研究与开发

智算网络中面向QoS感知的资源调度技术研究

赵明
（中信数字科技集团有限公司，北京 100004）

摘　要：为保障大语言模型推理与多智能体协同等新兴智能业务对服务质量（quality of service，QoS）的需求，构建高效的资源调度机制已成为智算网络中的核心问题。面向QoS多目标优化下的资源调度场景，设计了一种融合差异化智能业务建模、异构算网资源感知与高通量远程直接内存访问（remote direct memory access，RDMA）传输优化的调度框架；在该框架下提出基于强化学习（reinforcement learning，RL）的智能调度算法，在考虑业务优先级、算网资源负载与数据传输特性的基础上，自适应生成调度策略以提升QoS；引入网络实测数据与智算中心真实组网参数构建仿真平台，对所提算法进行性能验证。实验结果表明，该方法在任务完成率、平均时延和资源利用率等关键QoS指标上较现有方案具有明显优势。

关键词：智算网络；服务质量保障；资源调度；强化学习；远程直接内存访问网络
中图分类号：TN92
文献标志码：A
doi: 10.11959/j.issn.1000−0801.DXKX260062

Research on QoS-aware resource scheduling for AI computing networks

Zhao Ming
CITIC Digital Technology Group Co., Ltd., Beijing 100004, China

Abstract: To meet the quality of service (QoS) requirements of emerging intelligent services such as large language model inference and multi-agent collaboration, constructing an efficient resource scheduling mechanism has become a core issue in AI computing networks. For the resource scheduling scenario under QoS multi-objective optimization, a scheduling framework that integrates differentiated intelligent service modeling, heterogeneous computing-network resource awareness, and high-throughput remote direct memory access (RDMA) transmission optimization was proposed. Within this framework, a reinforcement learning (RL)-based intelligent scheduling algorithm was developed to adaptively generate scheduling policies by considering service priorities, computing-network resource loads, and data transmission characteristics, thereby improving QoS. Finally, a simulation platform was constructed using real-world network measurement data and actual topology parameters from an AI computing center to verify the performance of the proposed algorithm. Experimental results demonstrate that the proposed method achieves significant improve-

收稿日期：2026−01−23；**修回日期：**2026−04−10
通信作者：赵明，zhaoming@citic.com

ments over existing schemes in key QoS metrics such as task completion rate, average latency, and resource utilization.

Key words: AI computing network, QoS assurance, resource scheduling, reinforcement learning, RDMA network

0 引言

随着人工智能（artificial intelligence，AI）技术的迅猛发展，以大语言模型和智能体为代表的新型AI应用正在爆发式涌现[1]。此类任务不仅在模型应用本身具有参数规模大、计算密集度高、语义逻辑依赖性强等特点，还强调推理过程中的状态感知、自主决策与多轮交互能力，对底层算力与通信基础设施提出了前所未有的性能挑战。据相关测算，一次完整的大模型推理过程可能需要数十至上百次跨节点通信，延迟敏感性显著提升[2]。此外，在多智能体协同环境中，任务不仅需要满足个体响应时效，还需要支持智能体间的连续意图传递与多阶段任务链协同。因此，为满足AI时代复杂、动态的计算与通信需求，亟须构建具备异构资源聚合、高速互联、服务质量（quality of service，QoS）保障能力的智算网络。该网络通常由云端大规模图形处理器（graphics processing unit，GPU）集群、边缘智算节点与终端设备组成，结合远程直接内存访问（remote direct memory access，RDMA）[3]等高通量通信协议，可实现大规模模型任务的分布式执行与低时延响应，是新一代AI基础设施的核心支撑平台。

在智算网络中，调度技术作为连接服务需求与底层资源的桥梁，是保障多任务、高实时性计算任务顺利执行的关键环节[4]。与传统批处理或流媒体类服务不同，智算任务的调度需要面向服务语义进行识别与建模，充分考虑任务间依赖性、阶段性结构和时延响应约束，进而制定符合服务等级协议（service level agreement，SLA）的调度策略。基于此，实现智算网络中的任务调度面临3个方面的挑战：（1）算网资源能力高度异构，中央处理器（central processing unit，CPU）、GPU、神经网络处理器（neural processing unit，NPU）等计算资源在算力密度、能耗模型、负载动态性方面差异显著，且网络带宽、链路时延因区域与设备类型不同而波动较大；（2）任务需求高度多样化，大模型推理任务可能关注吞吐率与批量处理效率，而智能体交互任务则侧重于交互延迟、语义连续性与多智能体一致性；（3）调度算法复杂度高，尤其是在多目标优化与动态环境下构建实时、可扩展的调度策略，须兼顾全局资源效率与个体QoS保障，调度引擎的决策能力成为系统瓶颈。

针对上述挑战，已有大量研究围绕任务调度与服务保障展开探索。部分方法基于任务图[5]或队列模型[6]，采用最早截止时间优先、最小完成时间等启发式算法实现资源分配[7]；也有研究引入强化学习（reinforcement learning，RL）算法[8]或深度强化学习（deep reinforcement learning，DRL）[9]算法实现策略优化与调度自适应。在网络通信层面，RDMA作为一种低时延、高带宽的数据传输机制，被广泛用于AI模型训练、键值（key-value，KV）缓存同步与参数分发，能够显著提升任务间通信效率[10]。然而，当前大多数研究仍存在以下问题：（1）调度模型未显式考虑大模型生成或多智能体交互任务等智算业务对时延与语义一致性等QoS指标的敏感性；（2）调度策略未与底层网络状态联动，无法感知RDMA链路负载、路径拥塞等状态变化；（3）调度系统缺乏动态反馈与自我优化机制，导致QoS保障的实时性与精度不足。

为此，本文针对智算网络中面向QoS感知的资源调度问题，提出一种融合服务建模、资源感知与传输优化的调度系统架构，重点解决在高动

态性、异构性与时延敏感性环境中服务质量可保障的问题。本文将大模型推理任务与多智能体服务流程统一抽象为多阶段、异构资源依赖的调度需求，设计可感知算力节点状态、RDMA 链路带宽与网络反馈的联合调度机制，并基于 RL 构建动态策略生成器，实现对复杂任务场景下资源分配的持续优化。该方法既能适配大模型高吞吐执行场景，也可满足交互类服务对响应延迟与语义连续性的保障需求，具有良好的适应性与可扩展性。本文的主要贡献总结如下。

（1）构建一种适用于大模型与智能体业务的多 QoS 目标资源调度系统模型，支持任务链结构抽象、服务等级分类与资源状态建模。

（2）提出一种融合算力节点负载与 RDMA 链路状态感知的 RL 调度算法，实现任务完成率、响应时延与资源利用效率的联合优化。

（3）设计基于真实 RDMA 测试数据与智算网络架构的仿真平台，对所提方法进行对比验证，证明其在多个 QoS 指标上的优势。

1　相关研究

近年来，随着云计算、边缘计算与 AI 技术的融合发展，任务调度与资源管理问题在新一代计算网络体系中引发了广泛关注。在智算网络背景下，任务规模呈指数级增长，服务需求日益复杂，资源结构更加异构，如何高效调度任务、保障服务质量并实现计算与通信资源的融合优化，成为当前研究的热点与难点。现有研究从不同视角出发，围绕任务调度策略、QoS 感知机制、算力网络架构与协同机制等方面进行了系统探索，提出了多种理论模型与算法方案。本文将从以下 3 个方面回顾相关研究：任务调度的策略演进与挑战、QoS 保障机制的发展趋势、算力网络中资源融合与卸载协同的研究进展。

1.1　任务调度

任务调度作为计算与网络系统中最基础且关键的研究问题之一，旨在通过任务的合理分配与资源的高效利用，提升系统整体的响应速度、吞吐率与能效水平。

文献[11]提出了一种基于任务紧急度的列表调度算法。文献[12]在边缘云协同架构下提出了一种动态调度算法，实现了边缘节点间及云边间的双层调度。文献[13]面向边缘工业互联网中的低时延通信需求，提出了基于非碰撞理论的确定性调度与动态队列机制。此外，文献[14]提出 DRL 驱动的端到端确定性调度机制，在异构网络环境下实现了资源抽象与全局协同。

