计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(6): 1584−1596, 2026

云—边—端协同下基于区块链的动态优先级调度策略

李天宇　王兴伟　曾荣飞
（东北大学计算机科学与工程学院　沈阳　110170）

A Blockchain-Based Dynamic Priority Scheduling Scheme for Edge-Cloud-End Collaboration

Li Tianyu, Wang Xingwei, and Zeng Rongfei
（*College of Computer Science and Engineering, Northeastern University, Shenyang* 110819）

Abstract　In recent years, edge-cloud-end collaboration (ECEC) has emerged as an efficient and promising technology for empowering various computation-intensive applications, effectively bridging the gap between data analytics and physical states. Within ECEC, there is an urgent need for a reliable and optimal task scheduling scheme to maximize resource utilization and provide satisfactory services to end users. However, existing scheduling schemes still face numerous challenges, such as the instability and complexity of network topologies, the highly dynamic nature of task priorities, and the lack of trust mechanisms among end users. We propose a blockchain-assisted dynamic priority scheduling strategy capable of transmitting large-scale tasks to edge or cloud servers in dynamic scenarios. We first model the resource allocation and task scheduling problem, establish a dynamic priority assignment model, and employ the analytic hierarchy process to determine the priority of sub-tasks generated from large-scale tasks. Subsequently, we adopt a blockchain-based Lagrangian multiplier distributed island genetic algorithm, which transforms the formulated problem into a convex optimization form to achieve optimal resource allocation under specific schemes. Furthermore, the proposed architecture incorporates a blockchain verification mechanism to enhance system stability and strengthen data privacy protection. Extensive simulation results demonstrate that, compared with several baseline schemes, the proposed solution achieves a 10% reduction in both total system delay and privacy overhead in ECEC.

Key words　edge-cloud-end collaboration; task scheduling; resource allocation; blockchain; distributed island genetic algorithm

摘　要　近年来，云—边—端协同（edge-cloud-end collaboration, ECEC）已成为赋能各类计算密集型应用的高效且有前景的技术，其有效弥合了数据分析与物理状态之间的鸿沟。在ECEC中，亟需一种可靠且最优的任务调度方案，以最大限度地提升资源利用率并为终端用户提供满意的服务。然而，现有的调度方案仍面临诸多挑战，如网络拓扑结构的不稳定与复杂性、任务优先性的高度动态性和终端用户之间信任机制的缺失。提出了一种基于区块链辅助动态优先级调度策略，能够在动态场景下将大规模任务传输至边端或云端服务器。首先对资源分配与任务调度问题进行建模，建立动态优先级分配模型，采用层次分析法确定大规模任务生成子任务的优先级。采用了基于区块链的拉格朗日乘子的分布式岛模型增强遗传算法，将所建问题转化为凸优化形式，实现特定方案下的最优资源分配。此外，所提出的架构还引入区

收稿日期：2025-08-19；**修回日期**：2026-03-04
基金项目：国家自然科学基金项目（62432003, U25A20431, 62572110）
This work was supported by the National Natural Science Foundation of China (62432003, U25A20431, 62572110).
通信作者：王兴伟（wangxw@mail.neu.edu.cn）

块链验证机制，以提升系统稳定性并加强数据隐私保护。大量仿真实验结果表明，与多基线方案相比，所提方案在 ECEC 中可实现降低系统总时延和隐私开销的 10%。

关键词 云—边—端协同；任务调度；资源分配；区块链；分布式岛屿遗传算法

中图法分类号 TP393.02

DOI: 10.7544/issn1000-1239.202550600 **CSTR:** 32373.14.issn1000-1239.202550600

随着智能设备和移动互联网的快速普及，云计算（cloud computing, CC）作为一种集中式的数据处理与存储解决方案，在应对大规模计算任务方面展现出显著优势[1-2]。然而，云服务器（cloud servers, CS）与终端设备（end devices, ED）之间存在较大的通信延迟与带宽限制。为缓解上述问题，移动边缘计算（mobile edge computing, MEC）通过将计算资源调度至网络边缘服务器（edge servers, ES）附近，从而降低了任务响应延迟并提升了传输效率。然而，MEC 在实际应用中仍面临诸如 ES 的资源受限、计算能力不足以及负载动态波动等挑战[3-4]。

为应对移动服务与设备爆发式增长带来的上述挑战，云—边—端协同（edge-cloud-end collaboration, ECEC）调度作为一种极具前景的解决方案应运而生。ECEC 通过结合 CS 强大的计算与存储能力以及 ES 的低延迟和高响应性的优势，实现了资源的优势互补，显著提升了服务的效率与灵活性[5]。此外，ECEC 调度在优化数据传输、资源分配和能耗方面也展现出显著优势。然而，在复杂场景下，物联网设备之间缺乏信任是首要挑战。此外，由于边缘设备之间通信链路较长，且需处理如智能金融风控、个性化推荐和智能视频监控等大规模任务，可能需要将任务调度至当前通信范围之外的边缘站[6]。图 1 介绍了任务调度相关的具体流程，其包括任务分解、确定依赖关系、任务同步、组内通信和任务聚合这 5 个过程。在此类情形下，网络链路的不稳定性会削弱 ECEC 网络的安全性与可扩展性[7]。

针对上述挑战，发展基于区块链的 ECEC 集成框架是一种潜在的解决方案。区块链最初应用于比特币，是一种分布式账本机制。其去中心化模式通过维护不可篡改账本，实现了在彼此不信任的参与方之间建立可信关系[8]。当任务被调度至 ES 或 CS 时，相关调度信息会与区块的创建同步记录在当前区块链账本中。关键功能如认证与防篡改则通过智能合约与特定共识机制得以实现[9]。

由于 ECEC 网络中设备的异构性以及场景的复杂性和多样性，主流深度强化学习方法在大规模决策空间和多维状态空间探索中表现出较强的问题求解能力[10]。然而，在大规模场景下，动作和状态空间将急剧增大，训练时间显著延长，进一步加剧了收敛难度。传统机器学习方法高度依赖专家经验，往往导致系统延迟过高。因此，亟需部署轻量化的智能调度算法，以满足 ECEC 的安全性与高效性需求。

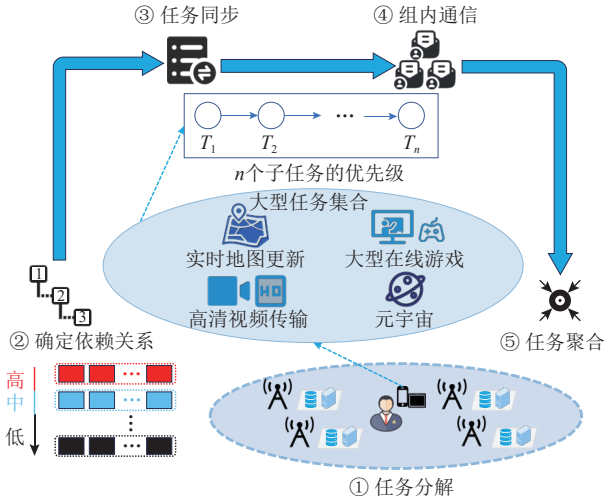


Fig. 1 Flowchart of task scheduling in ECEC

图 1 云边端协同中的任务调度流程图

针对上述挑战，本文提出了一种分布式启发式算法用于调度决策。通过拉格朗日乘子法将原始非凸问题转化为凸问题，从而获得特定决策下的最优资源分配方案。本文提出的方案——BBLM-DIGA（blockchain-based Lagrangian multiplier distributed island genetic algorithm），涵盖卸载决策、资源分配与隐私保护，旨在最小化系统总时延与隐私开销，同时确保区块高效分配。主要创新点如下：

1）为实现 ECEC 全域的安全资源共享与统一监管机制，提出了一种异构模型，并引入区块链技术。该模型包括区块链构建与更新、决策等模块，有效监控任务卸载过程，提升系统整体稳定性。

2）为了解决 ECEC 网络中大规模任务引起的子任务协调问题，建立了一个动态优先级分配模型。采用层次分析法确定大规模任务生成子任务的优先级。优先级的设置综合考虑任务大小、数据量、截

止日期、紧急程度等因素，从而确定子任务的最优执行顺序。

3)为降低总处理时延并满足多设备的隐私约束，提出了联合优化方案，涵盖任务卸载决策、计算资源分配和区块大小分配。基于拉格朗日乘子的分布式岛模型遗传算法将原问题转化为凸问题，实现最优资源分配与高效卸载。

4)通过与多种算法对比，验证了 BBLM-DIGA 方案的有效性。该方案依托区块链验证机制提升数据隐私保护能力，在保证系统性能最优的同时，保持了较低的计算复杂度。

