuing theory is utilized to characterize the non-linear congestion effects of heterogeneous nodes under high concurrency, a comprehensive cost model integrating energy consumption, latency, transmission overhead, and reliability risk is constructed, and a closed-form solution for the optimal response of data centers is derived. Second, a gradient-based Asynchronous Coordinate Descent (ACD) algorithm is designed, which supports the global scheduler in performing non-blocking policy updates using stale information, overcoming the inefficiency of synchronous waiting, and the convergence of the algorithm as well as the existence and uniqueness of the equilibrium are proven. Simulation results show that the convergence speed of the ACD algorithm is improved by approximately 26.9% compared to the traditional asynchronous gradient projection algorithm; in heavy-load system scenarios, the average response latency is reduced by approximately 45.17% compared to greedy strategies; in environments with strong network jitter, the total social cost fluctuation error is controlled within 8%, verifying the high throughput and strong robustness of this method.

Key words: multi-data center system, asynchronous task scheduling, leader - follower game

1 引言

以生成式人工智能^[1]、元宇宙^[2]及工业数字孪生^[3]为代表的新兴应用具有计算密集与时延敏感双重特性，打破了单一数据中心的吞吐极限，暴露了传统集中式云计算在带宽和时延上的局限性。为此，计算范式正向云-边协同演进，旨在通过广域网构建物理分散而逻辑统一的算力网络^[4]，实现跨地域资源的高效利用。作为该架构的核心载体，跨区域多数据中心系统（multi-data center system, MDCS）在复杂环境下的任务调度与资源编排面临着严峻挑战^[5]，主要集中在以下三个维度：(1) 节点异构性与非线性拥塞的叠加效应：MDCS节点在算力与能效上差异巨大，且面对突发流量时响应时间不随负载线性增长。当负载逼近容量极限时，排队时延呈指数级恶化^[6]。传统静态调度模型忽视了这一非线性规律，易导致重载场景下服务等级协议（service level agreement, SLA）违约风险急剧攀升。(2) 广域网环境下的通信不确定性与异步瓶颈：跨域调度的状态同步受限于广域网长时延与高抖动等不可控因素。传统依赖严格同步（Stop-and-Wait）的调度机制，极易引发“掉队者效应”（straggler effect），即系统整体迭代速率被最慢节点锁定，导

致算力空转并大幅降低系统吞吐量。(3) 全局最优与局部自利的博弈冲突：在多主体的算力网络生态中，各数据中心作为理性经济实体，决策上优先追求个体收益最大化或成本最小化。这种个体自利与系统全局总运营成本优化的内生性背离，极易使调度策略陷入低效的纳什均衡。

鉴于此，本文提出一种基于资源激励型 Stackelberg 博弈（resource incentive Stackelberg game, RISG）的异步任务调度方法。核心思想在于利用广域网的异步特性，构建一种能够容忍陈旧信息的梯度下降机制，在确保系统全局收敛的同时，最大化调度吞吐量。与现有工作相比，本文的主要贡献如下。

(1) 构建基于 M/M/1 排队论的非线性效用博弈模型：有别于广泛采用的线性成本假设^[7-8]，本文引入 M/M/1 理论刻画异构节点高并发下的物理拥塞特性，推导融合能耗、时延及风险的综合成本函数。该模型赋予了系统内生性的拥塞规避能力，克服了线性模型在重载下失效的问题。

(2) 提出容忍陈旧信息的异步坐标下降（asynchronous coordinate descent, ACD）算法：突破了传统集中式框架对全局一致性的依赖，允许调度器利用“陈旧梯度”进行非阻塞式策略迭代。理论与仿真实证，该算法在保证收敛至唯一

Stackelberg 均衡的同时，显著提升了系统的收敛速度。

(3) 实现激励相容的全局协同：本文将全局能效目标解耦为价格引导的个体效用最大化问题，实现了激励相容。引入的风险惩罚因子使系统具备感知道路风险并向云端迁移任务的自适应鲁棒性，有效解决了分布式智能体难以达成全局共识的难题。

本文第 2 节综述相关工作；第 3 节构建基于 M/M/1 排队论的节点拥塞与成本模型，并形式化 Stackelberg 博弈；第 4 节分析博弈均衡并设计 ACD 算法；第 5 节进行仿真验证；最后总结全文。

2 相关工作

随着算力网络的推进，多数据中心任务调度正向动态智能博弈与异步优化演进。本节从分布式/多智能体调度、集中式/混合型调度以及博弈论激励机制三个维度对近年的前沿成果进行综述。

2.1 基于多智能体强化学习的分布式调度

分布式调度通过决策权下放以缓解中心负荷，多智能体强化学习^[9-11]为其主流范式。现有研究通过集中训练分布式执行架构、引入长短期记忆网络应对非平稳环境^[12]、设计分层卸载的深度强化学习^[13]，以及结合 Transformer 预测任务流^[14]等智能手段，有效提升了动态复杂负载下的资源利用率并降低了响应时延。

然而，MARL 在跨区域多数据中心场景中面临落地瓶颈，在通信受限的异步环境中，去中心化智能体不仅收敛困难，且极易陷入策略振荡。更关键的是，纯分布式方法因缺乏全局视角的宏观调控，易导致局部资源热点，难以实现全网的最优负载均衡。

2.2 集中式与混合架构调度

集中式调度^[15-16]凭借全局视图，在优化宏观

成本与负载均衡方面具天然优势。借助软件定义网络^[17]，其策略正从传统贪婪^[7,18-20]或轮询算法^[21-22]，转向基于深度强化学习的智能决策^[23]。为平衡开销与响应速度，“中心决策、边缘执行”的混合架构逐渐兴起^[5]。然而，此类架构在广域网部署中面临严峻的“同步壁垒”。传统算法依赖严格的全局同步机制，极易受网络长时延与高抖动影响，导致系统吞吐量被最慢的“掉队者”节点锁定。因此，调度机制向非阻塞的异步化转型势在必行。

2.3 基于 Stackelberg 博弈的激励机制

针对多数据中心归属不同利益主体的现状，契合“调度器-执行者”层级结构的 Stackelberg 博弈被广泛应用。现有研究已通过该模型成功构建了车载边缘网络资源定价、多主体综合能源博弈调度^[24]及动态边缘资源分配^[8]等框架，并证明了均衡解在激励相容方面的有效性。

尽管理论完备，现有博弈模型仍存两大局限：一是绝大多数基于理想化的同步交互假设，无法适应跨区域 MDCS 的长时延特性；二是对节点排队成本的刻画过于简化（多假设为线性），未能真实反映高并发负载下服务质量（quality of service, QoS）因拥塞而急剧衰退的非线性物理规律。

3 系统模型

本文所提出的场景如图 1 所示。该场景在逻辑上纵向划分为三个功能平面：中部的任务生成平面由智慧城市中的智能工厂、移动用户及 IoT 传感器等终端设备构成，这些分散的任务源汇聚成具有随机到达特性的总任务流并上传至控制中心；左侧的全局管控平面以部署于核心网侧的全局资源调度器（global resource broker, GRB）为核心，其作为博弈的领导者负责感知全网状态并运行 ACD 算法，向各节点下发价格激励信号；顶部及右侧的异构执行平面则涵盖了分布在不同