1.2　QoS 保障

QoS 保障机制旨在确保不同服务请求在时延、带宽、分组丢失率等方面满足应用预期，是网络系统中保障用户体验的核心目标之一。

文献[15]提出一种信任策略，联合考虑请求动态调整与弹性资源配置。文献[16]构建了跨域 QoS 保障的两层软件定义网络框架，通过服务类别优先权重机制与控制器全局统计，满足端到端 QoS 需求。文献[17]基于有效容量理论，提出本地信息驱动多址接入机制。此外，文献[18]采用随机网络演算方法，实现了高可靠场景下的性能边界分析与分配优化。

1.3　算力网络

算力网络旨在融合计算、存储与传输资源，具备弹性调度能力，是支撑多智能体协作与大模型服务的关键底座。

文献[19]基于迁移 RL 联合考虑计算与通信开销，实现服务迁移最优化。文献[20]通过构建面向算力网络的三维资源调度模型，降低时延与能耗。文献[21]在卫星算力网络中通过 Lyapunov 优化实现边缘节点间协同负载平衡。文献[22]引入数字孪生算力网络，提出区块链增强的安全卸载与缓存调度策略。此外，文献[23]利用图学习建模多用户多任务卸载行为，实现计算效率与泛化能力的同步提升。

尽管已有研究围绕任务调度、QoS感知与资源融合开展了大量工作，但目前智算网络仍面临突出挑战。一方面，传统调度方法多针对静态工作流场景，采用启发式或规则驱动算法，难以适应AI任务的高动态性与并发特性，缺乏面向异构资源与任务特征的自适应机制；另一方面，现有QoS保障机制未能充分考虑智算业务类型之间的差异化需求，如大模型训练注重带宽与吞吐，推理更关注时延与抖动，智能体交互则强调连续性与响应一致性，亟须实现多目标、多粒度的QoS协同优化；此外，在传统算力网络架构基础上，构建具备智能感知、决策与调控能力的智算网络仍处于起步阶段，特别是在高通量RDMA等新型网络机制支持下，计算与通信的联合调度模型尚不成熟，阻碍了任务端到端性能的提升。

为应对上述挑战，本文围绕智算网络中面向QoS需求保障的资源调度问题，设计了一种融合异构资源感知、差异化服务建模与RDMA传输优化的智能调度框架，提出基于RL的策略自适应生成算法。该算法综合考虑任务优先级、系统负载、网络特性与服务目标，实现了多源任务在智算网络中的高效调度与QoS指标的动态优化。同时，本文结合真实RDMA链路测试数据与智算中心组网参数，构建可复现的仿真平台，从任务完成率、平均时延与资源利用率等维度验证了所提方法的有效性，为AI时代的新型智算系统提供了关键支撑。

2 系统架构

本文从整体架构、服务请求流程以及QoS保障机制3个方面开展研究，旨在为后续任务调度策略提供清晰的功能边界与建模基础。

2.1 系统整体架构

系统分层架构如图1所示。本架构采用分层设计，基于控制面和数据面解耦的思想，构建支持异构资源接入、服务编排调度与传输优化的算网融合体系结构，各层次介绍如下。

（1）控制面：负责全局资源感知、业务需求建模与调度策略生成。系统根据实时资源状态和网络反馈，为差异化业务需求动态更新调度策略，实现资源利用率与任务完成率的联合优化。

（2）数据面：承载任务数据、模型参数与推理结果的实际传输过程。系统部署RDMA网络，提供CPU/GPU间高速零复制通信通道，有效减少数据复制开销与主机CPU负载。

（3）资源抽象面：对异构计算资源（CPU、

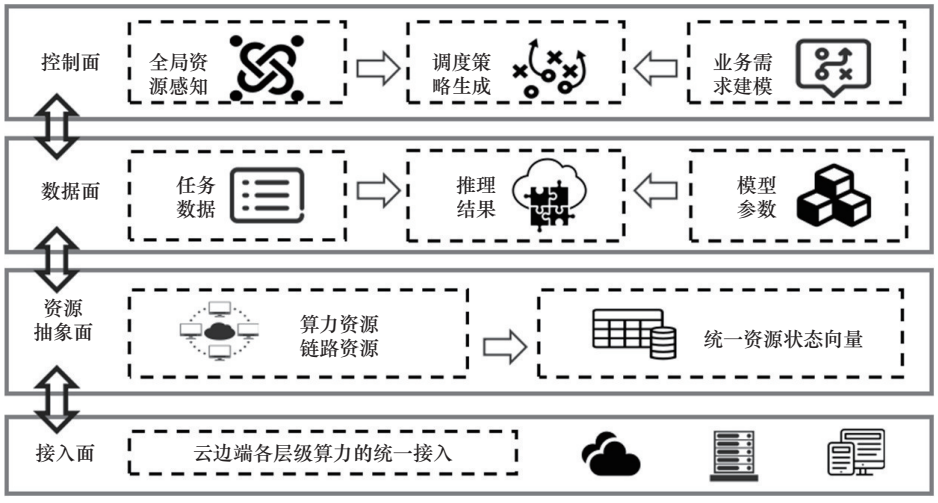

图1 系统分层架构

GPU、内存等）与链路资源（带宽、时延、占用率等）进行标准化抽象，统一表示为资源状态向量，供策略模块调用与评估。

（4）接入面：支持云边端各层级算力的统一接入，构建灵活的调度边界。边缘节点具备轻量化算力，适配交互性强、响应延迟敏感的智算任务，云端则承担大规模并行模型推理或训练任务。

综上，本系统架构通过控制面与数据面分离、资源抽象建模与高通量通信机制结合，使调度器可全面掌握全局资源状态，动态生成调度策略，并在RDMA网络支持下实现低时延任务传输，确保服务质量。

2.2　服务请求流程建模

为描述调度过程中任务的生命周期，本文构建了如图2所示的服务请求流程。

（1）任务初始化。每个智算任务（如智能体协同、大模型推理）以服务请求形式提交调度系统，请求中包含任务标识、所需资源类型、服务优先级、时延容忍度等参数。任务可抽象为多阶段服务链或有向无环图（directed acyclic graph，DAG）结构，以明确任务依赖与执行顺序。

（2）资源发现机制。系统周期性采集各计算节点与链路的状态信息，包括系统可用GPU数量、内存占用、RDMA端口负载等。该信息通过控制面广播或边缘上报机制同步，构建全局资源视图。

（3）路径构建与匹配。基于任务特性（如计算密集性、带宽敏感性、时延敏感性、链路跳数容忍度），选择合适的计算节点与RDMA链路组合，形成最优执行路径。

（4）调度执行与反馈。调度器生成具体的资源分配与调度指令，并推送至目标节点。任务完成后节点上报执行状态与QoS指标，实现闭环反馈，为策略优化提供训练信号。

（5）达成服务等级协议。节点上报任务执行的QoS指标，系统通过闭环反馈机制评估服务等级协议的达成情况，并为调度策略的持续自适应优化提供训练信号。

2.3　QoS保障需求分析

AI任务的QoS需求远高于传统服务，表现为对时延、带宽、抖动、完成率等多种指标的敏感性，且不同任务类型之间存在差异化目标。本文提出一种分层QoS需求模型，从任务、链路与系统这3个维度建模服务指标。

（1）任务级QoS指标：主要包括任务完成率、平均响应时延、排队等待时间、失败率等，体现用户侧体验。推理任务关注低时延与高稳定性，训练任务则更侧重吞吐与资源并行利用效率。