1 相关工作

现有大多数关于 ECEC 中任务调度的研究，主要聚焦于大规模通信链路下的动态场景，旨在通过将终端设备的计算任务部分或全部调度至 ES 或 CS，从而最小化系统总成本[11]。例如，文献 [12] 提出了一种基于流水线的调度方案，利用逐次凸近似方法，联合优化 ECEC 中的协同计算调度与资源分配，以降低移动设备的总时延。另有研究基于博弈论方法，针对 ECEC 系统中的多用户资源竞争设计了任务调度方法。该方法将问题建模为非合作博弈，并通过分析纳什均衡，实现任务处理效率与公平性的优化[13]。此外，还有学者提出了基于区块链的任务调度与资源分配框架，结合 Lyapunov 优化和混合型深度强化学习算法，实现任务调度、资源管理与区块挖掘等过程的联合优化[14]。

任务调度指的是资源分配与任务执行顺序的确定，以保障任务按特定策略完成，如基于优先级、最短任务优先或轮询等方法[15]。例如，有研究在车载边缘网络中建立排队模型以管理任务流，并采用改进的深度强化学习算法实现调度方案[16]。亦有学者提出基于深度强化学习的多优先级计算调度方案，面向时延敏感和受干扰约束的工业无线网络，通过优化计算决策、传输功率和调度比例力求降低任务成本[17]。此外，还有基于 ADMM-Consensus 的任务调度算法，针对新鲜度感知下的多优先级调度问题，同时采用分布式异步算法的收敛性与效率[18]。另有研究关注多无人机协作，将任务优先级与二元调度相结合，通过深度强化学习算法联合优化无人机轨迹与资源分配，以最大化系统长期平均收益[19]。

区块链作为提升任务调度安全性的关键技术，尤其适用于大规模 ECEC 场景[20]。其独特的共识机制和不可篡改的区块结构有效保障了用户间的信息隐私。有研究为激励车辆共享计算资源并确保资源分配的可靠性，提出了基于深度强化学习的区块链智能合约计算调度算法，并设计了共识节点选择机制以提升共识效率[21]。还有研究提出了基于分片技术的可扩展区块链 MEC 架构，并结合可靠的计算调度方法优化任务调度决策[22]。此外，针对多基站 ECEC 网络，相关研究通过 Lyapunov 优化理论简化问题复杂度，并提出了新型深度强化学习架构，以提升模型学习性能[23]。另有工作设计了基于区块链的空天地一体化模型，采用多智能体深度联邦强化学习，联合优化 CPU 频率、能耗、资源分配及区块大小，从而降低执行成本和隐私开销[24]。

综上可知，现有研究对于大规模任务在 ECEC 中远距离通信所面临的效率与安全性挑战关注不足。主要原因在于边缘节点分布广泛、单个边缘设备内各节点间缺乏信任，以及任务规模庞大导致调度决策空间过大。这些因素不仅阻碍了决策算法的收敛，也加大了安全且轻量化调度方案的设计难度。

2 系统模型与问题形成

2.1 网络模型

如图 2 所示，ECEC 中的任务协同网络架构主要由 1 个 CS、多个 ES 和终端设备(end device, ED)组成。CS 负责系统或区域内的数据集中存储、协同计算优化和资源整合，编号为 0。ES 由基站和 MEC 服务器构成，不同类型的 ES 部署于不同 ECEC 场景，承担任务调度与资源分配，ES 集合记为 $E=\{es_1,es_2,\cdots,es_N\}$，则其联合集合为 $E\cup\{0\}$。

终端层由不同场景下的各类终端设备及其产生的大规模应用组成，如工业物联网、车载边缘计算和智慧城市等典型场景，以及农田灌溉、智能驾驶、智能制造等应用。这些大规模任务需按优先级划分为若干子任务分别处理。例如，在智能驾驶中，紧急避障任务的优先级远高于地图实时更新和驾驶行为分析任务。终端设备集合记为 $U=\{ed_1,ed_2,\cdots,ed_M\}$。

由于 ES 之间距离较远，通信距离显著影响传输速率，传统香农公式难以应对动态变化的无线信道。为此，任务调度需依赖无线数据传输，并采用动态调度方法以适应信道变化。本文采用正交频分多址作为网络资源调度方式，其中任务上行速率 r_{ij}^{u} 定义为

$$r_{ij}^{\mathrm{u}}=B_{\mathrm{w}}\mathrm{lb}\left(1+\frac{P^{\mathrm{u}}|g_{ij}|^2 l_n^{-\delta}}{N_0+I_n^{\mathrm{u}}}\right), \tag{1}$$

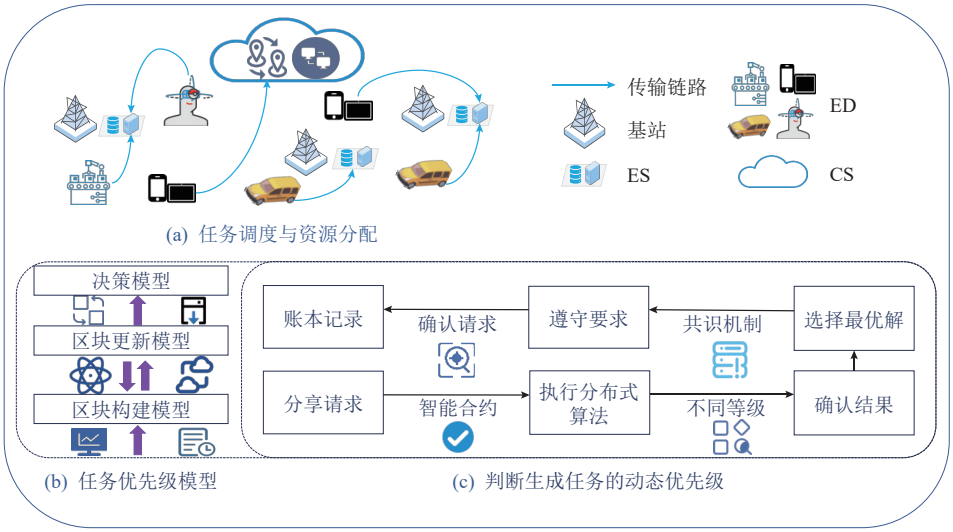


Fig. 2 System model diagram

图 2 系统模型图

其中B_w为带宽，g_{ij}为信道增益，l_n为 ES 间距离，δ为路径损耗指数，P^u为上行发射功率，N_0为噪声功率，I_n^u为上行干扰。对应的数据传输时延为

$$T_u^{trans}=\frac{D_i}{r_{ij}^u}。\tag{2}$$

下行速率r_{ij}^d及下行传输时延T_u^{trans}分别为

$$r_{ij}^d=B_w\mathrm{lb}\left(1+\frac{P^d|g_{ij}|^2l_n^{-\delta}}{N_0+I_n^d}\right),\tag{3}$$

$$T_d^{trans}=\frac{D_i}{r_{ij}^d}。\tag{4}$$

上述模型全面刻画了 ES 间的无线传输特性，为任务调度中的动态资源分配和时延分析提供了基础。

2.2 任务优先级模型

本文将任务定义为四元组$Q_i=(C_i,S_i,T_i,U_i)$，其中C_i为任务所需 CPU 周期数，S_i为数据量，T_i为最大可容忍时延，U_i表示任务紧急程度。任务按照泊松分布生成，各任务按到达时延排序，确保先到先处理。任务的等待时延T_i^{wait}为

$$T_i^{wait}=\max\left\{T_{i^*}^{pri},T_i^{arr}\right\}-T_i^{arr},\tag{5}$$

其中$T_{i^*}^{pri}$为主任务完成时延，T_i^{arr}为任务 i 的到达时刻。为确定任务优先级，采用层次分析法，构建目标层、准则层和方案层的层次结构，根据数据量、数据大小、最大容忍时延和紧急度等因素综合计算优先级。每个子任务优先级定义为$\boldsymbol{L}_i\doteq\boldsymbol{\Delta}\times\nabla$，其中$\boldsymbol{\Delta}$为准则层矩阵，$\nabla$为紧急度矩阵。具体建模方法见 3.1 节。

2.3 区块构建模型

由于节点间缺乏互信且不愿共享数据，可利用区块链中各区块提升数据隐私性。具体而言，区块链通过账本记录各节点数据，由 ES 或 CS 统一管理和控制。多个 ES 与 CS 共同作为区块链节点，保障区块链正常运行。当某节点状态发生变化时，账本信息实时更新，确保记录不可篡改。考虑到区块链中任务反馈数据有限，忽略验证开销。隐私开销主要包括区块构建和区块更新 2 个部分。区块构建的隐私开销T_s^{mg}定义为

$$T_s^{mg}=\frac{B}{f_{ij}^{ES}},\tag{6}$$

其中，B为区块大小，f_{ij}^{ES}为 ES 的计算资源分配。随后，各区块节点将交易打包并下载新区块。区块更新的隐私开销T_s^{md}定义为

$$T_s^{md}=\mu\mathrm{lb}\mathcal{M}\frac{B}{r_s^d},\tag{7}$$

其中，μ为传输参数，r_s^d为 ES 到相邻 ES 的下行速率，$\mathcal{M}$为终端设备数量，传播路径随设备数$\mathcal{M}$的增加呈对数增长。则每个 ES 的总隐私开销为

$$T_s=T_s^{mg}+T_s^{md}=\max_i\left\{\frac{B}{f_{ij}^{ES}}\right\}+\mu\mathrm{lb}\mathcal{M}\frac{B}{r_s^d}。\tag{8}$$