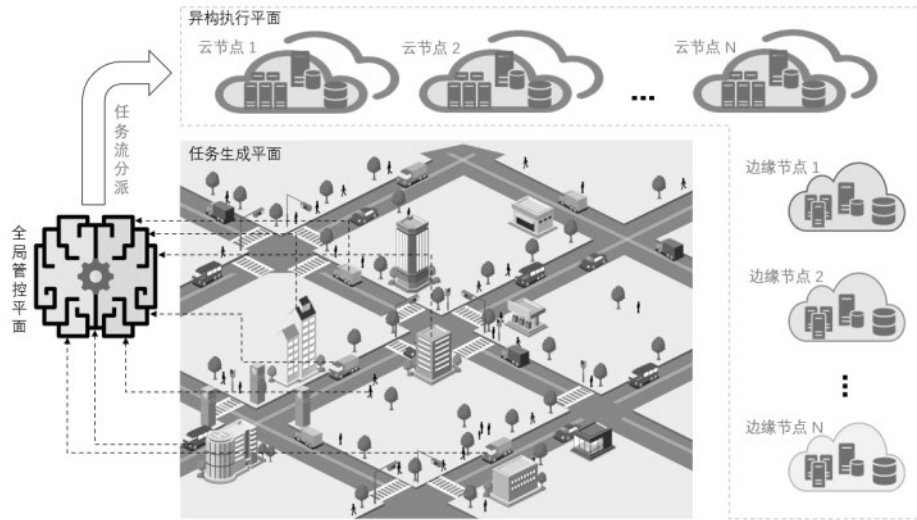


图1 跨区域云-边协同多数据中心系统的分层物理架构

地理维度的计算资源，包括拥有海量算力但距离较远的云节点服务器，以及贴近用户侧具备低时延优势的边缘节点服务器。全局资源调度器与各分布式节点之间通过广域网建立通信连接，这种交互拓扑虽然简化了管理路径，但也引入了不可忽视的随机通信时延，这构成了本文异步调度机制的物理基础。

本文所使用的符号及含义如表1所示，系统架构图如图2所示。

3.1 系统架构与实体定义

3.1.1 资源激励机制

针对跨区域MDCS中节点归属不同利益主体且具自利性的非合作特征，传统指令式调度极易失效。为此，本文引入资源激励机制。

定义1（资源激励机制）：资源激励机制（re-

source incentive mechanism）是指在分布式环境中，通过引入定价策略作为软性调控手段，将全局优化目标转化为微观经济信号的一种非合作博弈协调机制。

在该机制下：

(1) 全局调度层面：不强制指定任务的物理位置，而是发布动态的资源奖励价格，利用价格杠杆引导流量从高拥塞、高成本节点向低拥塞、低成本节点自发流动。

(2) 局部执行层面：各数据中心被建模为理性的经济实体，其决策逻辑遵循个体理性，即仅在预期获得的资源奖励能够覆盖其计算能耗、排队时延及可靠性风险等综合边际成本时才主动贡献算力资源。

通过资源激励机制，系统能够将宏观的社会

表1 符号及含义

符号	含义与解释
A, N, k	数据中心集合、系统数据中心总数及索引下标($k \in \{1, 2, \dots, N\}$)
$\Lambda_{total}, F_k, \rho_{sys}$	系统总任务负载、数据中心最大服务速率、系统平均负载率
R, R_k	全局资源奖励价格向量 $R = \{R_1, \dots, R_N\}$ 及对数据中心 A_k 的单位奖励价格
λ_k, λ_k^*	数据中心承诺承接的任务流率及其针对价格的最优响应
$\omega_e^k, \omega_{bw}^k, \theta_k, \omega_d^k$	数据中心 A_k 的单位计算能耗系数、带宽传输成本系数、可靠性风险系数、时延敏感系数
Ω_k	数据中心综合边际基准成本($\Omega_k = \omega_e^k + \omega_{bw}^k + \theta_k$)
R_{th}^k	数据中心 k 的启动阈值

总成本最小化问题解耦为各节点针对价格信号的局部效用最大化问题，从而在保障系统服务可用性的同时实现激励相容。

3.1.2 全局资源调度器

部署于核心网侧的全局资源调度器汇聚总任务负载 Λ_{total} ，在博弈模型中扮演领导者 (leader) 角色。其不强制指派任务，而是通过发布资源奖励价格向量 $R = \{R_1, R_2, \dots, R_N\}$ 引导底层资源分配，旨在满足全网服务质量约束的前提下，最小化系统的社会总成本 (social cost)。

3.1.3 分布式数据中心

系统包含 N 个异构的云-边数据中心节点，集合表示为 $A = \{A_1, A_2, \dots, A_N\}$ ，在博弈中扮演跟随者 (follower) 角色。节点 $A_k (k \in [1, N])$ 的异构处理上限由最大服务速率 F_k 唯一刻画。作为理性经济实体，各节点根据发布的奖励价格 R_k 与自身当前负载状态，自主决定承诺承接的任务速率 λ_k ，以最大化自身的净效用。

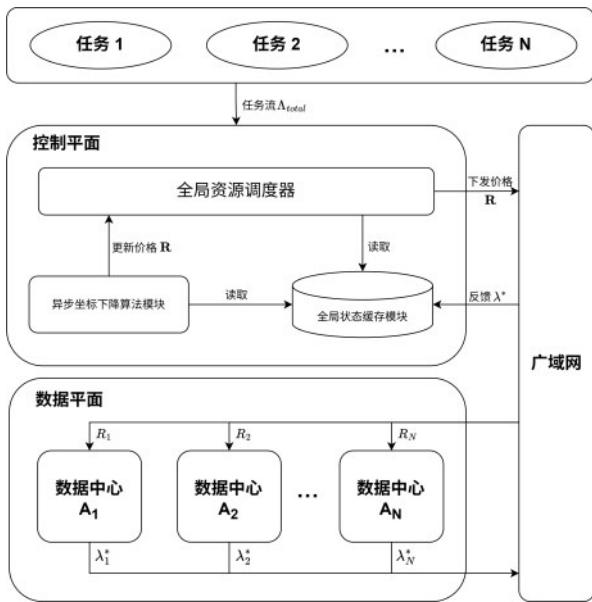


图2 系统架构图

3.2 任务流与排队时延模型

为准确刻画 MDSCS 中任务随机到达及节点响应性能随负载非线性恶化的现象，本节基于排队

论构建动态时延模型。

3.2.1 泊松任务流到达过程

设稳态时间窗口内外部任务到达遵循泊松分布，总负载为 Λ_{total} 。调度器将总负载拆解为连续流分量 λ_k 分配给各数据中心 A_k 。为保障任务完整性，系统须满足流量守恒约束：

$$\sum_{k=1}^N \lambda_k \geq \Lambda_{total}, \lambda_k \geq 0 \#(1)$$

3.2.2 异构服务能力与 M/M/1 排队模型

结合任务到达与大小的随机性，将各数据中心 A_k 建模为独立的 M/M/1 排队系统，其综合处理能力上限由最大服务率 F_k 刻画。根据 Little 定律，当分配的任务流率 $\lambda_k < F_k$ 时，任务的平均期望响应时间可表示为：

$$T_k(\lambda_k) = \frac{1}{F_k - \lambda_k} \#(2)$$

该非线性公式能够内生地惩罚过载行为，迫使流量自发向低负载节点溢出以实现软拥塞控制。

3.2.3 广域网环境下的综合运营成本函数定义

为克服传统模型仅考虑能耗与时延的局限，定义数据中心 A_k 处理流率 λ_k 的综合运营成本 $C_k(\lambda_k)$ 涵盖以下四个维度：

(1) 计算能耗成本。单位计算能耗系数设为 ω_e^k 。

(2) 排队拥塞惩罚。设 ω_d^k 为时延敏感系数。基于 M/M/1 模型，系统平均滞留任务数 $L_k = \frac{\lambda_k}{F_k - \lambda_k}$ 直观量化了高负载下的拥塞程度。

(3) 广域网带宽成本。单位数据传输成本系数设为 ω_{bw}^k ，刻画通信与计算的权衡。

(4) 节点可靠性风险成本。单位风险惩罚系数设为 θ_k ，赋予调度机制规避高故障率节点的自适应能力。

增强后的综合成本函数形式化为：



$$C_k(\lambda_k) = \left(\omega_e^k \lambda_k + \omega_{bw}^k \lambda_k + \theta_k \lambda_k \right) + \omega_d^k \frac{\lambda_k}{F_k - \lambda_k} = \Omega_k \lambda_k + \omega_d^k \frac{\lambda_k}{F_k - \lambda_k} \#(3)$$