（2）链路级QoS指标：涵盖RDMA链路带宽占用、路径跳数、端到端抖动、拥塞窗口状态等，用于评估网络传输状态。高带宽低抖动链路更适合大模型键值缓存的同步与分发。

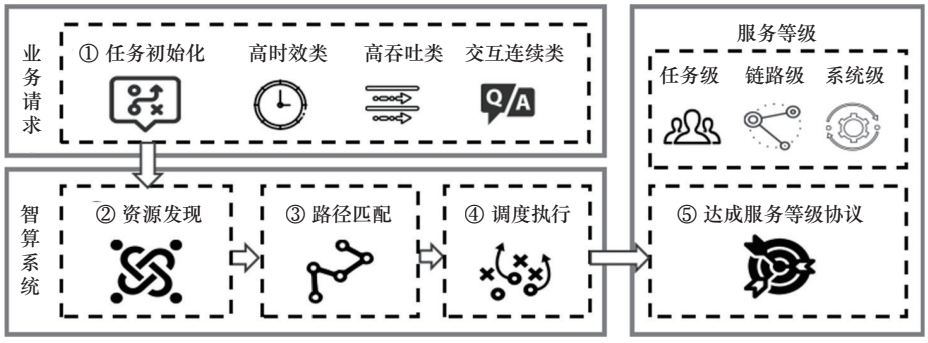

图2　服务请求流程

（3）系统级QoS指标：包括资源利用率、负载均衡程度、任务拥塞率、调度策略收敛速度等，体现系统的整体运行效率与调度机制质量。

此外，系统根据业务敏感性与响应需求，将任务划分为高时效类（如智能体推理响应）、高吞吐类（如大模型训练）、交互连续类（如多轮问答），并为每类服务匹配差异化QoS权重函数与约束边界。

3 问题建模

本文从整体系统视角出发，建立智算网络的服务请求模型与算网资源模型，引入多维QoS需求的指标建模，将调度问题形式化为一个多目标组合优化问题。

3.1 服务请求模型

在智算网络中，服务请求通常来源于大模型训练、模型推理、多智能体交互等应用场景，这些任务呈现出计算强度高、阶段依赖强、QoS约束严格等典型特征，须对其进行结构化建模。

本文将服务请求抽象为具有DAG结构的异构多阶段任务链，每个服务请求由一组相互依赖的子任务组成，每个子任务可映射至算力网络中的某一节点执行。在形式上，记任务T_i表示第i个服务请求，其任务属性由五元组定义为：

$$T_i=\{\mathrm{DAG}_i,\boldsymbol{\gamma}_i,\lambda_i,L_i,Q_i\} \tag{1}$$

其中，DAG_i表示任务的逻辑依赖结构图，图中的节点代表子任务，边表示先后执行关系及数据依赖；$\boldsymbol{\gamma}_i$表示任务的资源需求向量，描述所需GPU类型、内存容量、带宽阈值等；λ_i表示任务的到达时间，用于评估排队响应时延；L_i表示服务等级标签，用于区分不同QoS保障优先级，如高时效类、高吞吐类、低时延类任务；Q_i表示任务的QoS目标集合。

在任务执行过程中，每个子任务须被映射至物理资源节点，并沿网络链路完成数据交换与状态传输。设子任务$v_{ij}\in\mathrm{DAG}_i$表示第i个任务的第j个子任务，映射关系可定义为：$M_{ij}=(v_{ij},r_k,p_{jk})$。其中，$r_k$表示选中的计算节点，$p_{jk}$表示执行链路路径，须同时满足计算资源与传输资源约束。该映射关系决定了任务在系统中的执行路径，是调度策略的间接输出。

与传统云计算任务建模不同，智算业务不仅存在明显的阶段性结构，还具有显著的语义依赖和响应时效要求，故在建模任务结构的同时，须将服务语义连续性作为软约束引入调度优化中。此外，任务在提交调度系统时会附带服务等级信息与调度优先级。本文参考服务等级协议（service level agreement，SLA）框架设计，将服务等级分为3类：第1类为交互优先型任务，强调响应时延与上下文连续性；第2类为吞吐优先型任务，关注资源利用率与并发调度能力；第3类为通用任务，接受中等调度性能。在调度模型中，服务等级将通过权重参数影响目标函数中的优先级设置，实现对不同业务类型的差异化调度控制。

3.2 算网资源模型

由于智算网络通常包含云侧GPU集群、边缘节点、智能终端等多层次计算设施，同时网络层支持RDMA等低时延高吞吐协议，资源状态的建模必须充分考虑异构性、多层次性与动态变化特征。

（1）计算资源：本文将资源节点抽象为支持多类型算力的异构节点集合。设第k个资源节点r_k的状态描述为：

$$R_k=\{c_k^{\mathrm{CPU}},c_k^{\mathrm{GPU}},m_k,u_k,s_k\} \tag{2}$$

其中，c_k^{CPU}、c_k^{GPU}分别表示当前节点可用的CPU和GPU核数，m_k表示内存容量，u_k表示系统当前资源利用率（包括计算与存储负载），s_k表示该节点支持的并发RDMA流能力。该资源不仅可描述静态能力，也便于调度系统动态感知各节点的资源占用情况与调度可用性。

（2）网络资源：设任意两个资源节点r_m、r_n之间的状态为：

$$L_{mn}=\{b_{mn},d_{mn},o_{mn},\theta_{mn}\} \tag{3}$$

其中，b_{mn}表示链路带宽，d_{mn}表示端到端传输时延，o_{mn}表示当前拥塞程度或带宽占用率，θ_{mn}表示链路的RDMA传输能力，包括支持的最大并发传输通道数与拥塞窗口动态变化特性。该模型可用于路径选择中的性能评估与资源调度中的通信瓶颈判定。

（3）链路状态：为精准表征智算任务对网络环境的高敏感性，本文构建了端到端传输路径质量指标（path quality indicator，PQI），其定义为：

$$\mathrm{PQI}_1(t)=\alpha\cdot f(B_{\mathrm{u}})+\beta\cdot f(D_{\mathrm{j}})+\gamma\cdot f(C_{\mathrm{h}}) \tag{4}$$

其中，B_{u}表示带宽利用率，D_{j}表示时延抖动，C_{h}表示历史拥塞频率。各项指标通过Min-Max标准化函数$f(x)$处理至[0,1]，α、β、γ为对应的归一化权重系数。该变量作为生成最优节点分配决策的环境状态依据，使调度器能够实时捕捉网络拥塞趋势，从而实现对异构资源状态的深度感知。

3.3　QoS模型

为支撑智算网络中多类型任务的统一调度与差异化保障，本文从3个方面对QoS需求进行精细化建模，并构建支持服务等级分类和指标动态调整的多层级QoS约束体系。

（1）任务级建模：每个服务请求T_i附带一个QoS目标集合$Q_i=\{\Delta_i,\alpha_i,\varrho_i\}$，包含最大允许响应时延$\Delta_i$，最小服务完成率要求$\alpha_i$，以及必要时的最小吞吐率目标$\varrho_i$。系统须在调度过程中确保实际响应时延$T_{\mathrm{resp}}(T_i)\leqslant\Delta_i$，成功调度执行概率$P_{\mathrm{succ}}(T_i)\geqslant\alpha_i$。

（2）链路级建模：为提升调度路径选择的精度，本文引入链路可用性与通信可靠性建模。具体而言，对于执行路径中任意链路l_{mn}，其QoS约束表示为可用带宽$b_{mn}\geqslant b_{\mathrm{th}}$，实际传输时延$d_{mn}\leqslant d_{\max}$，拥塞度$o_{mn}\leqslant o_{\mathrm{th}}$。其中，$b_{\mathrm{th}}$为带宽可用性下限，用于保证任务传输的吞吐需求，$d_{\max}$表示任务允许的最大传输时延，用于限制路径传播时延，o_{th}表示链路拥塞度阈值，用于抑制高负载链路的选择。上述指标可通过资源状态感知模块进行周期性更新，并以PQI的形式提供给调度决策器，实现路径选择的性能引导。

（3）系统级建模：为防止资源负载过度集中与任务执行冲突，系统须引入负载均衡性约束与QoS级别差异化调度机制。所有节点的资源利用率满足$u_k\leqslant u_{\max},\forall r_k\in\mathcal{R}$，以防止资源过载；每类任务$T_i$的资源分配须满足其服务等级对应的调度优先级$\pi_i$与最小资源分配阈值$\gamma_i^{\min}$；当多任务并发执行时，须保证任务间干扰抖动不超过设定阈值，即$J\leqslant J_{\mathrm{th}}$，其中$J$表示系统平均调度抖动，$J_{\mathrm{th}}$为最大可接受抖动限值。