2.4 决策模型

决策模型包括本地任务处理与调度决策，即任务可选择在本地计算或调度至 ES 或 CS 处理。

1）本地计算

若 ED 生成的任务在本地处理，设其可用计算资源为F_i^{ED}，分配给任务的资源为f_i^{ED}，$0\leqslant f_i^{ED}\leqslant F_i^{ED}$定义为约束C1，则本地计算时延为

$$T_i^{ED_{total}}=\frac{C_i}{f_i^{ED}}。\tag{9}$$

2）ES 调度

若任务调度至ES，分配资源为f_{ij}^{ES}，$0\leqslant f_{ij}^{ES}\leqslant F_i^{ES}$

定义为约束C2, $T_{\mathrm{u}}^{\mathrm{trans}}$和$T_{\mathrm{d}}^{\mathrm{trans}}$分别为上行与传输时延下行传输时延。则总时延为

$$T_{ij}^{\mathrm{ES_{total}}}=\frac{C_i}{f_{ij}^{\mathrm{ES}}}+T_{\mathrm{u}}^{\mathrm{trans}}+T_{\mathrm{d}}^{\mathrm{trans}}\ 。\tag{10}$$

3)CS 调度

若任务调度至CS, 分配资源为f_i^{CS}, $0\leqslant f_i^{\mathrm{CS}}\leqslant F_i^{\mathrm{CS}}$定义为约束C3, 则总时延为

$$T_i^{\mathrm{CS_{total}}}=\frac{C_i}{f_i^{\mathrm{CS}}}+T_{\mathrm{u}}^{\mathrm{trans}}+T_{\mathrm{d}}^{\mathrm{trans}}\ 。\tag{11}$$

综上, 任务调度决策需综合考虑本地计算与调度至ES或CS的资源分配与时延约束, 以优化整体性能。

2.5 研究问题描述

在该融合网络中, 随着任务数量激增、任务复杂性增强、ES覆盖范围动态变化及网络通信不稳定, 降低整体网络开销变得极具挑战。本文旨在最小化整体任务处理时延与隐私开销, 并保障区块生成速率与任务调度过程相匹配。由于ES具备动态通信与计算能力, 核心问题在于如何为多终端合理分配调度决策与计算资源。为此, 目标函数设计如下:

$$\mathcal{P}1:\ \min_{(a_{ij},b_i,c_i,f_{ij},B)}\ T\left(a_{ij},b_i,c_i,f_{ij}^{\mathrm{ES}},f_i^{\mathrm{ED}},f_i^{\mathrm{CS}},B\right),\tag{12}$$

$$\text{s.t. C1,C2,C3},$$

$$\sum_{i\in U}\sum_{j\in E}f_{ij}^{\mathrm{ES}}\leqslant F_{ij}^{\mathrm{ES_{total}}},\tag{12a}$$

$$\sum_{i\in U}b_i f_i^{\mathrm{ED}}\leqslant F_i^{\mathrm{ED_{total}}},\tag{12b}$$

$$\sum_{i\in U}c_i f_i^{\mathrm{CS}}\leqslant F_i^{\mathrm{CS_{total}}},\tag{12c}$$

$$a_{ij}T_{ij}^{\mathrm{ES_{total}}}\leqslant T_i,\ \forall i\in U,\forall j\in E,\tag{12d}$$

$$b_i T_i^{\mathrm{ED_{total}}}\leqslant T_i,\ \forall i\in U,\tag{12e}$$

$$c_i T_i^{\mathrm{CS_{total}}}\leqslant T_i,\ \forall i\in U,\tag{12f}$$

$$\sum_{i\in U}\sum_{j\in E}(a_{ij}+b_i+c_i)=1,\tag{12g}$$

$$a_{ij},b_i,c_i\in\{0,1\},\ \forall i\in U,j\in E,\tag{12h}$$

$$S_{\mathrm{block}}^{\min}\leqslant B\leqslant S_{\mathrm{block}}^{\max}\ 。\tag{12i}$$

综合考虑ECEC网络下的大任务调度、资源分配及隐私开销问题。其中C1, C2, C3及式(12a)~(12i)为约束条件。二元变量a_{ij}, b_i, c_i分别表示任务调度至ES、本地执行及调度至CS的决策。约束式(12a)~(12c)保证ES, ED, CS的计算资源不超过其最大值($F_{ij}^{\mathrm{ES_{total}}}$, $F_i^{\mathrm{ED_{total}}}$, $F_i^{\mathrm{CS_{total}}}$为总计算资源); 式(12d)~(12f)确保任务总时延不超过最大容忍时延; 式(12h)限定每个任务只能调度至1个ES或CS, 保证任务完整性; 式(12i)则规定ES区块的数据容量范围。

由于该优化问题$\mathcal{P}1$属于混合整数非线性规划, 目前尚无多项式时间算法能有效解决其指数级复杂性。为此, 本文提出BBLM-DIGA方案, 基于动态优先级, 利用区块链建立完善的安全与隐私机制, 并结合改进型遗传算法与拉格朗日乘子法, 分别解决调度决策、资源分配与区块分配等难题。

3 基于区块链的动态优先级调度策略

3.1 动态优先级分配策略设计

本节采用层次分析法(AHP)动态分配任务权重。该方法包括AHP中的目标层、准则层和方案层, 分别对应任务i的优先级、影响因素及决策目标。

首先, 构建判断矩阵$\boldsymbol{J}=(j_{mn})$, 其定义如下:

$$j_{mn}=\begin{cases}\dfrac{1}{j_{nm}}=k,k=1,2,\cdots,9, & \text{if } m\neq n,\\ 1, & \text{if } m=n\ 。\end{cases}\tag{13}$$

j_{nm}表示准则m相对于n的重要性。当$m=n$时, $j_{mn}=1$, 否则矩阵对称; 若m比n重要k倍, 则n相对于m为$1/k$。

通过特征值法计算判断矩阵的相对权重矩阵$\boldsymbol{\varpi}=(\varpi_1,\varpi_2,\cdots,\varpi_i)^T$, 满足如下:

$$\boldsymbol{J}\boldsymbol{\varpi}^*=\lambda_{\max}\boldsymbol{\varpi},\tag{14}$$

其中$\boldsymbol{\varpi}^*$为特征向量, $\lambda_{\max}$为最大特征值, 权重通过归一化平均法得到。

准则层的紧急度矩阵∇定义为

$$\nabla=(\boldsymbol{\varpi}_C,\boldsymbol{\varpi}_S,\boldsymbol{\varpi}_T,\boldsymbol{\varpi}_U)\ 。\tag{15}$$

$\boldsymbol{\varpi}_C$, $\boldsymbol{\varpi}_S$, $\boldsymbol{\varpi}_T$, $\boldsymbol{\varpi}_U$分别表示任务的计算量、数据量、最大容忍时延和紧急度权重。各任务的准则层矩阵$\boldsymbol{\Delta}$为

$$\boldsymbol{\Delta}=\begin{pmatrix}\varpi_1^C & \varpi_1^S & \varpi_1^T & \varpi_1^U\\ \varpi_2^C & \varpi_2^S & \varpi_2^T & \varpi_2^U\\ \vdots & \vdots & \vdots & \vdots\\ \varpi_m^C & \varpi_m^S & \varpi_m^T & \varpi_m^U\end{pmatrix}\ 。\tag{16}$$

一致性检验用于提高判定合理性, 一致性比率的定义为:

$$CR=\frac{\lambda_{\max}-p}{p-1},\tag{17}$$

若$CR<0.1$则通过检验。最终, 任务优先级权重向量$\boldsymbol{L}_m$计算为:

$$\boldsymbol{L}_m=\boldsymbol{\Delta}\times\nabla=(\boldsymbol{L}_1,\boldsymbol{L}_2,\cdots,\boldsymbol{L}_k)^{\mathrm{T}}\ 。\tag{18}$$

AHP的主要流程通过构建判断矩阵、求解特征

向量、构建准则层矩阵、进行一致性检验，最终得到终端任务的优先级权重。

3.2 区块构建和更新方法设计

在 ECEC 网络架构中，区块链由固定的 CS 或 ES 维护，任务调度相关信息也由 ES 记录。所有交易通过共识机制验证后存储在持久、不可篡改的账本中，确保数据安全可追溯，提升 ES 与 CS 的安全性和可靠性。