其中, $\Omega_k = \omega_e^k + \omega_{bw}^k + \theta_k$ 为数据中心 A_k 的边际基准成本。

3.3 效用模型

在博弈模型中, 各数据中心 A_k 作为独立理性的跟随者, 通过权衡任务奖励与运营成本以最大化自身效用。

3.3.1 决策变量与参数定义

给定全局调度器下发的单位资源奖励价格 R_k , A_k 的核心决策变量为承诺承接的任务流率 λ_k 。为保证物理可行性, 必须满足服务能力约束:

$$0 \leq \lambda_k < F_k \#(4)$$

3.3.2 效用函数构建

数据中心 A_k 的效用函数 $U_k(\lambda_k)$ 为总奖励收入与综合运营成本之差 (即 $U_k(\lambda_k) = R_k \lambda_k - C_k(\lambda_k)$)。代入前文成本函数, 形式化为:

$$U_k(\lambda_k) = (R_k - \Omega_k) \lambda_k - \frac{\omega_d^k \lambda_k}{F_k - \lambda_k} \#(5)$$

其中, $(R_k - \Omega_k) \lambda_k$ 表征扣除基础成本后的净基础收益, 后一项表征由排队拥塞引起的非线性时延惩罚 (即对 SLA 违约风险的经济量化)。同时, 定义局部优化问题 \mathcal{P}_k 的唯一最优解为 $\lambda_k^*(R_k)$, 即数据中心 A_k 针对价格 R_k 的最优响应策略。

3.4 调度中心成本模型

全局资源调度器作为 Stackelberg 博弈的领导者, 拥有系统的全局视图和定价权。全局资源调度器的核心职能是充当规划者角色, 旨在以最小的经济代价完成全网任务流的调度, 同时保障系统的服务可用性。

3.4.1 决策变量与预判机制

全局资源调度器作为博弈的领导者, 旨在保

障服务可用性的前提下, 以最小经济代价完成全网任务调度。调度器将价格向量 $R = \{R_1, R_2, \dots, R_N\}$ 作为决策变量, 通过预判各节点的最优响应策略 $\lambda_k^*(R_k)$, 建立以最小化总奖励金额为目标函数 $J(R)$:

$$\min_R \mathcal{J}(R) = \sum_{k=1}^N R_k \cdot \lambda_k^*(R_k) \#(6)$$

为保证调度的物理可行性与经济有效性, 该优化问题需满足两组关键约束:

(1) 服务可用性约束 (流量守恒)。所有节点承接的流率之和必须覆盖外部总任务负载 Λ_{total} , 即:

$$\sum_{k=1}^N \lambda_k^*(R_k) \geq \Lambda_{total} \#(7)$$

(2) 个体理性约束。奖励价格 R_k 必须高于节点启动的边际成本阈值, 方可激励其贡献算力, 即:

$$R_k > \Omega_k + \frac{\omega_d^k}{F_k}, \forall k \in \{1, \dots, N\} \#(8)$$

综上所述, 调度中心的全局成本优化模型可归纳为如下包含响应变量 $\lambda_k^*(R_k)$ 的非线性规划问题 \mathcal{P}_{leader} :

$$(\mathcal{P}_{leader}): \min_R \sum_{k=1}^N R_k \lambda_k^*(R_k) \#(9)$$

$$\text{s.t. C1: } \sum_{k=1}^N \lambda_k^*(R_k) \geq \Lambda_{total}$$

$$\text{C2: } R_k > \Omega_k + \frac{\omega_d^k}{F_k}, \forall k$$

3.5 Stackelberg 博弈形式化建模

基于上述系统与成本模型, 全局资源调度器与各数据中心间的交互可形式化为两阶段资源激励型 Stackelberg 博弈:

(1) 价格激励阶段 (领导者决策): 全局资源调度器作为领导者, 旨在最小化社会总成本 $J(R)$, 率先制定并下发最优资源奖励价格向量 $R = \{R_1, \dots, R_N\}$ 。

(2) 流量响应阶段 (跟随者决策): 各数据中心 A_k 作为跟随者, 在接收价格信号 R_k 后, 在物理约束 $0 \leq \lambda_k < F_k$ 内并求解局部效用最大化问题, 并向调度器反馈最优承诺流率 $\lambda_k^*(R_k)$ 。

该博弈的求解目标是寻求一个 Stackelberg 均衡点, 即任何一方单方面改变策略均无法获益的状态。

定义 2 (Stackelberg 均衡): 策略对 (R^*, λ^*) 构成该博弈的一个 Stackelberg 均衡, 当且仅当满足以下两个条件:

(1) 跟随者最优性: 对于任意给定的均衡价格 R_k^* , $\lambda_k^*(R_k)$ 是数据中心 A_k 的最优响应, 即满足:

$$U_k(\lambda_k^*(R_k^*), R_k^*) \geq U_k(\lambda_k(R_k^*), R_k^*), \forall \lambda_k \in [0, F_k), \forall k \#(10)$$

(2) 领导者最优性: 在所有数据中心均采用最优响应策略 $\lambda^*(R)$ 的前提下, R^* 最小化了领导者的目标函数, 即满足:

$$J(R^*, \lambda^*(R^*)) \leq J(R, \lambda^*(R)), \forall R \in \Omega_R \#(11)$$

其中 Ω_R 为满足式(9)中约束 C1 和 C2 的可行价格空间。

4 博弈均衡分析与算法设计

本文采用逆向归纳法求解该 Stackelberg 均衡: 首先推导跟随者的最优响应闭式解, 进而将其代入领导者模型转化为单层优化问题, 最后提出异步坐标下降 (ACD) 算法进行求解。

4.1 跟随者最优响应分析

数据中心 A_k 旨在物理约束 $0 \leq \lambda_k < F_k$ 下最大化局部效用 $U_k(\lambda_k)$ 。对式(5)求一阶与二阶偏导数:

$$\frac{\partial U_k}{\partial \lambda_k} = (R_k - \Omega_k) - \frac{\partial}{\partial \lambda_k} \left(\frac{\omega_d^k \lambda_k}{F_k - \lambda_k} \right) \#(12)$$

$$\frac{\partial^2 U_k}{\partial \lambda_k^2} = -\frac{2\omega_d^k F_k}{(F_k - \lambda_k)^3} \#(13)$$

由于 $F_k > \lambda_k$ 且参数均为正, 二阶导数恒小于 0, 表明效用函数为严格凹函数, 存在唯一全局最大值。

作为理性实体, 数据中心仅在边际收益大于 0 时参与调度。令式(12)中 $\lambda_k = 0$ 且 $\frac{\partial U_k}{\partial \lambda_k} > 0$, 即可导出个体理性约束 (即启动阈值):

$$R_k > \Omega_k + \frac{\omega_d^k}{F_k}$$

当价格 R_k 满足上述启动条件时, 令一阶导数等于 0 求极值:

$$R_k - \Omega_k = \frac{\omega_d^k F_k}{(F_k - \lambda_k)^2} \#(14)$$

结合物理约束 $\lambda_k < F_k$ 取正根, 可得数据中心针对价格 R_k 的最优响应闭式解:

$$\lambda_k^*(R_k) = F_k - \sqrt{\frac{\omega_d^k F_k}{R_k - \Omega_k}} \#(15)$$

该解析解中的开方项确保了承接流率始终不会触及物理极限 F_k , 赋予了系统自发规避拥塞崩溃的内生能力。

4.2 领导者全局优化求解

将数据中心最优响应闭式解 $\lambda_k^*(R_k)$ 代入 \mathcal{P}_{leader} , 全局优化问题转化为仅关于价格向量 R 的单层约束优化问题:

$$\min_R \mathcal{J}(R) = \sum_{k=1}^N R_k \cdot \left(F_k - \sqrt{\frac{\omega_d^k F_k}{R_k - \Omega_k}} \right) \#(16)$$

$$s.t. \sum_{k=1}^N \left(F_k - \sqrt{\frac{\omega_d^k F_k}{R_k - \Omega_k}} \right) \geq \Lambda_{total} \#(17)$$

为验证该问题的凸性, 考察目标函数 $J(R)$ 的海森矩阵 (为对角阵)。令常数 $C_k = \sqrt{\omega_d^k F_k}$, 求子目标函数 $J_k(R_k)$ 的二阶偏导数:



$$\frac{\partial^2 \mathcal{J}_k}{\partial R_k^2} = \frac{C_k}{(R_k - \Omega_k)^2} \left(\frac{1}{4} R_k - \Omega_k \right) \# (18)$$