3.4　优化问题

本文将智算网络中的资源调度问题形式化为一个待约束的多目标优化问题，旨在满足SLA要求的前提下，实现资源高效利用、任务完成率提升与响应时延降低的统一优化。

设系统中共有M个待调度任务$\mathcal{T}=\{T_1,T_2,\cdots,T_M\}$，调度系统需在时间窗口内将任务映射至异构算力资源池中的节点集合$\mathcal{R}=\{r_1,r_2,\cdots,r_K\}$，并通过链路集合$\mathcal{L}=\{l_{mn}\}$执行任务中间数据的传输。定义任务调度决策变量$x_{ik}\in\{0,1\}$，表示任务$T_i$是否被调度到节点$r_k$上执行；链路选择变量$y_{imn}\in\{0,1\}$表示任务$T_i$在执行路径中是否使用链路$l_{mn}$。在任务调度过程中，调度智能体通过输出全局节点与链路的选择策略，间接决定子任务的具体执行节点与传输路径。

为了综合评估系统调度性能，本文构建如下三目标优化模型：

$$\min\mathcal{J}=w_1\cdot\overline{T_{\mathrm{resp}}}-w_2\cdot\overline{C_{\mathrm{rate}}}-w_3\cdot\overline{U_{\mathrm{res}}} \tag{5}$$

其中，$\overline{T_{\mathrm{resp}}}$表示所有任务的平均响应时延，$\overline{C_{\mathrm{rate}}}$为任务完成率，$\overline{U_{\mathrm{res}}}$表示系统整体资源利用率，

w_1、w_2、w_3为3个目标的权重系数，可依据服务场景调整。

除前文提到的约束外，该最小化问题还受到如下约束。

（1）资源容量约束：

$$\sum_{i=1}^{M} x_{ik}\gamma_i \leqslant C_k, \quad \forall r_k \in \mathcal{R} \tag{6}$$

其中，C_k表示节点r_k的最大算力容量。该约束表示每个资源节点的算力容量不被超过。

（2）任务分配唯一性约束：

$$\sum_{k=1}^{M} x_{ik} \leqslant 1, \quad \forall T_i \in \mathcal{T} \tag{7}$$

该约束表示一个任务只能被分配给一个节点。

（3）链路可用性与路径约束：

$$y_{imn} \cdot b_{mn} \geqslant \gamma_i^{\text{comm}}, \quad \forall l_{mn} \in \mathcal{L} \tag{8}$$

其中，γ_i^{comm}表示任务的资源需求向量中的数据传输需求。该约束表示链路带宽须满足任务的数据传输需求。

上述问题属于典型的多目标组合优化问题，具有非线性目标、多变量多约束交织、离散与连续变量混合及状态空间指数级增长等复杂特性。

4 调度算法

为应对智算网络中高动态、多QoS目标约束的资源调度挑战，本文提出一种基于RL的调度算法。

4.1 马尔可夫决策过程模型

本文将调度问题建模为一个长期的马尔可夫决策过程（Markov decision process，MDP），以实现智算网络中资源调度策略的自适应优化。该模型形式化定义了调度智能体在复杂系统中进行感知、决策与学习的流程，包含以下核心组成。

4.1.1 状态空间

状态空间用于表征系统当前的运行状态与任务调度上下文。本文设计的状态向量$\boldsymbol{s}_i$综合考虑了任务属性、节点资源状态与链路传输情况。

（1）任务特征：当前任务的计算资源需求$\boldsymbol{\gamma}_i$（包括GPU、CPU等指标）、任务服务等级标签$L_i \in \{1,2,3\}$（代表SLA等级）、QoS目标集合$Q_i = \{\Delta_i, a_i, q_i\}$。

（2）资源节点特征：每个节点剩余CPU或GPU资源比例、当前任务队列长度与平均排队等待时间、资源利用率变化率（反映负载动态性）。

（3）链路特征：实时网络资源状态与链路PQI，包括可用RDMA链路带宽、当前链路时延与利用率，此外还包含历史链路拥塞频率（反映通信稳定性）。

（4）系统统计特征：当前任务平均响应时延、当前周期任务完成率、系统GPU平均利用率。

所有特征经向量化编码与归一化处理后组成状态向量，用于神经网络输入。

4.1.2 动作空间

动作空间定义了智能体在当前状态下可执行的调度决策，本文构建如下连续动作空间。

（1）资源节点选择：输出一个概率向量$\boldsymbol{a}_1 \in \mathbf{R}^{\boldsymbol{K}}$，$\mathbf{R}^{\boldsymbol{K}}$为$K$维向量空间，表示将任务分配到$K$个计算节点上的选择概率。

（2）链路路径选择：输出链路偏好权重向量$\boldsymbol{a}_2 \in \mathbf{R}^{\boldsymbol{L}}$，$\mathbf{R}^{\boldsymbol{L}}$为$L$维向量空间，用于指示在任务传输中优先选用的RDMA路径。

（3）是否立即调度：输出二元动作$a_3 \in \{0,1\}$，判断任务是立即执行，还是延迟等待更优调度窗口。

在训练过程中，本文采用高斯策略网络输出动作分布的均值与标准差，使用重参数化技巧实现连续动作采样，确保端到端可微优化。

4.1.3 奖励函数

为了引导调度策略在多目标条件下优化QoS

表现，本文结合优化目标设计如下长期奖励函数：

$$r_t = w_1 \cdot \eta_t - w_2 \cdot \tau_t + w_3 \cdot \phi_t - w_4 \cdot \zeta_t \tag{9}$$

其中，η_t表示当前周期成功完成的任务占比，反映完成率；τ_t表示当前周期内平均任务响应时延；ϕ_t为节点资源的利用率；ζ_t为RDMA链路负载惩罚项，反映调度带来的通信压力。4个指标均为经过归一化处理后的无量纲数值。w_1~w_4为采用经验设定与网格搜索相结合的方法确定的权重因子，决定了调度器对不同QoS指标的优化偏好。具体如下。

首先根据智算业务对时延敏感性的基本要求确定权重初值，随后在仿真环境中通过网格搜索法在[0, 1]范围内进行步进调优，寻找使综合评价指标最优的组合（本文实验中设定w_1=0.4, w_2=0.3, w_3=0.2, w_4=0.1）。权重的调整反映了不同的调度倾向。若提高w_2（时延权重），算法将优先选择高性能节点以压缩处理时间，但可能会导致部分节点负载过高；若提高w_4（负载惩罚权重），算法将展现出更强的避障特性，通过牺牲部分时延性能来换取全网链路的负载均衡，防止热点拥塞的产生。

此外，为提升多智能体交互任务的语义连贯性与上下文一致性，本文引入语义一致性软约束项。定义任务T_i与其下一个任务T_j的调度间隔为δ_{ij}，若超出允许最大间隔$\delta_{\max}$，则将违反次数计入奖励惩罚：

$$r_t \leftarrow r_t - \lambda_s \cdot \mathbb{I}\left(\delta_{ij} > \delta_{\max}\right) \tag{10}$$

其中，λ_s表示惩罚强度，$\mathbb{I}(\cdot)$为指示函数。该机制可促使决策时优先保障多阶段语义任务的连续性。

4.1.4　RL目标

RL的最终目标是寻找策略$\pi_\theta(a|s)$，使得长期期望折扣奖励最大：

$$\max_\theta \ E_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] \tag{11}$$

其中，$\gamma \in (0, 1)$为折扣因子，控制未来奖励的衰减程度。

4.2　TQC-QoS算法

本文在MDP模型基础上，以截断分位数评论家（truncated quantile critics，TQC）算法[24]为核心，设计了一种融合QoS感知的智能调度算法TQC-QoS。该算法融合算力与网络状态感知能力，通过QoS指标驱动的多目标奖励函数，引导智能体在高动态环境中学习鲁棒、高效的调度策略。

相较传统的软演员–评论家（soft actor-critic，SAC）算法[25]、双延迟深度确定性策略梯度（twin delayed deep deterministic policy gradient，TD3）算法[26]，TQC算法在价值估计精度、训练稳定性以及对不确定环境的适应能力方面均较优，更适合用于面向QoS保障的智算网络资源调度问题。TQC-QoS算法架构如图3所示。为实现调度策略的自适应演化与高效部署，TQC-QoS算法采用集中训练、边缘推理的两阶段架构。在训练阶段，智能体在仿真环境中进行多轮交互，通过经验回放不断优化策略与Q值分布；在推理阶段，仅保留训练收敛的策略网络，用于实时决策和在线任务调度。其整体流程如下。