ES 在分配任务时，先收集通信范围内的任务信息，并通过分布式决策算法及账本查询，寻找相邻 ES 或 CS 以做进一步调度，如图 3 所示，具体流程如下：

1）每个终端生成任务交易并向当前 ES 发送请求。区块链引入后，通过预处理方法动态分配区块大小，通过工作量证明机制进行区块挖矿，确保区块生成满足设定难度。

2）ES 中的智能合约处理调度请求并触发分布式决策算法，选定最优调度目标。

3）ES 对执行结果进行回溯验证，若结果有效且调度时延不超过最大容忍时延 T_i，则选取总时延最短的 ES 作为目标并生成交易。

4）区块链通过校验每个区块哈希值验证交易有效性，保证链完整性。最优解存储于区块中，任何篡改都会导致后续区块哈希值变化，确保数据不可伪造。

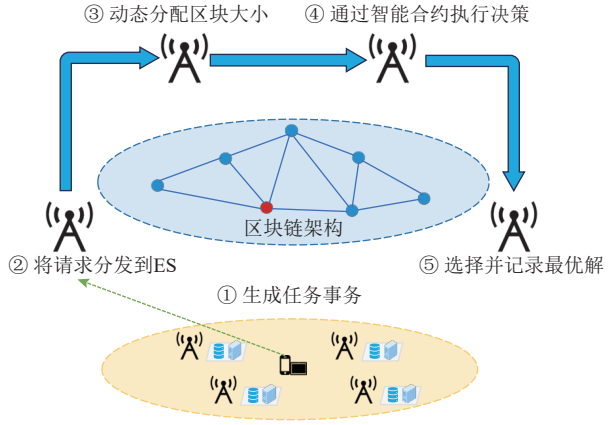


Fig. 3 Methods for building and updating blockchain

图 3 区块链构建和更新方法

该机制实现了任务调度全流程的安全性、可追溯性和链上数据的一致性。

3.3 拉格朗日问题转化

本节将 $\mathcal{P}1$ 由非凸问题转化为凸优化问题，证明在特定方案下计算资源分配的凸性，并利用拉格朗日乘子法求解最优资源分配。随后，将调度方案编码，并采用分布式启发式算法获得近优解，从而提升 ECEC 网络的整体安全性与效率。

首先，针对 $\mathcal{P}1$ 中的离散变量 a_{ij}, b_i, c_i 与连续资源分配变量 f，对原问题进行分解与解耦，将其划分为资源分配与调度决策 2 个子问题。现对如何将原问题转化为资源分配子问题加以说明。

在此，假定二元决策变量 a_{ij}, b_i, c_i 已给定，区块大小 B 已进行预分配，则 $\mathcal{P}1$ 可分解为 $\mathcal{P}2$：

$$\mathcal{P}2:\ \min_{(f_{ij}^{\mathrm{ES}},f_i^{\mathrm{EU}},f_i^{\mathrm{CS}})} T\left(\tilde{a}_{ij},\tilde{b}_i,\tilde{c}_i,f_{ij}^{\mathrm{ES}},f_i^{\mathrm{ED}},f_i^{\mathrm{CS}},\tilde{B}\right), \tag{19}$$

$$\text{s.t. C1,C2,C3}, \tag{19a}$$

$$\sum_{i\in U}\sum_{j\in E}\tilde{a}_{ij}f_{ij}^{\mathrm{ES}} \leqslant F_{ij}^{\mathrm{ES}_{\mathrm{total}}}, \tag{19b}$$

$$\sum_{i\in U}\tilde{b}_i f_i^{\mathrm{EU}} \leqslant F_i^{\mathrm{ED}_{\mathrm{total}}}, \tag{19c}$$

$$\sum_{i\in U}\tilde{c}_i f_i^{\mathrm{CS}} \leqslant F_i^{\mathrm{CS}_{\mathrm{total}}}。 \tag{19d}$$

我们将 $\mathcal{P}1$ 转化为子问题 $\mathcal{P}2$。在 $\mathcal{P}2$ 中，一旦去除耦合变量 a_{ij}, b_i, c_i，任务的资源分配问题即可独立拆分。因此，$\mathcal{P}2$ 可进一步分解为 $N+2$ 个子问题：包括 N 个 ES 资源分配子问题、1 个 CS 资源分配子问题及 1 个 EU 资源分配子问题。其拉格朗日构建方程为：

$$\begin{aligned}L\left(f_{ij}^{\mathrm{ES}},f_i^{\mathrm{EU}},f_i^{\mathrm{CS}},\lambda_j,\psi,\Pi\right)=&\sum_{i\in U}\sum_{j\in E}\left(\hat{a}_{ij}T_{ij}^{\mathrm{ES}_{\mathrm{total}}}+\hat{b}_i T_i^{\mathrm{ED}_{\mathrm{total}}}+\right.\\&\left.\hat{c}_i T_i^{\mathrm{CS}_{\mathrm{total}}}+T_s^{\mathrm{md}}\right)-\sum_{j\in E}\lambda_j\left(\sum_{i\in U}\hat{a}_{ij}f_{ij}^{\mathrm{ES}}-F_{ij}^{\mathrm{ED}_{\mathrm{total}}}\right)-\\&\psi\left(\sum_{i\in U}\hat{b}_i f_i^{\mathrm{ED}}-F_i^{\mathrm{ED}_{\mathrm{total}}}\right)-\Pi\left(\sum_{i\in U}\hat{c}_i f_i^{\mathrm{CS}}-F_i^{\mathrm{CS}_{\mathrm{total}}}\right),\end{aligned} \tag{20}$$

其中 λ_j, ψ, Π 为拉格朗日乘子。对拉格朗日函数分别求偏导，得到的最优解 $f_{ij}^{\mathrm{ES}^*}$, $f_i^{\mathrm{EU}^*}$, $f_i^{\mathrm{CS}^*}$ 为：

$$f_{ij}^{\mathrm{ES}^*}=\sum_{\substack{m\in U,\\ m\neq i}}\frac{F_{ij}^{\mathrm{ES}_{\mathrm{total}}}}{\sqrt{\hat{a}_{mj}C_m+B(n)}}, \tag{21}$$

$$f_i^{\mathrm{ED}^*}=\sum_{\substack{m\in U,\\ m\neq i}}\frac{F_i^{\mathrm{ED}_{\mathrm{total}}}}{\sqrt{\hat{b}_m C_m}},\quad f_i^{\mathrm{CS}^*}=\sum_{\substack{m\in U,\\ m\neq i}}\frac{F_i^{\mathrm{CS}_{\mathrm{total}}}}{\sqrt{\hat{c}_m C_m}}。 \tag{22}$$