由个体理性约束可知分母恒正，故当满足有效激励条件 $R_k > 4\Omega_k$ 时，二阶导数恒大于0。据此定义有效激励子域 \mathcal{D} :

$$\mathcal{D} \triangleq \left\{ R \mid R_k \geq \max(R_{th}^k, 4\Omega_k), \forall k \right\} \# (19)$$

在子域 \mathcal{D} 内， $J(R)$ 呈严格凸性，系统存在唯一的 Stackelberg 均衡点 $(R^*, \lambda^*(R^*))$ 。该数学性质具有明确的物理意义：(1) 在中高负载下，为激励节点承担指数级时延风险，最优价格必然包含较高的拥塞溢价并自然落入子域 \mathcal{D} 内，从而为系统最复杂的重载区域提供了严格的全局最优保证；(2) 在轻载下，解可能退化至启动阈值 R_{th}^k 附近，此时拥塞项极低，算法转而通过 KKT 条件收敛于稳定边界解。该性质不仅确立了算法处理核心痛点（拥塞）的理论完备性，也印证了实验中轻载下各算法性能趋同与反转的规律。

针对式(17)的耦合约束，引入非负拉格朗日乘子 $v \geq 0$ （即系统的影子价格），构建拉格朗日函数 $L(R, v)$:

$$L(R, v) = \sum_{k=1}^N \left((R_k - v) \lambda_k^* \right) + v \Lambda_{total} \# (20)$$

分别推导 $L(R, v)$ 关于 R_k 与 v 的梯度:

$$\nabla_{R_k} L = \lambda_k^* + \frac{1}{2} (R_k - v) \sqrt{\omega_d^k F_k} (R_k - \Omega_k)^{-3/2} \# (21)$$

$$\nabla_v L = \Lambda_{total} - \sum_{k=1}^N \lambda_k^*(R_k) \# (22)$$

式(22)直观反映了供需缺口：当总供给小于总需求时，梯度为正，驱动影子价格 v 上升，迫使调度器提高奖励价格以刺激算力供给。

4.3 异步坐标下降算法设计

为克服广域网长时延导致的“同步阻塞”瓶颈，本文设计了允许利用陈旧信息进行非阻塞更新的 ACD 算法。全局资源调度器维护全局状态缓存向量 $\lambda^{cache} = \{\lambda_1^c, \dots, \lambda_N^c\}$ ，基于事件驱动机制

执行异步迭代:

(1) 非阻塞监听。全局资源调度器收到 A_k 的最新反馈 $\lambda_k^*(R_k)$ 时，立即更新缓存分量 $\lambda_k^c \leftarrow \lambda_k^*(R_k)$ 。

(2) 局部坐标更新。对于刚刚更新了状态的数据中心 A_k ，全局资源调度器立即沿其梯度反方向更新对应的价格分量 R_k 。第 t 次迭代的更新规则为:

$$R_k^{(t+1)} = \left[R_k^{(t)} - \alpha \cdot \nabla_{R_k} \mathcal{L}(R^{(t)}, v^{(t)}) \right]_{R_{th}^k}^+ \# (23)$$

其中， $\alpha > 0$ 为价格更新步长， $[\cdot]_{R_{th}^k}^+$ 表示将价格投影到可行域 $(R_{th}^k, +\infty)$ 内，以满足个体理性约束。该规则的设计机理在于拉格朗日函数的可分离性结构：梯度 $\frac{\partial L}{\partial R_k}$ 仅取决于局部流率 λ_k^* 与全局影子价格 v ，而与其余节点状态 $\lambda_{j \neq k}$ 无关。这一数学特性使得调度器可彻底解耦节点决策，仅凭单点最新反馈即可独立更新，最大化了异构节点间的并行计算效率。

(3) 全局对偶更新。利用缓存的陈旧状态近似估算全局供需缺口，更新影子价格:

$$v^{(t+1)} = \left[v^{(t)} + \beta \cdot \left(\Lambda_{total} - \sum_{j=1}^N \lambda_j^c \right) \right]^+ \# (24)$$

其中， $\beta > 0$ 为对偶步长， $[\cdot]^+$ 表示 $\max(0, \cdot)$ ，确保影子价格非负。影子价格 v 表征了系统满足总负载的边际难度（市场供需紧张度）。由于总供给 $\sum \lambda_j$ 为全局耦合量，异步下无法实时获取所有节点最新值。为此，算法采用“基于陈旧信息的近似梯度估计”，利用包含滞后状态的缓存求和项 $\sum \lambda_j^c$ 替代真实供给。只要通信时延存在上限且步长足够小，这种基于陈旧梯度的迭代仍被理论保证收敛至全局最优解。

ACD 算法在数学上属于异步坐标下降范畴。经典分布式优化理论指出，该类算法的收敛性主

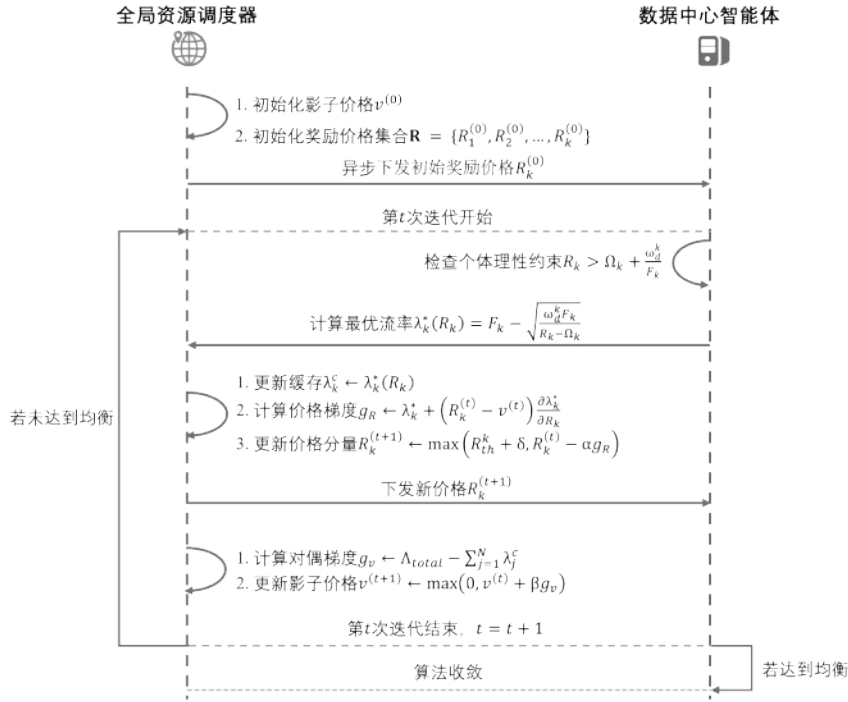


图3 基于RISG博弈的异步任务调度交互时序图

要依赖于目标函数的凸性与时延有界性：Bertsekas 与 Tsitsiklis^[25]证明了时延有界且步长适配时严格凸优化的异步全局收敛性；Tseng^[26]确认了坐标交替更新在凸性下的收敛保证；Liu 与 Wright^[27]进一步证明了利用陈旧梯度进行更新仍可保持线性收敛速率。

本文模型严格满足上述理论的先决条件：

- (1) 目标函数严格凸：如 4.2.1 节证明，在有效激励子域 \mathcal{D} 内，全局目标函数 $J(\mathbf{R})$ 的海森矩阵正定（严格凸）。结合 4.1.1 节跟随者最优响应的唯一性，确保了系统存在唯一的 Stackelberg 均衡点。
- (2) 通信时延有界：算法显式设定了最大允许滞后步数 τ_{max} （见 5.1 节），通过主动丢弃超时的陈旧反馈包，严格满足了异步理论中的时延有界假设。

综上所述，ACD 算法结合可行域投影更新与缓存状态处理，完全符合异步优化的收敛先决条件。因此，算法生成的迭代序列 $\{R^{(t)}\}$ 在理论上被严格保证收敛至唯一的 Stackelberg 均衡点，这