本文提出的基于TQC-QoS算法的智算网络资源调度算法的伪代码如下。

算法1　基于TQC-QoS算法的智算网络资源调度算法

输入：任务到达序列$\mathcal{J}$、智算网络环境$\mathcal{E}$、折扣因子γ、软更新系数τ、批量大小B、评论家（critic）数N_c、分位数个数N、截断数K、目标熵$\mathcal{H}_0$、最大训练步数$T_{\max}$

输出：在线调度策略π_θ

（1）初始化策略网络$\pi_\theta(a|s)$、分位数critic集合$Z_{\phi_i}(s,a)_{i=1}^{N_c}$、目标critic集合$Z_{\bar{\phi}_i}(s,a)_{i=1}^{N_c}$、温度参数$\alpha$、经验回放池$\mathcal{D}$；

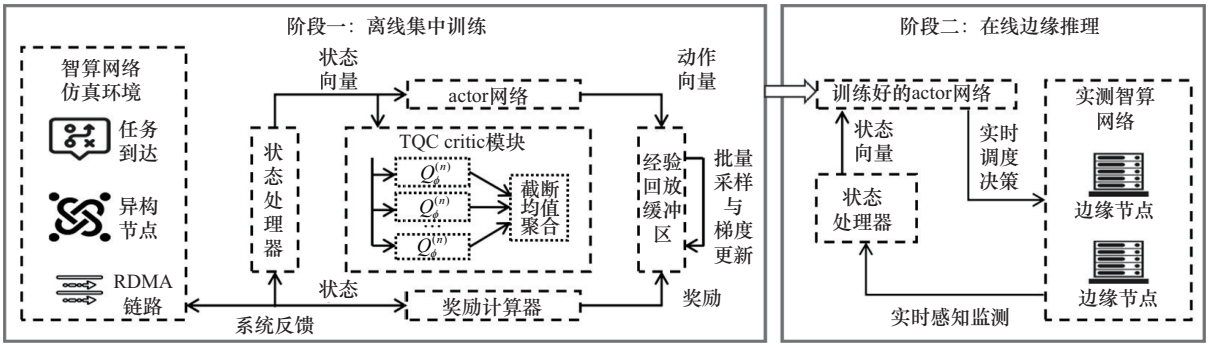

图3　TQC-QoS算法架构

（2）初始化环境$\mathcal{E}$，观测初始状态s；

（3）$t=0$；

（4） **while** $t<T_{\max}$ **do**

（5） $t=t+1$；

（6） 按策略采样动作$a\sim\pi_\theta(a|s)$；

（7） 执行动作，将转移样本(s,a,r,s',done)存入经验池$\mathcal{D}$；

（8） **if** $|\mathcal{D}|\geqslant B$ **then**

（9） 从$\mathcal{D}$中随机采样B条样本；

（10） 按当前策略采样目标动作$a'^*_j\sim\pi^*\theta(a'_j|s'^*j)$；

（11） 对每个j，由目标critic得到分位数集合$\mathcal{Z}'_j=\bigcup_{i=1}^{N_c}z_{\phi_i}^{(n)}(s'_j,a'_j)_{n=1}^{N}$；

（12） 对$\mathcal{Z}'_j$升序排序，并截断最大K个分位数，得到$\mathcal{Z}'_{j,\text{trunc}}$；

（13） 构造目标分位数集合$y_j=r_j+\gamma(1-\text{done}_j)\cdot\left(\mathcal{Z}'_{j,\text{trunc}}-\alpha\log\pi_\theta(a'_j|s'_j)\right)$；

（14） 根据式（8）更新分位数critic；

（15） 根据式（9）更新策略网络；

（16） 根据式（10）更新温度参数；

（17） 软更新目标critic:$\overline{\phi}_i\leftarrow\tau\phi_i+(1-\tau)\overline{\phi}_i$；

（18） **end if**

（19） $s=s'$；

（20） **end while**

（21）输出训练后的在线调度策略π_θ

首先，初始化网络参数与经验回放池（见算法1第（1）行）以模拟多节点智算网络的运行状态，包括任务到达分布、异构资源负载与RDMA链路时延模型。其次，在每个调度周期内，智能体根据当前状态S通过策略网络$\pi_\theta(a|s)$输出动作分布参数（见算法1第（5）、（6）行），并经重参数化技巧采样动作$a\sim\pi(a|s)$。再次，系统执行该动作获得奖励r与新状态s'，并将四元组(s,a,r,s',done)存入经验池中（见算法1第（7）行）。当经验池累积足够样本后，进入批量训练阶段：从经验池中采样B条样本，利用目标网络计算截断后的分位数目标值，以缓解Q值过估计问题（见算法1第（9）～（13）行）。最后，依次根据式（8）、式（9）和式（10）更新critic网络、演员（actor）网络及温度参数（见算法1第（14）～（16）行）。

神经网络的具体更新方式如下。

（1）critic网络更新

TQC算法不直接计算单一Q值，而是学习分位数函数$Q_\phi^{(n)}(s,a)$，并利用截断平均减少高估偏差。其更新目标为：

$$\mathcal{L}_Q=E\left(s,a,r,s'\right)\left[\sum_{n=1}^{N}\varrho_{\tau_n}\left(r+\gamma\bar{Q}_{\text{trunc}}(s',a')-Q_\phi^{(n)}(s,a)\right)\right]\quad(12)$$

其中，$\varrho_{\tau_n}(\cdot)$为分位数损失函数，$\bar{Q}_{\text{trunc}}$表示去除上分位数后的平均Q值估计。

（2）actor网络更新最大化分布式软Q值的期望以提升策略性能：

$$\mathcal{L}_\pi = E_s\left[\alpha \log \pi_\theta(a|s) - Q_{\text{trunc}}(s,a)\right] \quad (13)$$

其中，α为熵系数，用于调节探索与利用的平衡。

（3）温度参数更新：TQC 算法采用可学习的熵温度自适应机制：

$$\mathcal{L}_\alpha = E_{\boldsymbol{a}}\left[-\alpha\left(\log \pi_\theta(a|s) + \mathcal{H}_0\right)\right] \quad (14)$$

其中，$\mathcal{H}_0$为目标策略熵。

推理阶段仅保留策略网络用于前向推理，输入为系统当前状态向量，输出为连续动作变量。该过程不涉及梯度计算，推理复杂度约为$O(d_{\text{h}} \cdot d_{\text{out}})$，可在边缘节点或集中调度控制器上毫秒级执行。

在时间复杂度方面，TQC-QoS 算法的单轮训练开销主要来源于多个分位Q网络的反向传播，整体复杂度为：

$$O\left(B \cdot N_{\text{q}} \cdot \left(d_{\text{h}}^2 + d_{\text{h}} \cdot d_{\text{out}}\right)\right) \quad (15)$$

其中，B为批处理大小，N_{q}为分位Q网络数，d_{h}为隐藏层宽度，d_{out}为动作维度。推理阶段仅涉及策略网络的前向计算，复杂度约为$O(M \cdot H^2)$，其中M为输入维度，H为隐藏层宽度，在常规算力节点上即可实现毫秒级响应。

在空间复杂度方面，本文设计的状态空间维度D_{s}与网络规模呈线性增长关系。设网络节点数为N，关键链路数为L，则$D_{\text{s}} \approx n_{\text{task}} + N \cdot n_{\text{node_feat}} + L \cdot n_{\text{link_feat}}$。在大规模智算网络场景下，虽然状态维度增加，但由于 TQC 算法具有更强的分布表征能力，其收敛速度和策略稳定性相较于传统算法具有更明显的优势。此外，得益于集中训练、分布推理的模式，复杂的计算压力集中在具备高性能计算能力的控制面，而执行节点仅须运行轻量化的前向推理。这种架构确保了算法在节点规模扩展至百级甚至千级时，依然能保持高效的调度性能。