3.4 BBLM-DIGA 策略实现

本文引入基于区块链的分布式孤岛遗传策略作为任务调度算法，其主要流程如图 4 所示，具体包括：种群初始化、适应度计算、选择、交叉、变异和迁移。

1）种群初始化

通过基于染色体的编码方法实现种群初始化，生成多样化的初始解集，为任务分配建模决策与资源分配提供基础。

2）适应度函数设计

有效的适应度函数可引导染色体进化。资源受限时，可能出现时延约束违规，将约束转化为惩罚项

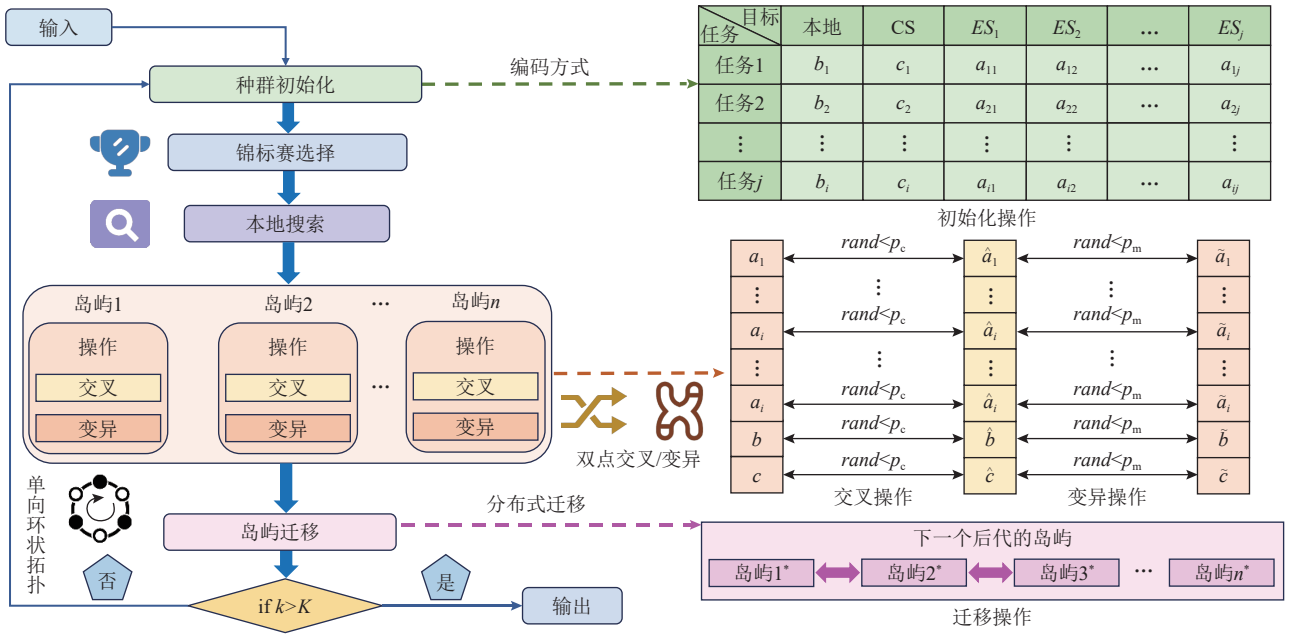


Fig. 4 The algorithm framework of BBLM-DIGA

图 4 BBLM-DIGA 算法框架

Pen，以实现约束的放松并降低系统开销：

$$Pen=\iota\left(\sum_{i\in U}\sum_{j\in E}J\left[a_{ij}T_{ij}^{\mathrm{ES}^*}-T_i^{\max}>0\right]+\sum_{i\in U}J\left[b_iT_i^{\mathrm{ED}^*}-T_i^{\max}>0,c_iT_i^{\mathrm{CS}^*}-T_i^{\max}>0\right]\right),\tag{23}$$

其中ι为惩罚因子，J为指示函数。适应度函数定义为

$$Fit=T\left(a_{ij},b_i,c_i,f_{ij}^{\mathrm{ES}^*},f_i^{\mathrm{ED}^*},f_i^{\mathrm{CS}^*},\tilde{B}\right)+Pen\text{。}\tag{24}$$

3)锦标赛选择

采用锦标赛选择，随机选取Q个个体，保留适应度最优者。锦标赛规模动态调整：

$$\sigma_m=\sigma_{\min}+\left\lceil(\sigma_{\max}-\sigma_{\min})\frac{\mathrm{lb}(1+t)}{\mathrm{lb}(1+T)}\right\rceil,\tag{25}$$

其中$\sigma_{\max}$和$\sigma_{\min}$分别表示锦标赛的最大规模和最小规模，t和T分别表示当前迭代和最大迭代。

4)交叉与变异

采用均匀交叉机制，提升后代多样性。以概率$Prob_{\mathrm{c}}$进行基因交换，概率$Prob_{\mathrm{m}}$实施变异，从而产生更优解。

5)迁移操作

基于消息传递接口的通信协议实现子种群间的信息交换，采用环形拓扑结构，周期性地将优秀个体迁移至相邻子种群，设置迁移频率为F_i。

算法 1 展示了 BBLM-DIGA 的主要流程，包括初始化、交叉、变异、选择与迁移等操作。得益于分布式结构，BBLM-DIGA 特别适用于 ECEC 架构下的大规模任务卸载问题。首先，通过建立多个岛屿，并在不同岛屿上并行运行优化遗传算法，实现种群分割与迁移，从而保持种群多样性并生成表现更优的岛屿。随后，利用精英选择机制，从各岛屿中选取适应度最优的个体，直至找到最优个体Q^*。

在交叉与变异阶段，采用双点交叉和双点变异操作，始终保证约束条件的满足，有效缩小搜索空间，提高算法效率。迁移过程中，通过人工选择算子保留各代进化中的最优个体以形成精英解，这也作为算法收敛的判据。

算法 1. BBLM-DIGA 策略。

输入：初始种群规模 S_{pop}，最大迭代次数 T，交叉概率 $Prob_{\mathrm{cro}}$，变异概率 $Prob_{\mathrm{mut}}$，锦标赛规模 $\sigma_{\min}$和 $\sigma_{\max}$，迁移周期F_i，惩罚因子ι；

输出：任务调度方案O_i，最优资源分配方案f_{ij}。

① 采用染色体编码方式初始化种群K；

② 获取任务优先级L_{mut}；

③ 设全局最优解G_{best}为空；

④ for t = 1 to T

⑤ 计算个体适应度值，包含惩罚项；

⑥ 按适应度降序对种群个体进行排序；

⑦ 动态计算当前锦标赛规模σ_{mut}；

⑧ 采用锦标赛选择保留适应度较优个体；

⑨ 以概率$Prob_{\mathrm{c}}$进行均匀交叉，以概率$Prob_{\mathrm{mut}}$进行基因变异；

⑩ if t mod F_i == 0

⑪ 选取每个子种群的最优个体Q_m^*；

⑫ 将Q_m^*迁移至相邻子种群，并接收来自其他子种群的优秀个体；

⑬ 删除种群K中最差个体，并将迁入个体并入；

⑭ end if

⑮ 更新局部最优解G_{loc}；

⑯ 若G_{loc}优于G_{best}，则更新全局最优G_{best}；

⑰ end for

将对 BBLM-DIGA 方案的时间复杂度进行分析。在 ES 与 CS 之间进行任务选择时，时间复杂度为$O(IJ)$。在初始化过程中，共生成Q个个体。针对资源分配方案在T次迭代过程中，评估每一个个体时，都需要遍历I个任务，并为每个任务在J个计算节点中计算并寻找最优的资源分配方案，其计算的时间复杂度为$O(TQIJ)$。精英选择操作在最坏情况下的时间复杂度为$O(Q^2)$。对于个体列表的排序操作，时间复杂度为$O(TQ\log Q)$。此外，交叉与变异操作的时间复杂度均为$O(TQI)$。因此，省略较小的复杂度项后，整体的时间复杂度可表示为$O(TQIJ)$。

4 实验验证

4.1 实验设置和对比方法

主要实验参数如表 1 所示，实验则采用 Python 3.8 编程，构建了一个动态 ECEC 网络架构，包括一个 CS、若干彼此分散的 ES 以及多个用户，任务生成服从泊松分布。为了确保结果的可靠性和准确性，我们进行了一系列实验。提出的 BBLM-DIGA 方案通过与以下 5 种基线方法进行比较来评估。

1）DIGA[25]。该方法未采用拉格朗日乘子法和区块链进行预处理，因而无法有效分配区块大小，也难以在初始化阶段获得合理的初始解。通过对比可体现 DIGA 对 BBLM-DIGA 收敛速度与性能的显著提升作用。

2）LM-GA[26]。GA 具备较强的局部搜索能力，能处理多种复杂决策问题，在个体多样性与收敛精度方面表现良好。作为 BBLM-DIGA 的初始版本，与其对比有助于突出 BBLM-DIGA 在解质量上的优势。

3）LM-AGWO[27]。AGWO 通过引入自适应机制改进了原始 GWO，在全局与局部搜索间实现平衡。该方法融合数学建模的约束处理方式，并采用三狼群协同优化，在多峰函数求解中表现优异。

4）LM-GRE[28]。贪心方案依据优化目标对各终端用户产生的任务进行卸载。该方法可提升资源利用率和系统性能，但仅聚焦于任务调度的短期效果。

5）LM-DPSO[29]。DPSO 在保留 PSO 全局搜索能力的基础上，调整了粒子表示、速度与位置更新机制。其采用离散变量编码，速度定义为当前解改变的概率，适用于解决离散优化问题。

Table 1 Configuration Parameters of BBLM-DIGA

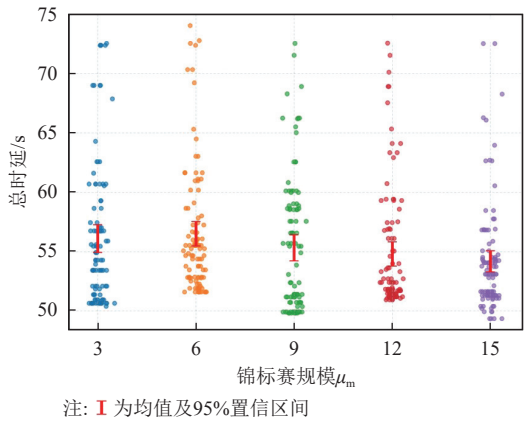
表 1 BBLM-DIGA 配置参数

配置	取值
ES 数量	[20, 80]
任务数量	[50, 300]
任务大小/MB	[100, 1 000]
云服务器数量	1
任务计算量/Gcycle	[0.5, 3]
任务紧迫度	[1, 5]
最大截止时间/s	[1, 5]
最小带宽/MHz	5
信道增益	$[10^{-8}, 10^{-7}]$
噪声/（dBm·Hz^{-1}）	−174
种群规模	[20, 70]
锦标赛规模	[3, 15]