与第 5 节实验中观察到的稳健收敛现象高度一致。

5 实验分析

5.1 实验设置

本文基于 Python 3.10 与 SimPy 框架开发离散事件仿真平台，以精确模拟异步并发与资源竞争过程，数值与优化求解基于 NumPy 和 SciPy 库构建。

5.1.1 仿真环境与拓扑

(1) 网络拓扑构建。系统部署 $N=20$ 个异构数据中心，包含云节点 ($N_c=5$) 与边缘节点 ($N_e=15$)。云节点具备高算力与高稳定性，但广域网带宽成本较高；边缘节点传输成本极低，但计算资源受限且故障风险较高。各节点的物理参数均在给定范围内服从均匀分布（具体设置见表 2）。

(2) 广域网通信模型。为真实反映广域网的长尾时延特征，单向通信时延建模为对数正态分布：

$$\tau_{comm} \sim \text{LogNormal}(\mu, \sigma^2) \# (25)$$



算法1 用于RISG博弈均衡求解的异步坐标下降算法(ACD)

输入:	[主线程: GRB端]	[子线程: 数据中心 A_k 端]
总任务负载 Λ_{total} ;	1. WHILE $ R^{(t)} - R^{(t-1)} \geq \epsilon$ DO:	1. WHILE 系统中 DO:
数据中心物理参数 $\{F_k, \omega_e^k, \omega_d^k\}$;	2. 非阻塞监听网络端口;	2. IF 收到主节点下发的新价格 R_k THEN:
收敛精度 c .	3. IF 收到来自数据中心 A_k 的反馈包 λ_k^* THEN:	3. 根据个体理性约束检查启动条件;
输出: 最优价格向量 R^* ;	4. 更新缓存: $\lambda_k^c \leftarrow \lambda_k^*$;	4. 计算最优响应流率: $\lambda_k^* \leftarrow F_k - \sqrt{\left(\frac{F_k}{\omega_d^k}\right) / (R_k - \Omega_k)}$;
最优流率分配 λ^* ;	5. 计算价格梯度: $g_R \leftarrow \lambda_k^* + \left(R^{(t)} - v^{(t)}\right) \frac{\partial \lambda_k^*}{\partial R_k}$;	5. 向主节点发送反馈包 λ_k^* ;
初始化:	6. 更新价格分量: $R_k^{(t+1)} \leftarrow \max\left(R_{th}^k + \delta, R_k^{(t)} - \alpha g_R\right)$;	6. END IF
1. 设定初始价格 $R^{(0)} > R_{th}$, 影子价格 $v^{(0)} \geq 0$;	7. 下发新价格 $R_k^{(t+1)}$ 至数据中心 A_k ;	7. END WHILE
2. 初始化缓存 $\lambda^{cache} \leftarrow 0$;	8. 计算对偶梯度: $g_v \leftarrow \Lambda_{total} - \sum_{j=1}^N \lambda_j^c$;	
3. 设定迭代计数 $t \leftarrow 0$;	9. 更新影子价格: $v^{(t+1)} \leftarrow \max\left(0, v^{(t)} + \beta g_v\right)$;	
	10. $t \leftarrow t + 1$;	
	11. END IF	
	12. END WHILE	

此外, 为了测试异步算法的极限性能, 本文设定最大允许滞后步数 $\tau_{max} = 5$, 即允许全局调度器利用滞后不超过5个迭代周期的陈旧反馈信息进行策略更新, 以还原网络拥塞与数据包乱序现象。

5.1.2 任务负载生成

为模拟任务的随机到达与异构处理需求, 本文基于泊松过程构建动态任务生成器:

(1) 任务到达过程: 总任务负载 Λ_{total} 被建模为一个连续时间随机过程, 其到达的时间间隔服从参数为 Λ_{total} 的指数分布:

$$P(Interval \leq t) = 1 - e^{-\Lambda_{total}t} \#(26)$$

(2) 服务需求: 本文采用统计抽象的方法, 假设每个的任务大小服从指数分布。具体而言, 对于处理能力为 F_k 的数据中心 A_k , 其处理单个任务

表2 仿真环境与异构节点参数设置

参数类别	参数	取值或分布范围
全局环境	N	20
	Λ_{total}	[3000, 9500]
	τ_{comm}	LogNormal(0.1s, 0.5s)
云节点	F_k	Unif[8000, 12000]
	ω_e^k	Unif[0.1, 0.3]
	ω_{bw}^k	Unif[0.05, 0.10]
	θ_k	Unif[0.001, 0.01]
边缘节点	F_k	Unif[1000, 3000]
	ω_e^k	Unif[0.5, 1.0]
	ω_{bw}^k	Unif[0.001, 0.01]
	θ_k	Unif[0.1, 0.5]
算法参数	ω_d^k	Unif[1.0, 5.0]
	α, β	1.0×10^{-5}

所需的服务时间 S_k 是一个满足 $S_k \sim Exp(\mu_k)$ 的随机变量。其中 μ_k 为平均服务率, 参数 F_k 本质上表征了数据中心的物理计算容量与全网平均任务大小的比值, 即 $F_k = E[1/S_k]$ 。

(3) 负载强度定义

为量化系统的压力水平, 定义全网平均负载率 ρ_{sys} 为总需求与总供给之比:

$$\rho_{sys} = \frac{\Lambda_{total}}{\sum_{k=1}^N F_k} \#(27)$$

实验覆盖了 $\rho_{sys} \in [0.20, 0.90]$ 区间, 以全面评估算法从轻载到极限拥塞状态下的鲁棒性。

5.1.3 对比基准算法

为全面评估所提算法的综合性能, 本文选取5类代表性策略作为对比基准。其中, 不感知实时拥塞的静态加权轮询调度 (weighted round-robin, WRR) 作为量化动态收益的基础性能下界; 重载下极易引发流量振荡的贪婪最小成本调度 (greedy minimum cost, GREEDY) 用于验证本文模型的拥塞规避能力。在机制对比方面, 采用相同博弈模型但强制同步的同步博弈梯度算法 (Sync-Game) 旨在量化异步机制克服长时延瓶颈的核心价值; 采用常规投影策略的异步梯度投影

算法 (asynchronous gradient projection, Async-GP) 则排除了异步共有增益, 重点凸显本文“坐标下降”机制在处理陈旧信息与抑制振荡上的独特性。最后, 引入高并发维护开销大的经典分布式优化算法——异步交替方向乘子法 (asynchronous alternating direction method of multipliers, Async-ADMM), 以此验证 ACD 算法在轻量级与高性能间取得的更优平衡。

5.2 异步算法收敛性分析

为验证 ACD 算法的时间效率优势, 本节在中载场景及对数正态分布的随机网络时延下, 对比了四种算法的收敛轨迹 (如图 4 与表 3 所示)。实验结果呈现出以下特征:

(1) Sync-Game: 受限于“掉队者效应”, 收敛曲线呈阶梯状缓慢下降且伴随剧烈振荡, 收敛耗时最长 (91 秒), 难以在长尾时延网络中达到理想均衡。

(2) Async-GP: 虽消除阻塞提升了速度 (26 秒), 但全量参数更新将不同时效的陈旧信息混合, 引入了较大的噪声干扰, 致使其收敛速度略逊于本文算法。

(3) Async-ADMM: 表现出严重的不稳定性与“过冲”现象 (如 110 秒附近的尖峰)。尽管其最终社会总成本 (14,497) 在数值上最低, 但这实际上是由于高并发异步环境下缺乏严格校正, 算法未能维持全局流量守恒约束 (即实际供给 $\sum \lambda_k < \Lambda_{total}$), 导致其收敛至物理上不可行的失效状态。

(4) ACD: 展现出最优的收敛性能。得益于坐标下降机制每次仅针对单一节点进行局部更新, 有效隔离了陈旧信息的全局传播, 其曲线平滑且快速下降。ACD 仅需 19 秒即可稳定收敛至社会总成本最低的可行解 (20,026), 耗时较 Sync-Game 与 Async-GP 分别减少了 79.1% 与 26.9%, 充分验证了其在高抖动环境下的时间效率与极强的鲁棒性。