综上，通过引入分布式价值估计与截断机制，TQC-QoS 算法在 RL 稳定性、调度策略精度以及多 QoS 目标自适应方面表现出更优性能，为智算网络提供了一种可扩展、可落地的智能调度方案。

5　仿真分析

为验证所提智能调度算法的有效性与适用性，本文基于自建的智算网络仿真平台进行了系统实验。

5.1　仿真实验设置

实验平台结合 GEANT 实际网络数据集与事件驱动式任务模型，能够较真实地反映算力节点异构、RDMA 链路动态变化及任务多样性共存的运行特征。该平台构建云、边、端 3 层异构拓扑：云侧节点提供大规模 GPU 集群算力，边缘节点部署轻量化 GPU/CPU 资源以支撑交互型任务，终端节点用于任务发起与状态反馈。链路带宽范围为 1~10 Gbit/s，端到端时延为 0.3~1.2 ms，带宽与拥塞状态随时间动态变化，以反映真实负载波动。任务到达服从泊松分布，平均到达率λ为每秒 10 次请求。任务类型分为交互型、吞吐型和通用型 3 类（比例为 0.34 : 0.33 : 0.33），数据量为 32~128 MB，任务链深度为 2~3 阶段。

本文设置 5 种算法与所提 TQC-QoS 算法进行对比。

（1）近端策略优化（proximal policy optimization，PPO）算法[27]：作为经典的在线策略算法，其通过裁剪概率比率确保策略更新平稳。但在高动态的智算网络中，PPO 算法对样本的利用率较低，且在面对多维、连续的动作空间时，往往难以快速收敛至全局最优的 QoS 调度策略。

（2）TD3 算法：通过截断双Q学习缓解了价值函数的高估偏差。然而，在算网融合场景下，环境状态具有高度不确定性，TD3 的单一Q值估计难以精确刻画奖励分布的统计特性，导致其在极端网络工况下的鲁棒性不足。

（3）贪心算法：基于最小响应时延的贪心调

度策略。该方法计算开销极低，但在多目标优化场景下，仅关注局部最优。

（4）轮询算法：按序循环分配任务，其优势在于确保了节点间的负载均衡，但难以满足大模型推理等高敏感业务的时延约束。

（5）随机算法：采用随机分配策略作为性能下界的基线参考。

本文提出的TQC-QoS算法针对上述RL算法的局限性，引入截断分位数机制，不仅能有效抑制价值过估，更能通过分布式的价值表示捕捉网络环境的随机性，从而在复杂约束下实现更精准的QoS风险控制。所有DRL类算法均采用两层全连接神经网络作为策略与价值网络，超参数保持一致。训练阶段采用集中式训练、分布式推理的架构，在同一数据集与环境配置下进行对比。

为全面评估各算法在不同场景下的性能，本文从任务层、链路层与系统层选取以下指标。

- 平均响应时延：任务从提交到完成的平均时延。
- 任务成功率：在时延约束内成功完成任务的比例，体现QoS保障能力。
- 算力平均利用率：算力资源使用效率。
- 链路平均占用率：RDMA网络的带宽利用情况。
- 平均步奖励：由每回合所有步所获奖励的平均值计算得到，反映整个连续调度过程中任务完成率、响应时延及资源利用率三者间的整体优化水平。所有指标均由仿真环境在每个调度周期实时采集，并在多次实验中取平均值及标准差。

5.2 性能指标与结果分析

（1）训练性能对比

首先，本文评估了3类RL算法在智算网络环境下的收敛特性与训练稳定性。

各算法折扣因子设为0.99，批量大小为512，学习率均设置为测试所得最佳值。训练阶段每个回合包含50个调度周期，总训练轮数为3 000。图4展示了4种算法在训练阶段的平均单步奖励随训练回合数变化的收敛曲线，灰色阴影部分表示原始数据波动范围。

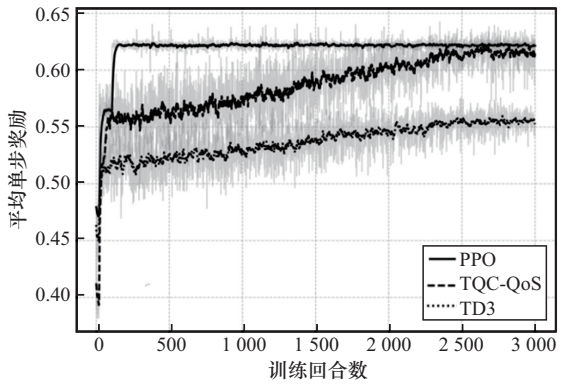

图4 4种算法在训练阶段的平均单步奖励随训练回合数变化的收敛曲线

从图4可以看出，3种算法在收敛速度与稳定性方面表现出明显差异。总体而言，PPO算法的收敛速度最快且曲线最为平稳，在约200轮训练后即趋于稳定，且平均单步奖励始终保持最高水平，表明其基于剪切目标函数的策略更新机制能够在高动态环境中有效平衡探索与利用，从而实现快速收敛。TQC-QoS算法在训练初期波动较大，但整体趋势为持续上升，最终在约2 500轮后收敛至较高奖励水平，体现出其通过截断Q值降低过估计风险、提升训练稳定性的优势。TD3算法在早期阶段收敛较慢且波动幅度较小，最终奖励值最低，说明其双Q网络修正机制在多QoS目标下存在一定的学习滞后性。

此外，图5展示了不同算法在训练过程中各主要指标的收敛情况。整体上，各指标在训练初期均经历快速变化阶段，随后逐步趋于稳定。具体而言，PPO算法在训练约150轮后即可实现平稳收敛，其时延显著低于TQC-QoS和TD3，且成功率始终保持在0.9以上，表明其能在复杂算网环境中实现较优的QoS保障。3种算法的GPU利用率与链路占用率在训练后期均趋于低稳态，说

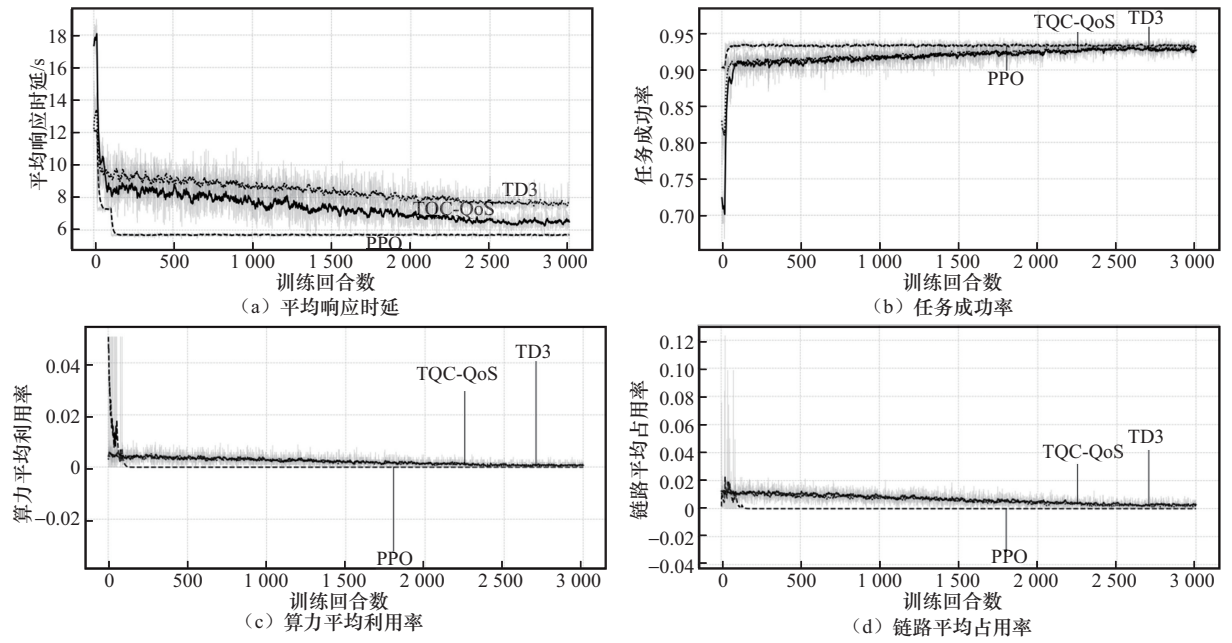

图5　不同算法在训练过程中各主要指标的收敛情况

明调度策略已学会在资源利用与网络负载之间进行平衡。

综合比较可知，在离线训练时，PPO 算法在收敛速度与稳定性方面最优，TQC-QoS 则稍差一些，而 TD3 算法受限于过于保守的策略更新。该结果验证了在智算网络调度场景中，引入多目标奖励建模与策略约束机制有助于提升 RL 算法的收敛效率与鲁棒性，为复杂资源调度提供了可行的优化路径。