4.2 参数选择分析

图 5 展示了不同锦标赛规模μ_m对 BBLM-DIGA 策略总时延的影响。可以看出，随着μ_m的增大，总时延在一定范围内呈现先下降后上升的趋势。当μ_m=9 时，算法的平均总时延最低，且 95% 置信区间最窄，说明此时的调度性能最优，稳定性也较高。过小或过大的锦标赛规模可能导致搜索能力不足或过早收敛，从而影响全局优化效果。

图 6 评估了种群规模p_k对 BBLM-DIGA 总时延的影响。实验结果表明，随着种群规模p_k的增加，总

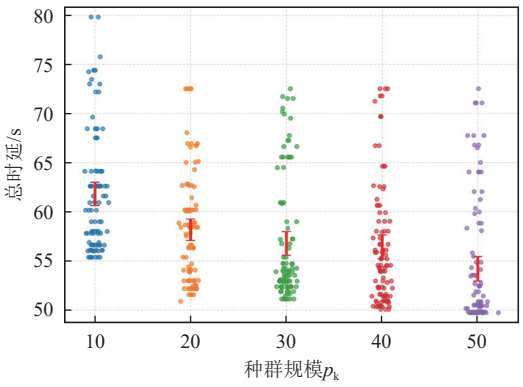


注：I 为均值及95%置信区间

Fig. 5 Influence of the tournament size on the total delay

图 5 锦标赛规模对总时延的影响

时延逐渐下降，并在 $p_k = 40$ 时达到最优，表现出最低的平均值和最小的波动范围。种群过小可能限制解空间的多样性，而过大则增加计算开销。因此，适中规模能在优化精度与计算效率之间取得较好平衡。



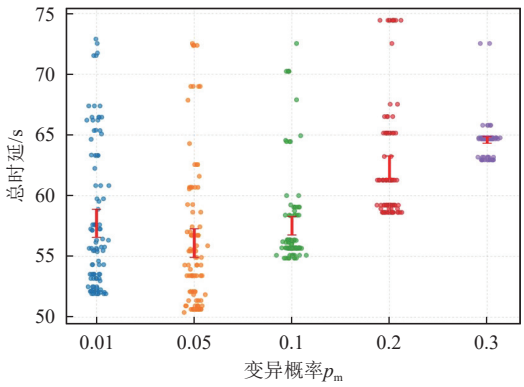
注：I 为均值及95%置信区间

Fig. 6 The influence of the population size on the total delay

图 6 种群规模对总时延的影响

图 7 和图 8 分别展示了变异概率 p_m 与交叉概率 p_c 对 BBLM-DIGA 策略总时延的影响。从图 7 可见，适中的变异概率(如 $p_m = 0.05$)能够有效降低总时延，且具有较小的波动范围，说明此时算法在跳出局部最优和保持解的多样性之间达到了较好平衡。过低的变异概率可能导致搜索陷入局部最优，而过高则增加了解的不确定性。同样地，图 8 显示，交叉概率 p_c 对算法性能也有显著影响，从实验结果来看，当 $p_c = 0.5$ 时，总时延最低且波动范围最小，表明此设置下个体间信息交换最为有效，能够在保持种群多样性和收敛速度之间取得最佳平衡。综上，适当调整变异与交叉概率是提升 BBLM-DIGA 性能的重要因素。

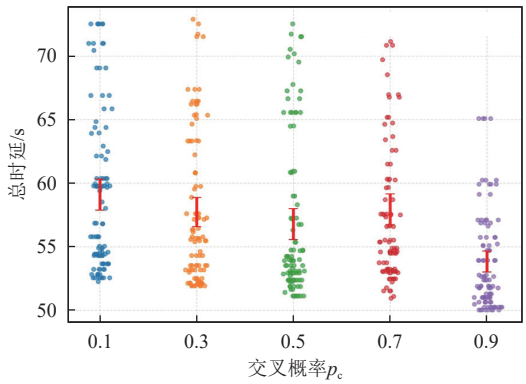
为了量化各因素对任务优先级的影响，本文采用层次分析法确定权重向量 $\nabla = (\varpi_C, \varpi_S, \varpi_T, \varpi_U) =$ (0.15,0.20,0.35,0.30)。结果表明，最大容忍时延是决定任务优先级的关键因素，这源于边缘计算环境中时延敏感性直接影响服务质量；紧急度次之，反映任务重要性等级；数据量主要影响传输时延；计算量权重最低，因 ES 和 CS 的计算资源相对充足。



注：I 为均值及95%置信区间

Fig. 7 The influence of the mutation probability on the total delay

图 7 变异概率对总时延的影响



注：I 为均值及95%置信区间

Fig. 8 The influence of the crossover probability on the total delay

图 8 交叉概率对总时延的影响

表 2 对比了 4 种权重配置策略的系统性能。实验结果显示，当前权重配置实现了 54.3 ms 的总时延和 0.06 的拥塞率(*CR*)，性能最优。均衡配置虽然公平，但总时延增至 58.3 ms，*CR* 值升至 0.08。时延优先策略的性能与当前配置相近，而紧急度优先策略导致时延增至 56.2 ms，*CR* 值为 0.07。这验证了本文权重配置在平衡时延性能与资源利用率方面的合理性。

Table 2 Weight Analysis Based on AHP

表 2 基于 AHP 的权重分析

类型	ϖ_C	ϖ_S	ϖ_T	ϖ_U	总时延/s	*CR*
均衡权重	0.25	0.25	0.25	0.25	58.3	0.08
时延优先	0.15	0.20	0.35	0.30	54.7	0.06
紧急优先	0.15	0.15	0.25	0.45	56.2	0.07

4.3 性能对比分析

本节选取 5 种基线算法，与所提出的 BBLM-DIGA 策略进行性能对比。实验中预处理参数设置为第 3 节中获得的最优值，并在控制变量的前提下进行模拟对比分析。图 9 进一步展示了在任务数据总量相同的前提下，不同任务大小对 6 种算法系统总时延的影响。随着任务数据量增加，传输负担加重，对计算和通信资源的需求也随之增长。结果显示，BBLM-DIGA 依托其独特的分布式架构，能更好地应对多局部最优的问题，在收敛速度和计算时延

方面均优于其他对比算法。

图 10 展示了系统总时延随子任务数量变化的对比结果。可以观察到，随着任务数量的增加，系统总时延显著上升。这主要是由于任务数量增加导致问题规模扩大。在初始规模较小时，各算法表现接近；但随着规模扩大，缺乏拉格朗日和区块链预处理的 DIGA 收敛效果明显变差。而 LM-AGWO 等算法由于处理的是二进制变量问题，易陷入局部最优。相比之下，BBLM-DIGA 更适用于大规模问题，且在性能上具有明显优势。

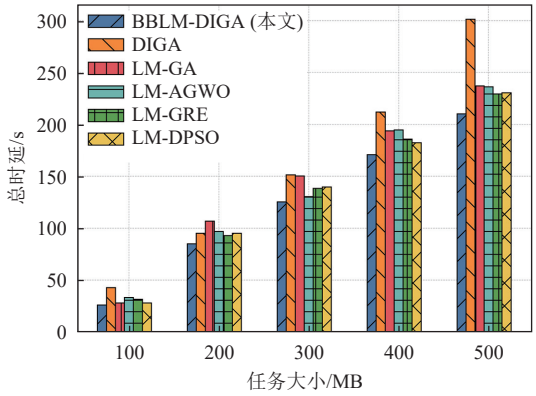


Fig. 9 Performance analysis of comparative algorithms with different sizes of tasks

图 9 不同任务大小的对比算法性能分析

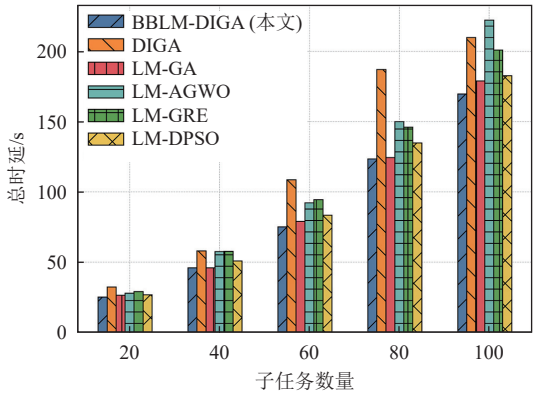


Fig. 10 Performance analysis of comparative algorithms with different number of tasks

图 10 不同任务数量的对比算法性能分析

图 11 分析了任务复杂性对系统总时延的影响。随着任务复杂性的提升，各算法的总时延均有所增加，这主要源于更高的计算资源需求带来的附加计算延迟。BBLM-DIGA 在各复杂性水平下均表现出显著优越性。其预处理方案可有效识别最优区块分配方案，并生成一组优化后的初始解，避免了传统算法陷入局部最优的问题，从而实现更快的收敛。