表 3 不同算法的收敛时间和最终社会总成本

算法	收敛时间 (秒)	最终社会总成本
Sync-Game	91	19,053
Async-GP	26	17,633
Async-ADMM	59	14,497
ACD	19	20,026

5.3 系统性能对比分析

5.3.1 不同负载下的社会总成本

社会总成本综合量化了系统能耗、带宽及拥塞时延。图 5(a) 与表 4 展示了系统平均负载率 $\rho_{sys} \in [0.2, 0.9]$ 时各策略的性能演化规律, 实验结果呈现出显著的阶段性特征:

(1) 轻载阶段 ($\rho_{sys} < 0.3$) 的性能反转: 此时全网排队时延趋于零, 总成本由静态的能耗与传输开销主导。无需通信与迭代的 GREEDY 算法凭借直接命中最低成本节点而效率最高; 相比之下, ACD 算法因维持全局均衡博弈产生了轻微的额外迭代开销, 导致成本略高。

(2) 中高载阶段 ($\rho_{sys} > 0.3$) 的拥塞规避优势: 随着负载攀升, 非线性排队效应激增。WRR 算法因缺乏拥塞感知引发任务积压, GREEDY 算法则因贪婪切流陷入全局振荡, 两者性能均迅速衰退。反观 ACD 算法, 在 $\rho_{sys} > 0.4$ 后展现出极其优异的全局鲁棒性。特别是在 $\rho_{sys} = 0.9$ 的极限拥塞下, ACD 成功将总成本控制在 36,939, 较 WRR 降低了 14.82%, 有效抑制了系统级的成本爆炸。

5.3.2 任务响应时延

图 5(b) 与表 5 展示了随着负载攀升, 各算法在任务响应时延上的演化趋势:

(1) 低负载阶段 ($\rho \leq 0.5$) 的极低时延: 此时系统处于物理传输主导的线性区间, 排队效应尚未显现。所有算法均能将平均响应时间维持在 5×10^{-4} 秒以内的极低水平, 呈现出相近的基准性能。

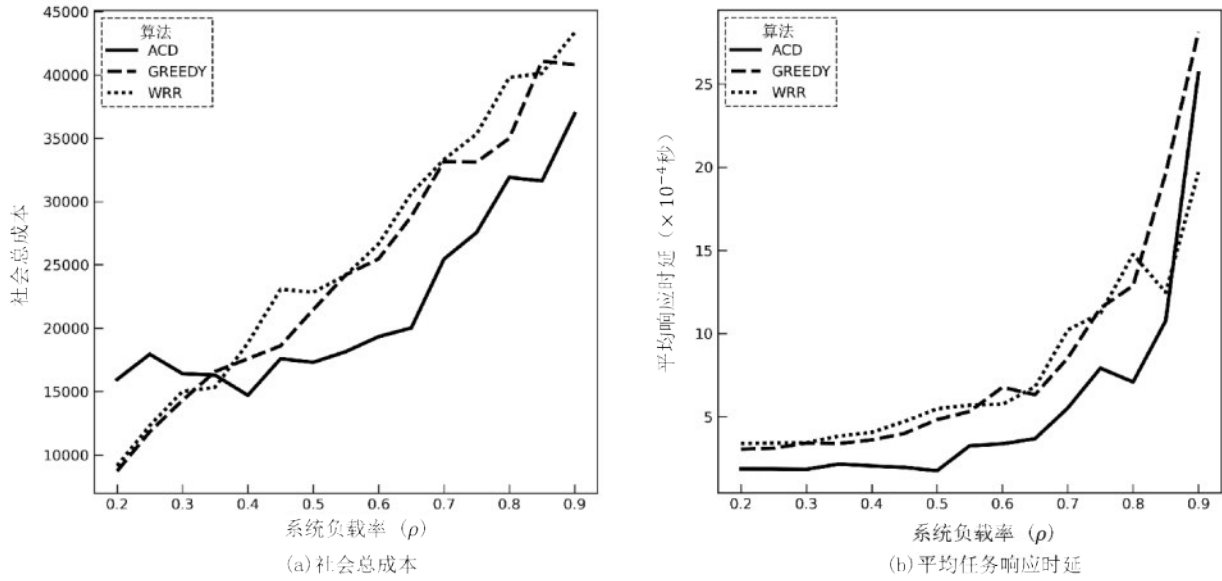


图5 不同调度方法在不同负载下的性能评估

表4 不同负载场景下的社会总成本量化对比

负载场景	系统负载率 (ρ_{sys})	ACD成本	GREEDY 成本	WRR成本	ACD相对GREEDY 优化率	ACD相对WRR 优化率
轻载	0.25	17,941	11,821	12,301	-51.77%	-45.85%
中载	0.60	19,330	25,472	26,656	24.11%	27.48%
重载	0.85	31,626	41,084	40,128	23.02%	21.19%
极限	0.90	36,939	40,813	43,362	9.49%	14.82%

(2) 中高负载阶段 ($\rho > 0.5$) 的启发式崩溃：随着负载攀升，启发式算法的局限性暴露无遗。GREEDY 因短视地将流量导向当前“最快”节点，导致这些节点瞬间由空闲转为极度拥塞，其时延在 $\rho=0.9$ 时呈指数级陡升至最高的 28.12×10^{-4} 秒。WRR 则因无法实时感知节点波动，在 $\rho=0.8$ 附近即发生局部提前饱和，触发了高达 14.78×10^{-4} 秒的剧烈尖峰。

(3) 极限压测下 ACD 的表现：相比之下，ACD 在 $\rho=0.2$ 到 0.85 区间内始终保持着极低且平稳的时延（在 $\rho=0.8$ 时仅为同期的约 50%）。特别是在 $\rho=0.9$ 的极限压测下，ACD 算法成功避免了非线性拥塞带来的时延指数级爆炸，相比贪婪策略降低了约 30% 的时延。这一结果表明，在面对流量突发（Traffic Burstiness）引发的瞬时过载

挑战时，所提方法能够利用价格杠杆快速抑制拥塞蔓延，保障系统在峰值流量冲击下的服务韧性。其机理在于成本函数中的拥塞惩罚项赋予的前瞻性：当某节点排队时间即将增加时，其资源价格自动上涨，迫使调度器提前将流量溢出至其他节点，从而在系统逼近物理极限前提供了可靠的 QoS 保障。

5.4 鲁棒性与适应性分析

5.4.1 节点可靠性与风险规避机制

为验证模型在边缘设施不稳定时的风险规避能力，实验将边缘节点的风险惩罚系数 θ_k 从默认的低水平 $[0.1, 0.5]$ 提升至 $[1.0, 2.0]$ ，以模拟高风险波动。图 6 展示了系统在极高负载下负载分布的策略性演化：

(1) 基础场景的充分利用：在低风险下，系

表5 不同负载场景下的平均任务响应时延对比

负载场景	系统负载率 (P_{sys})	ACD时延 ($\times 10^{-4}s$)	GREEDY 时延($\times 10^{-4}s$)	WRR时延 ($\times 10^{-4}s$)	ACD相对GREEDY 优化率	ACD相对WRR 优化率
轻载	0.25	1.859	3.107	3.412	40.17%	45.52%
中载	0.60	3.368	6.762	5.737	50.19%	41.29%
重载	0.85	10.766	19.637	12.470	45.17%	13.66%
极限	0.90	25.652	28.120	19.706	8.78%	-30.17%

统追求传输成本最优，边缘与云节点均保持极高的资源利用率（负载中位数分别为 0.935 与 0.962）。

(2) 高风险场景的精细化卸载：当边缘风险激增时，云节点箱线图整体显著上移（中位数攀升至 0.971），主动承接增量流量；同时，边缘节点负载呈现明显的离散化下延（下须延伸至 0.91 附近）。这表明算法实现了针对实时状态的精细化流量卸载，而非“一刀切”地弃用边缘节点。