（2）推理性能对比

为验证所设计的 RL 调度算法在复杂算网环境中的综合效能，对 TQC-QoS 算法、PPO 算法、TD3 算法、贪心算法、轮询算法与随机算法这 6 种算法进行了性能对比分析。测试结果主要包括两类指标：一类为策略收益指标，即平均步奖励，用于衡量算法在长期交互中的累积收益；另一类为系统运行指标，包括平均响应时延、任务成功率、算力利用率与链路占用率，用以综合评估算法在任务调度、资源匹配与网络负载方面的优化能力。

各算法平均步奖励对比如图 6 所示。由图 6 可知，不同算法的平均步奖励存在显著差异。TQC-QoS 算法与 PPO 算法整体表现最优，平均步奖励分别为 0.630 与 0.622，均显著高于 TD3 算法、贪心算法、传统轮询算法以及随机算法。这说明基于分位数回报估计的 TQC-QoS 算法在复杂动态算网环境中能够更稳定地实现策略优化，其分布式价值函数设计有效缓解了训练不稳定与策略过估计问题；而 PPO 算法凭借其剪切式目标函数与约束式策略更新机制，展现出收敛速度快、训练稳定性高的特点，但在实际推理时效果较差。相比之下，TD3 算法在奖励表现上略低，主要由于其在高维动作空间中对策略噪声的抑制不足，易导致收敛缓慢。启发式与随机调度算法虽能在部分场景下获得可接受的性能，但整体奖励水平明显偏低，体现出其在多目标复杂优化问题中缺乏全局搜索与自适应学习能力。

图 7 展示了各算法多指标归一化性能雷达图。由图 7 可知，TQC-QoS 算法与 PPO 算法形成了最外层包络，表明其在多维 QoS 指标上综合性能最佳。其中，TQC-QoS 算法在任务成功率和资源利用率方面表现突出，能够在保证高成功率的同时保持较高的算力利用水平；PPO 算法在时延控制

上表现最佳，说明其稳定的策略更新机制有助于形成低时延的任务映射路径。相比之下，TD3算法在各维度性能上较为均衡，但整体略逊于TQC-QoS算法与PPO算法；贪心算法虽在算力利用率上最高，但以牺牲时延与成功率为代价，呈现高资源占用率、低服务稳定性的特征；轮询算法与随机算法在所有维度上表现较弱，尤其在成功率和时延两项上明显不足，难以满足智算网络环境下的多目标优化需求。

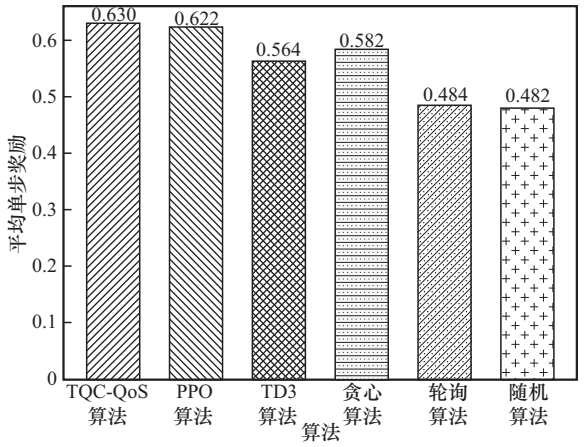

图6　各算法平均步奖励对比

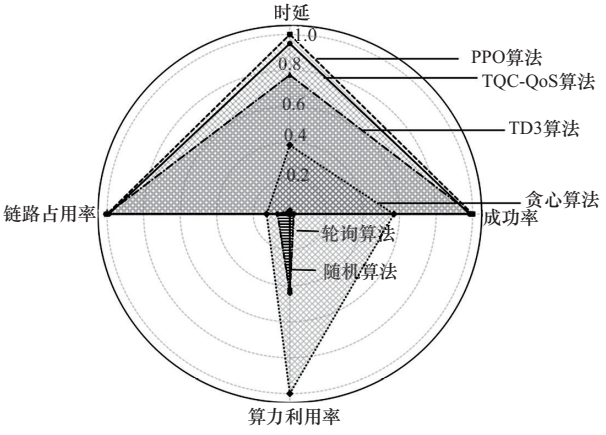

图7　各算法多指标归一化性能雷达图

综合图6与图7可知，基于DRL的TQC-QoS算法与PPO算法在奖励收益与QoS平衡性方面均表现出较强优势。其中，TQC-QoS算法在推理阶段的性能更稳健，能够实现较高资源利用率与任务成功率；而PPO算法在训练效率与时延控制方面更具优势，适合需要快速策略收敛的场景。因此，在需要长期稳定运行与跨域任务调度的智算网络系统中，TQC-QoS算法更具工程应用价值，而在对实时性要求更高的动态业务场景中，PPO算法则更适合部署为在线优化核心策略。

（3）算法鲁棒性对比

为进一步验证所提调度算法在多场景扰动下的泛化性与稳定性，本文设计了10组随机种子下的鲁棒性实验。测试环境保持与训练阶段一致，均基于GEANT真实网络拓扑，随机扰动链路带宽、任务到达率与算力节点可用性等关键参数。每轮测试均独立初始化任务分布与网络状态，对6种算法进行推理阶段评估，比较其在平均步奖励、平均响应时延和任务成功率3个维度的性能分布特性。

奖励分布箱线图如图8所示。从图8可以看出，不同算法在平均步奖励方面存在显著差异。TQC-QoS算法与PPO算法在所有测试轮次中均表现出较高且稳定的奖励水平，其中TQC-QoS算法的中位数最高且波动最小，说明其在策略评估与更新过程中具备更强的稳定性与泛化能力。PPO算法奖励略低于TQC-QoS算法，主要是因为其对资源利用率的优化较差。相比之下，TD3算法与贪心算法的奖励水平明显偏低，而传统的轮询算法与随机算法在奖励分布上呈现出明显的集中下移，说明其缺乏对任务状态与网络动态的自适应调节能力。总体而言，基于分位数回报的TQC-QoS在长期性能与稳定性之间实现了较优平衡。

图9、图10展示了6种算法的平均响应时延箱线图和任务成功率箱线图。从平均响应时延来看，PPO算法与TQC-QoS算法明显优于其他算法，分别保持在6 s与6.1 s左右，且波动区间较小，表明其在动态链路拥塞与异构算力分布条件下能够有效完成任务调度。TD3算法与贪心算法的时延中位数相对较高，而轮询与随机策略的时延普遍超过12 s，说明传统策略难以适应复杂任务

到资源的多目标优化。成功率方面，TQC-QoS 算法、PPO 算法与 TD3 算法均保持在 0.93 以上，贪心算法略低（约 0.89），而轮询算法与随机算法明显不足（小于 0.87）。综合来看，TQC-QoS 算法在时延控制与成功率保障上均表现出更优的分布特性。