图 12 对比了在不同 ES 分配的最大计算资源条件下 6 种算法的系统总时延表现。随着 ES 最大计算能力的提升，所有算法的总时延均下降，这是由于更多计算资源被分配至任务，增强了计算性能。然而，在问题复杂度较高的场景下，LM-AGWO 和 LM-GRE 等算法难以有效学习最优策略，容易陷入局部最优。而 BBLM-DIGA 无论在何种复杂度下，始终保持较强的整体性能，展现出良好的适应性和稳定性。

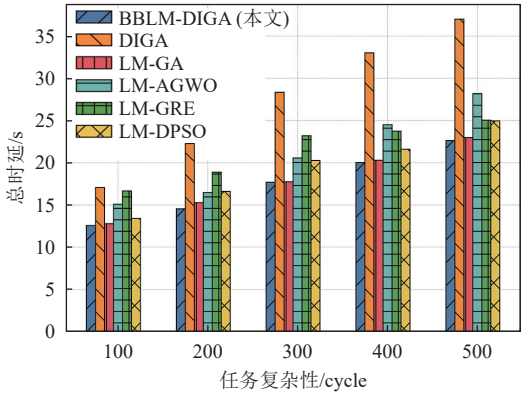


Fig. 11 Performance analysis of comparative algorithms with different complexities of tasks

图 11 不同任务复杂性的对比算法性能分析

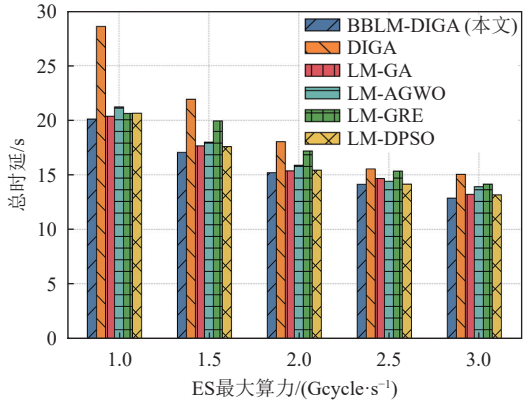


Fig. 12 Performance analysis of comparative algorithms with different computing powers of ES

图 12 不同 ES 最大计算能力的对比算法性能分析

通过对 BBLM-DIGA 策略在收敛性和性能方面的综合分析可以看出，该策略不仅在任务调度决策、计算资源分配和区块链资源调度方面提升了效率，降低了系统时延和隐私开销，构建了不可篡改的区块链结构，从而缓解了隐私泄露风险。与现有方法相比，BBLM-DIGA 的总时延平均降低约 10%，展现出较强的适应性与优化能力。因此，BBLM-DIGA 策略适用于集成 ECEC 的区块链架构中的大规模任务管理场景。

图 13 展示了真实场景的基础设施部署情况，包括 CS、ES 及节点密度分布。表 3 给出了 BBLM-DIGA

在真实场景下的性能分析结果。实验从更大规模的任务数量、任务大小、任务复杂性和 ES 最大算力 4 个维度评估了算法的总时延表现，验证了 BBLM-DIGA 在不同参数配置下的鲁棒性和适应性。

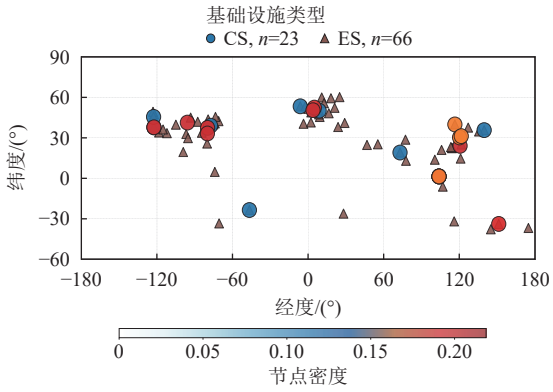


Fig. 13 The distribution range of ES and CS in real scenarios

图 13 真实场景下 ES 和 CS 分布范围

Table 3 Performance Analysis of BBLM-DIGA in Real Scenarios

表 3 真实场景下 BBLM-DIGA 的性能分析

任务数量影响		任务大小影响		任务强度影响		ES 算力影响	
数量	总时延/s	大小/MB	总时延/s	强度/cycle	总时延/s	算力/(cycle·s^{-1})	总时延/s
100	153.09	100	28.95	100	153.09	1 000	184.58
200	716.79	200	99.12	500	158.31	1 500	178.21
300	1 368.41	300	136.73	1 000	164.62	2 000	172.61
400	2 269.74	400	180.24	1 500	170.64	2 500	166.34
500	3 248.16	500	221.39	2 000	174.98	3 000	161.01

4.4 时间复杂度与实时性验证

如表 4 所示，我们首先测试了岛屿数量 N 对实际执行时间的影响。实验结果表明，随着岛屿数量从 1 增加到 10，执行时间从 26.42 s 显著降低至 3.28 s，降幅达 87.6%。同时，标准差从 2.18 s 降至 0.44 s，表明算法稳定性随并行度提升而增强。加速比从 1.00 提升至 8.05，呈现出良好的并行扩展性。

为进一步验证 BBLM-DIGA 的实际性能，我们在图 14 中对比了 4 种算法在不同任务规模下的实际执行时间。虽然 BBLM-DIGA 的理论复杂度为 $O(TKIJ+TK^2)$，但通过分布式并行、自适应迁移等机制，其实际执行时间远低于理论上界。对比实验表明，虽然 LM-GRE 在实际执行时间中相较高于 BBLM-DIGA，但其性能精度远不如 BBLM-DIGA。BBLM-DIGA 在各类任务规模下均表现出优异的时间效率。

Table 4 Influence of the Number of Islands on the Execution Time of BBLM-DIGA

表 4 岛屿数量对 BBLM-DIGA 执行时间的影响

岛屿数量	执行时间/s	标准差/s	加速比
1	26.42	2.18	1.00
3	10.58	0.87	2.50
5	5.31	0.42	4.98
7	4.15	0.51	6.37
10	3.28	0.44	8.05

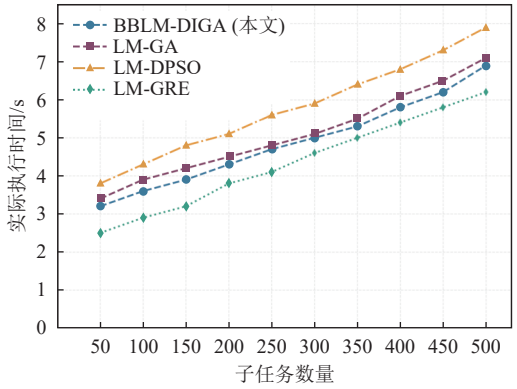


Fig. 14 Analysis of actual execution time for different comparative strategies

图 14 不同的对比策略的实际执行时间分析

4.5 验证隐私开销

区块链的集成不可避免地引入了额外的计算和通信开销，主要源于共识机制执行、链上数据存储和网络同步。为此，BBLM-DIGA 采用边缘层和云层分别维护独立区块链，避免节点在同一链上竞争，降低单链负载；采用任务调度与区块验证的异步执行机制，防止共识过程阻塞实时调度；动态调整区块容量以适应负载变化，避免吞吐量瓶颈。图 15 对比了不同方案在节点数量变化下的隐私开销。结果表明，

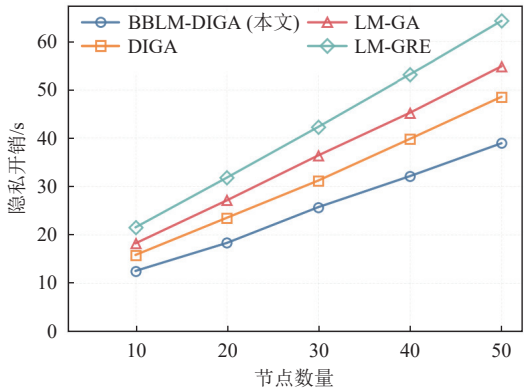


Fig. 15 Analysis of privacy overhead of different strategies with different node numbers

图 15 不同节点数量下不同策略的隐私开销分析

BBLM-DIGA 方案的隐私开销始终最低，相比其他方案，BBLM-DIGA 的隐私验证开销降低了约 40%，验证了双层区块链架构在隐私保护效率方面的显著优势。

图 16 展示了引入区块链框架后适应度函数与总时延之间的关系。总体而言，随着迭代次数的增加，适应度值逐渐降低，表明性能优化稳步提升。在早期阶段，由于区块链同步和共识过程，总时延表现出一定的波动。然而，随着迭代的推进，计算负载趋于均衡，并行优化机制逐渐发挥作用。这表明，在分布式环境中，通过消息传递接口并行执行算法，同时确保全局最优解安全存储在区块链上，可以有效平衡计算开销和优化稳定性，从而实现可靠且高效的优化过程。