(3) 内生动态适应性（核心机理）：这一现象深刻体现了算法对节点异构性的动态适应能力。面对云端（高算力/高稳定）与边缘（低时延/低稳定）的物理属性差异，RISG 并未采用静态切分策略，而是依据异构节点的实时边际成本差异，自动将对可靠性敏感的流量从高风险边缘侧向核心云侧迁移。其底层逻辑在于：当 θ_k 增大推高边际运营成本 Ω_k 时，节点的最优响应流率 λ_k^* 会通过成本函数的数学传导被自然抑制。这种无需硬编码规则的内生规避机制，使得系统能以牺牲少量传输成本为代价，换取全局可靠性的最

大化。

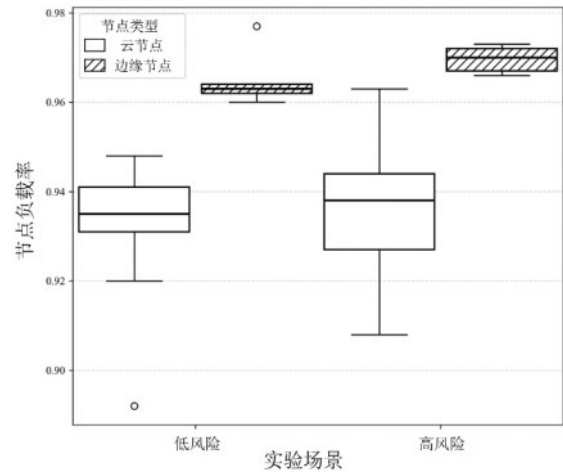
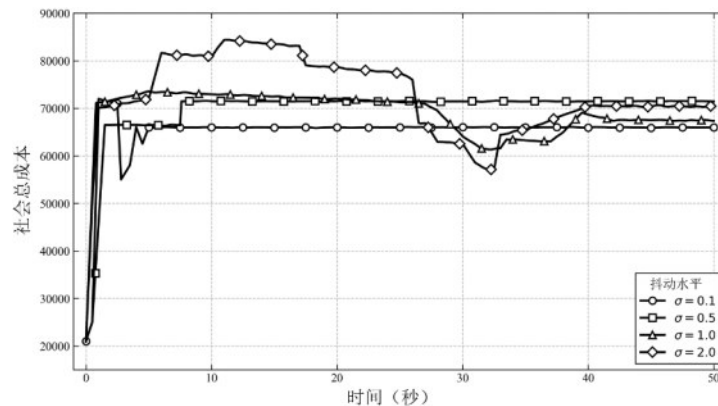


图7 不同网络抖动强度下的算法收敛轨迹对比

图6 不同可靠性风险下的负载率分布对比

5.4.2 抗网络抖动性能

为评估算法在恶劣通信链路下的鲁棒性，实验设定了四组不同的网络时延标准差 $\sigma \in \{0.1, 0.5, 1.0, 2.0\}$ 。图7展示了不同抖动强度下算法收敛轨迹的演化过程，展示了网络不确定





性对异步博弈的影响。

(1) 低抖动环境下的极速收敛 ($\sigma=0.1$): 通信近似理想状态, 算法呈现极高的更新频率。经过初期短暂的参数微调与试错后, 仅需约7秒即可迅速越过振荡期, 达到社会总成本约65000的稳定均衡点。

(2) 高抖动环境下的滞后突变与回调 ($\sigma \geq 1.0$): 随着长尾时延加剧, 收敛轨迹出现显著的“滞后突变”。在初期 (0-25s), 调度器因缺乏新鲜反馈陷入“信息真空期”而短暂表现为高成本台阶; 进入30s-40s区间, 被网络阻塞的状态更新包集中到达触发大幅修正。尽管陈旧信息在32s附近引发了短暂的超调 (overshoot), 但算法凭借后向反馈迅速回调并恢复稳定。

在 $\sigma=2.0$ 的极端高抖动环境下, 算法仍能收敛至误差8%以内的均衡解。这充分验证了ACD算法对网络时延不确定性的鲁棒性。通过允许利用陈旧信息进行非阻塞更新, 算法有效克服了广域网长尾时延带来的同步瓶颈, 实现了在强随机网络环境下的稳定协同。

5.5 算法扩展性分析

为验证在大规模分布式环境下的适用性, 实验在保持中等负载与高抖动 ($\rho=0.5, \sigma=1.0$) 的前提下, 将数据中心节点数量 N 从10逐步扩展至100。时间结果如图8与表6所示。

(1) 同步机制: 受“掉队者效应”掣肘, 随着节点规模扩张, 长尾高时延节点出现的概率急剧上升。其收敛时间呈现超线性的陡峭增长, 从 $N=10$ 时的87秒激增至 $N=100$ 时的259秒, 严重限制了其在大规模系统中的扩展能力。

(2) 异步基准算法: 两类异步算法虽避免了同步阻塞, 但暴露出不同程度的不稳定性。Async-ADMM在百节点规模下因缺乏有效的对偶变量约束, 出现剧烈的非单调震荡 (耗时149秒); Async-GP在小规模下表现极佳 ($N=10$ 时仅需27秒), 但随着 N 增加, 陈旧梯度引发的累积误差急剧放大, 迫使算法消耗更多轮次修正方向, 最终耗时增至162秒。

(3) ACD算法: 得益于局部坐标更新的目标解耦机制, ACD有效隔离了节点规模扩张带来的计算耦合与通信干扰。在 $N=100$ 的大规模场景下, ACD的收敛时间稳控在123秒, 相比Sync-Game大幅降低了52.5%, 相比同类异步算法Async-ADMM和Async-GP也分别提升了约17.4%和24.1%。这一结果有力证明了所提方法在应对大规模跨区域算力调度时, 具备极其卓越的扩展性与资源配置效能。

6 结论

本文针对跨区域MDCS面临的异构性与广域网长尾时延挑战, 提出了一种基于资源激励Stackelberg博弈的异步调度方法。该方法通过构建非线性拥塞成本模型并设计异步坐标下降 (ACD) 算法, 有效克服了传统同步架构的“掉队者效应”。理论与实验证明, ACD算法严格保证收敛至唯一均衡解; 相比异步梯度投影算法, 其收敛速度提升约29.17%; 在系统重载 ($\rho=0.85$) 工况下, 响应时延较贪婪策略降低约45.17%, 且在强网络抖动环境中展现出卓越的鲁棒性, 成功实现了社会总成本与QoS的双重

表6 不同节点数量下的算法收敛时间对比

算法	N=10	N=20	N=30	N=40	N=50	N=60	N=70	N=80	N=90	N=100
Sync-Game	87s	91s	126s	156s	167s	201s	212s	203s	233s	259s
Async-ADMM	42s	59s	48s	63s	99s	110s	92s	117s	123s	149s
Async-GP	27s	26s	52s	56s	69s	63s	69s	75s	114s	162s
ACD	16s	19s	32s	39s	46s	53s	72s	61s	77s	123s

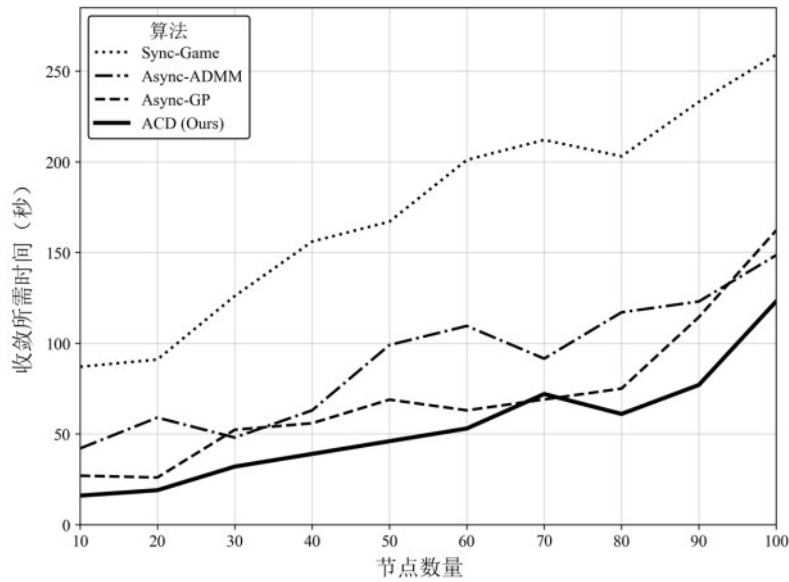


图8 不同节点数量下的算法收敛时间对比

优化。

尽管所提方法展现出显著的效率优势，未来研究仍拟在以下两方面进行拓展：一方面，拟引入更通用的G/G/1排队模型或数据驱动方法，以突破现有M/M/1模型的局限，增强对现实复杂流量的拟合精度；另一方面，后续将探索分层或多领导者（multi-leader）博弈架构，以克服集中式调度的单点瓶颈，进一步提升超大规模网络下系统的可扩展性与容错能力。

参考文献：

[1] 张贵鹏, 孙毓忠. 大模型训练的混合并行技术综述[J]. 计算机学报, 2026, 49(1): 109-131.
Zhang G, Sun Y. A survey of hybrid parallelism for large model training[J]. Chinese Journal of Computers, 2026, 49(1): 109-131.