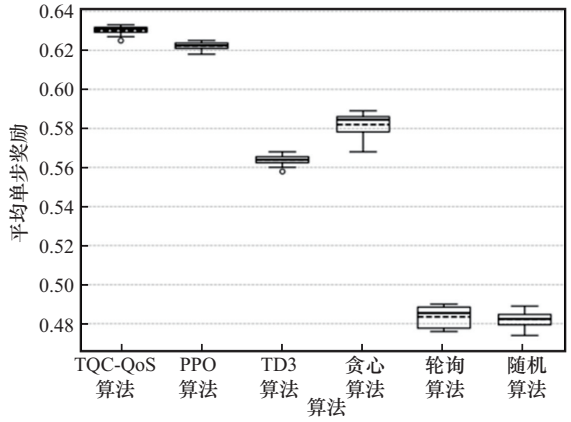

图 8　奖励分布箱线图

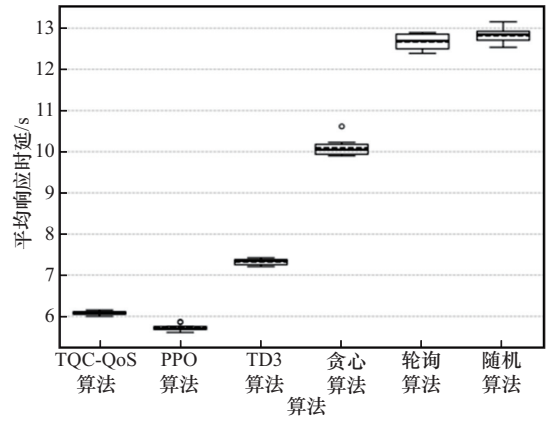

图 9　平均响应时延箱线图

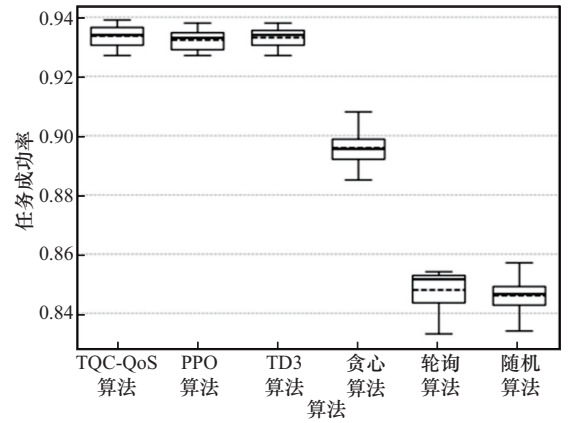

图 10　任务成功率箱线图

6　结束语

本文围绕智算网络中面向 QoS 需求保障的资源调度问题，系统研究了调度建模方法与智能调度算法设计。首先，从大模型推理与多智能体协同等新型业务特征出发，构建了融合任务结构、异构算力资源与网络链路状态的调度模型，对任务–资源–链路之间的映射关系进行了统一刻画，并引入多维 QoS 指标对调度目标进行形式化描述。其次，本文提出了一种基于 RL 的智能调度算法框架，并选用 TQC-QoS 算法作为核心学习机制，通过分布式价值建模与截断机制提升调度策略的稳定性与鲁棒性，实现了多 QoS 目标约束下的自适应资源调度。最后，通过构建贴近实际智算中心组网结构的仿真平台，引入网络实测数据进行验证。实验结果表明，本文方法在关键 QoS 指标上均优于对比方案，验证了所提调度模型与算法在复杂智算网络场景下的有效性与实用价值。

参考文献：

[1] Weisz J D, He J, Muller M, et al. Design principles for generative AI applications[C]//Proceedings of the CHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2024: 1-22.

[2] Agrawal A, Kedia N, Panwar A, et al. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve[C]//Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). Santa Clara: USENIX Association, 2024: 117-134.

[3] Hu J B, Shen H Q, Liu X C,et al. RDMA transports in datacenter networks: survey[J]. IEEE Network, 2024, 38(6): 380-387.

[4] Sun B, Huang Z, Zhao H, et al. . Llumnix: Dynamic scheduling for large language model serving[C]//Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). Santa Clara: USENIX Association, 2024: 173-191.

[5] Sun B Q, Theile M, Qin Z Y, et al. Edge generation scheduling for DAG tasks using deep reinforcement learning[J]. IEEE Transactions on Computers, 2024, 73(4): 1034-1047.

[6] Huang J T, Golubchik L, Huang L B. When Lyapunov drift based queue scheduling meets adversarial bandit learning[J]. IEEE/ACM Transactions on Networking, 2024, 32(4): 3034-3044.

[7] Ahmed G, Sheltami T, Ghaleb M, et al. Energy-efficient Internet of drones path-planning study using meta-heuristic algorithms[J]. Applied Sciences, 2024, 14(6): 2418.

[8] Zhang Y N, Tzeng E, Du Y L, et al. Large-scale reinforcement learning for diffusion models[C]//Computer Vision-ECCV 2024. Cham: Springer Nature Switzerland, 2024: 1-17.

[9] Paul A, Singh K, Kaushik A, et al. Quantum-enhanced DRL optimization for DoA estimation and task offloading in ISAC systems[J]. IEEE Journal on Selected Areas in Communications, 2025, 43(1): 364-381.

[10] Gangidi A, Miao R, Zheng S B, et al. RDMA over Ethernet for distributed training at meta scale[C]//Proceedings of the ACM SIGCOMM 2024 Conference. New York: ACM Press, 2024: 57-70.

[11] Cai K, Wu Q W, Zhou M C, et al. Dynamically scheduling deadline-constrained interleaved workflows on heterogeneous computing systems[J]. IEEE Transactions on Services Computing, 2025, 18(2): 758-769.

[12] Ma X, Zhou A, Zhang S, et al. Dynamic task scheduling in cloud-assisted mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(4): 2116-2130.

[13] Lu Y Z, Yang L, Yang S X, et al. An intelligent deterministic scheduling method for ultralow latency communication in edge enabled industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2023, 19(2): 1756-1767.

[14] Cheng Z R, Zhang W T, Yang D, et al. Intelligent end-to-end deterministic scheduling across converged networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(4): 2504-2518.

[15] Zhao G M, Wang J Z, Xu H L, et al. Joint request updating and elastic resource provisioning with QoS guarantee in clouds[J]. IEEE/ACM Transactions on Networking, 2024, 32(1): 110-126.

[16] Ali J, Song H H, Roh B H. An SDN-based framework for E2E QoS guarantee in Internet of Things devices[J]. IEEE Internet of Things Journal, 2025, 12(1): 605-622.

[17] Zhao L L, Chi X F, Ma S D. Delay-QoS-aware local-information-driven multiple access for MTC networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(5): 5848-5862.

[18] Zhu J, Wang S W. QoS-guaranteed resource allocation in mobile communications: a stochastic network calculus approach[J]. IEEE/ACM Transactions on Networking, 2024, 32(6): 5159-5171.

[19] Peng Y, Tang X G, Zhou Y Q, et al. Computing and communication cost-aware service migration enabled by transfer reinforcement learning for dynamic vehicular edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23 (1): 257-269.

[20] Gao X Y, Sun Y P, Chen H, et al. Joint computing, pushing, and caching optimization for mobile-edge computing networks via soft actor-critic learning[J]. IEEE Internet of Things Journal, 2024, 11(6): 9269-9281.

[21] Zhang X Y, Liu J, Zhang R, et al. Energy-efficient computation peer offloading in satellite edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(4): 3077-3091.

[22] Xu C, Zhang P F, Xia X F, et al. Digital-twin-assisted intelligent secure task offloading and caching in blockchain-based vehicular edge computing networks[J]. IEEE Internet of Things Journal, 2025, 12(4): 4128-4143.

[23] Dai Y Y, Rao X Y, Gu B, et al. Graph learning-based multiuser multitask offloading in wireless computing power networks[J]. IEEE Internet of Things Journal, 2025, 12(15): 29230-29239.

[24] Kuznetsov A, Shvechikov P, Grishin A, et al. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics[C]//Proceedings of the 37th International Conference on Machine Learning. New York: ACM Press, 2020: 5556-5566.

[25] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proceedings of the 35th International Conference on Machine Learning (ICML 18). Stockholm: PMLR, 2018: 1861-1870.

[26] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]//Proceedings of the 35th International Conference on Machine Learning (ICML 18). Stockholm: PMLR, 2018: 1587-1596.

[27] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[PP]. V2. arXiv(2017-08-28) [2026-01-10] arXiv:1707.06347, 2017.

[作者简介]

赵明（1980−），男，中信数字科技集团公司高级工程师，主要研究方向为云计算、数据通信、网络信息与安全、人工智能、数据中心等。