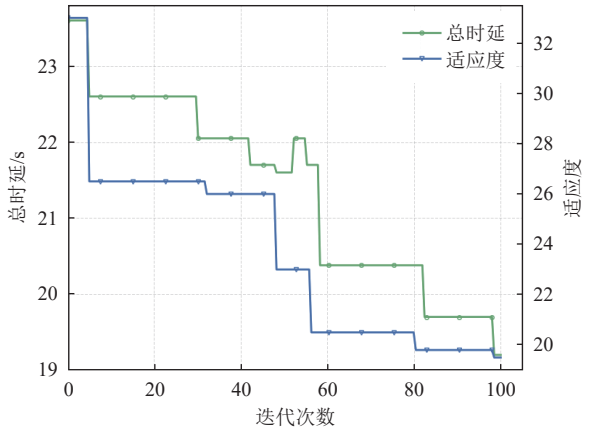


Fig. 16 The convergence analysis on blockchain architecture

图 16 对区块链架构的收敛性分析

5 结 论

本文针对云边端协同计算环境中大规模任务调度与资源分配问题，提出了一种基于区块链辅助的动态优先级调度策略。该策略在充分考虑网络动态性与任务异质性的基础上，通过构建动态优先级模型与分布式优化框架，实现了任务调度的高效性与系统资源的最优利用。具体而言，我们引入层次分析法对任务进行多级优先级划分，并采用基于区块链的拉格朗日乘子分布式岛模型遗传算法，将复杂调度问题转化为可求解的凸优化形式，从而提升算法的收敛速度与解的质量。此外，区块链架构的引入增强了系统对数据隐私与任务完整性的保障能力。大量的实验评估和性能分析表明，所提出的 BBLM-DIGA 策略在系统总延迟和隐私开销方面优于其他卸载方案。

作者贡献声明：李天宇负责方案整体设计、算法思路、实验方法和论文撰写；王兴伟、曾荣飞提出指导意见并修改论文。

参 考 文 献

[1] Duan Sijing, Wang Dan, Ren Ju, et al. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey[J]. IEEE Communications Surveys & Tutorials, 2022, 25(1): 591−624

[2] Li Xinhua, Xu Zhenghui, Fang Fang, et al. Task offloading for deep learning empowered automatic speech analysis in mobile edge-cloud computing networks[J]. IEEE Transactions on Cloud Computing, 2022, 11(2): 1985−1998

[3] Li Tianyu, Wang Xingwei, Zeng Rongfei, et al. Enhancing edge-cloud collaboration with blockchain-assisted digital twin intelligence offloading scheme[J]. IEEE Transactions on Mobile Computing, 2025, 24(10): 9619−9635

[4] Zhao Liang, Han Yingcan, Hawbani A, et al. MEDIA: An incremental DNN based computation offloading for collaborative cloud-edge computing[J]. IEEE Transactions on Network Science and Engineering, 2023, 11(2): 1986−1998

[5] Zhao Liang, Li Tianyu, Zhang Enchao, et al. Adaptive swarm intelligent offloading based on digital twin-assisted prediction in VEC[J]. IEEE Transactions on Mobile Computing, 2023, 23(8): 8158−8174

[6] Liu Haoqiang, Huang Wenzheng, Kim D I, et al. Towards efficient task offloading with dependency guarantees in vehicular edge networks through distributed deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2024, 73(9): 13665−13681

[7] Deng Yiqin, Chen Zhigang, Chen Xiaohao, et al. Task offloading in multi-hop relay-aided multi-access edge computing[J]. IEEE Transactions on Vehicular Technology, 2022, 72(1): 1372−1376

[8] Yao Su, Wang Mu, Qu Qiang, et al. Blockchain-empowered collaborative task offloading for cloud-edge-device computing[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(12): 3485−3500

[9] Xu Shilin, Guo Caili, Hu R Q, et al. Blockchain-inspired secure computation offloading in a vehicular cloud network[J]. IEEE Internet of Things Journal, 2021, 9(16): 14723−14740

[10] Samy A, Elgendy I A, Yu Haining, et al. Secure task offloading in blockchain-enabled mobile edge computing with deep reinforcement learning[J]. IEEE Transactions on network and service management, 2022, 19(4): 4872−4887

[11] Chen Haowei, Deng Shuiguang, Zhu Hongze, et al. Mobility-aware offloading and resource allocation for distributed services collaboration[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(10): 2428−2443

[12] Zhang Jiangjiang, Ning Zhenhu, Ali R H, et al. A many-objective ensemble optimization algorithm for the edge cloud resource scheduling problem[J]. IEEE Transactions on Mobile Computing, 2023, 23(2): 1330−1346

[13] Guo Yongan, Ma Derui, She Hao, et al. Deep deterministic policy gradient-based intelligent task offloading for vehicular computing with priority experience playback[J]. IEEE Transactions on Vehicular Technology, 2024, 73(7): 10655−10667

[14] Xu Chi, Zhang Peifeng, Yu Haibin, et al. D3QN-based multi-priority computation offloading for time-sensitive and interference-limited industrial wireless networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73(9): 13682−13693

[15] Qiao Nan, Yue Sheng, Zhang Yongmin, et al. PoPeC: PAoI-centric task offloading with priority over unreliable channels[J]. IEEE/ACM Transactions on Networking, 2024, 32(3): 2376−2390

[16] Hao Hao, Xu Changqiao, Zhang Wei, 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

[17] Xu Lei, Xu Qimin, Chen Cailian, et al. Efficient task-network scheduling with task conflict metric in time-sensitive networking[J]. IEEE Transactions on Industrial Informatics, 2023, 20(2): 1528−1538

[18] Zhou Jian, Yang Qi Zhao Lu, et al. Mobility-aware computation offloading in satellite edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(10): 9135−9149

[19] Su Qian, Zhang Qinghui, Li Weidong, et al. Primal-dual-based computation offloading method for energy-aware cloud-edge collaboration[J]. IEEE Transactions on Mobile Computing, 2023, 23(2): 1534−1549

[20] Chen Zheyi, Yu Zhengxin. Intelligent offloading in blockchain-based mobile crowdsensing using deep reinforcement learning[J]. IEEE Communications Magazine, 2023, 61(6): 118−123

[21] Nguyen P X, Tran D H, Onireti O, et al. Backscatter-assisted data offloading in OFDMA-based wireless-powered mobile edge computing for IoT networks[J]. IEEE Internet of Things Journal, 2021, 8(11): 9233−9243

[22] Zhai Longzhen, Feng Shaohong. A novel evacuation path planning method based on improved genetic algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2022, 42(3): 1813−1823

[23] Ju Ying, Chen Yuchao, Cao Zhiwei, et al. Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(5): 5555−5569

[24] Lang P, Tian D, Duan X, et al. Blockchain-based cooperative computation offloading and secure handover in vehicular edge computing networks[J]. IEEE Transactions on Intelligent Vehicles, 2023, 8(7): 3839−3853

[25] Li Tianyu, Wang Xingwei, He Qiang, et al. Enhancing SLA-DTNA for intelligence resource reservation in edge-cloud-end collaborative[J]. IEEE Transactions on Services Computing, 2025, 18(6): 3750−3765

[26] Li Tianyu, Zhao Liang, Mao Chaojin, et al. A genetic algorithm based adaptive offloading scheme for domain generalization in VEC[C]// Proc of the 19th Int Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). Piscataway, NJ: IEEE, 2023: 161−168

[27] Cong Yuliang, Xue Ke, Wang Cong, et al. Latency-energy joint optimization for task offloading and resource allocation in MEC-assisted vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2023, 72(12): 16369−16381

[28] Chai Furong, Zhang Qi, Yao Haipeng, et al. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT[J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7783−7795

[29] Chai Zhengyi, Zhao Yingjie, Li Yalun. Multitask computation offloading based on evolutionary multiobjective optimization in industrial Internet of things[J]. IEEE Internet of Things Journal, 2024, 11(9): 15894−15908



Li Tianyu, born in 1998. PhD candidate. His main research interests include task scheduling, resource allocation, and digital twin.

李天宇, 1998 年生。博士研究生。主要研究方向为任务调度，资源编排、数字孪生。



Wang Xingwei, born in 1968. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include cloud computing and next generation Internet.

王兴伟, 1968 年生。博士，教授，博士生导师。CCF 高级会员。主要研究方向为云计算、新一代互联网。



Zeng Rongfei, born in 1983. PhD, professor. His main research interest includes the security and economics of networks.

曾荣飞, 1983 年生。博士，教授。主要研究方向为网络安全和节能。