[2] 李鸣, 宋文鹏, 宗燕, 等. 基于区块链的元宇宙生态体系架构[J]. 计算机研究与发展, 2024, 61(9): 2364-2383.
Li M, Song W, Zong Y, et al. Blockchain based metaverse ecosystem architecture[J]. Journal of Computer Research and Development, 2024, 61(9): 2364-2383.

[3] 冯运, 童翊轩, 王耀南, 等. 重大装备集群机器人协同制造数字孪生技术综述[J]. 自动化学报, 2025, 51(7): 1463-1479.
Feng Y, Tong Y, Wang Y, et al. Digital twin technology for collaborative manufacturing of major equipment by cluster robots:

A review[J]. Acta Automatica Sinica, 2025, 51(7): 1463-1479.

[4] 陈星延, 张雪松, 谢志龙, 等. 面向“云—边—端”算力系统的计算和传输联合优化方法[J]. 计算机研究与发展, 2023, 60(4): 719-734.
Chen X, Zhang X, Xie Z, et al. A computing and transmission integrated optimization method for cloud-edge-end computing first system[J]. Journal of Computer Research and Development, 2023, 60(4): 719-734.

[5] Wu Y, Tang S, Ce Y, et al. Task scheduling in geo-distributed computing: A survey[J]. IEEE Transactions on Parallel and Distributed Systems, 2025, 36(10): 2073-2088.

[6] Dean J, Barroso L A. The tail at scale[J]. Communications of the ACM, 2013, 56(2): 74-80.

[7] Yang Y, Shen H, Tian H. Scheduling workflow tasks with unknown task execution time by combining machine-learning and greedy-optimization[J]. IEEE Transactions on Services Computing, 2024, 17(3): 1181-1195.

[8] Cheng S, Ren T, Zhang H, et al. A Stackelberg-game-based framework for edge pricing and resource allocation in mobile edge computing[J]. IEEE Internet of Things Journal, 2024, 11(11): 20514-20530.

[9] Hady M A, Hu S, Pratama M, et al. Multi-agent reinforcement learning for resources allocation optimization: A survey[J]. Artificial Intelligence Review, 2025, 58(11): 354.

[10] Ning Z, Xie L. A survey on multi-agent reinforcement learning and its application[J]. Journal of Automation and Intelligence, 2024, 3(2): 73-91.

[11] Albrecht S V, Christianos F, Schäfer L. Multi-agent reinforce-



- ment learning: Foundations and modern approaches[M]. Cambridge: MIT Press, 2024.
- [12] Srinivasan A, Singh U, Tirkkonen O. Multi-agent reinforcement learning approach scheduling for In-X subnetworks[C]// Proceedings of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall). Piscataway, NJ: IEEE, 2024: 1-7.
- [13] Wu J, Tang M, Jiang C, et al. Cloud-Edge - End collaborative task offloading in vehicular edge networks: A multilayer deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2024, 11(22): 36272-36290.
- [14] Tian S, Xu K, Xiang S, et al. Task offloading and resource scheduling in mobile edge-cloud computing based on edge competition and task prediction[J]. IEEE Transactions on Services Computing, 2025, 18(5): 3262-3275.
- [15] Long S, Wang C, Long W, et al. An efficient task scheduling algorithm in the cloud and edge collaborative environment[J]. Chinese Journal of Electronics, 2024, 33(5): 1296-1307.
- [16] Song Y, Ma Z, Chen N, et al. Comparative analysis of centralized and distributed multi-UAV task allocation algorithms: A unified evaluation framework[J]. Drones, 2025, 9(8): 530.
- [17] 管吉喆, 程光, 周余阳. 基于交换机迁移的控制平面饱和和攻击防御方法[J]. 计算机学报, 2024, 47(12): 2889-2908.
Guan J, Cheng G, Zhou Y. Control plane saturation attack defense based on switch migration[J]. Chinese Journal of Computers, 2024, 47(12): 2889-2908.
- [18] Habibpour Roudsari M N. Improved task scheduling in heterogeneous distributed systems using intelligent greedy Harris hawk optimization algorithm[J]. Evolutionary Intelligence, 2024, 17(5): 4199-4226.
- [19] Mahdizadeh M, Montazerolghaem A, Jamshidi K. Task scheduling and load balancing in SDN-based cloud computing: A review of relevant research[J]. Journal of Engineering Research, 2025, 13(4): 3132-3146.
- [20] Salimi R, Azizi S, Abawajy J. A greedy randomized adaptive search procedure for scheduling IoT tasks in virtualized fog - cloud computing[J]. Transactions on Emerging Telecommunications Technologies, 2024, 35(5): e4980.
- [21] Priya S S, Rajendran T. Load balancing using improved weighted round robin algorithm in cloud computing environment[J]. International Journal of Cloud Computing, 2024, 13(5): 463-484.
- [22] Sree Priya S, Rajendran D T. Enhanced weighted round robin: A new paradigm in cloud load balancing[J]. Indian Journal of Science and Technology, 2025, 18(15): 1220-1228.
- [23] Ali E M, Abawajy J, Lemma F, et al. Analysis of deep reinforcement learning algorithms for task offloading and resource allocation in fog computing environments[J]. Sensors, 2025, 25(17): 5286.
- [24] Zhu B, Wang D. Master-slave game optimal scheduling for multi-agent integrated energy system based on uncertainty and demand response[J]. Sustainability, 2024, 16(8): 3182.
- [25] Bertsekas D P, Tsitsiklis J N. Parallel and Distributed Computation: Numerical Methods[M]. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [26] Tseng P. Convergence of a block coordinate descent method for nondifferentiable minimization[J]. Journal of Optimization Theory and Applications, 2001, 109(3): 475-494.
- [27] Liu J, Wright S J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties[J]. SIAM Journal on Optimization, 2015, 25(1): 351-376.

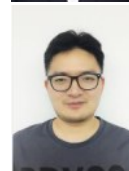
[作者简介]



王睿昊 (2003-), 男, 浙江大学软件学院硕士研究生, 主要研究方向为智能物联网、数据评估、大模型训练加速等。



陈益杉 (1995-), 女, 博士, 江西理工大学信息工程学院副教授、硕士生导师, 主要研究方向为云计算、边缘智能、物联网等。



赵周行 (1989-), 男, 硕士, 浙江通信行业服务有限公司研发总监, 主要研究方向为人工智能架构、多模态模型、嵌入式系统等。



赵新奎 (1990-), 男, 博士, 浙江大学软件学院百人计划研究员、博士生导师, 浙江省数智服务技术重点实验室副主任、浙江大学先进计算与新兴服务研究所副所长, 主要研究方向为智能操作系统、云计算与分布式系统、多智能体协同系统等。



丁智暘 (2004-), 男, 西北工业大学教育实验学院本科生, 主要研究方向为云计算、智能物联网等。

(1996-), 男, 博士, 浙江大学软件学院特聘研究员、硕士生导师, 主要研究方向为多智能体协同、智能物联网、数据评估、数据安全等。