

基于深度强化学习的多租户算网资源分配算法

胡宇翔^{1,2}, 冯旭¹, 董永吉^{1,2}, 和孟佯^{3,4}, 庄雷⁵, 宋艳蕊⁵

- (1. 信息工程大学信息技术研究所, 河南 郑州 450002; 2. 先进通信全国重点实验室, 河南 郑州 450002;
3. 郑州大学网络空间安全学院, 河南 郑州 450002; 4. 嵩山实验室, 河南 郑州 450002;
5. 郑州大学计算机与人工智能学院, 河南 郑州 450001)

摘要: 随着智能化业务的迅猛发展, 传统网络架构与计算能力之间的既有关系已难以满足当前需求, 算网融合的实施势在必行。在算网融合所催生的新型算力网络框架下, 高效且智能的资源调度策略成为提升用户体验的关键环节, 但现有的资源调度算法优化目标单一, 无法满足多租户差异化的业务需求。为此, 提出了一种基于深度强化学习的多目标资源调度 (MODRLRS, Multi objective deep reinforcement learning resource scheduling) 算法来调用网络中的计算资源和网络资源, 该算法通过构建帕累托最优解集的方法对算网资源进行多目标调度优化以满足不同租户的个性化业务需求。仿真对比实验表明, 相比其他多目标资源调度算法, 新算法提升了 4.9% 的请求接受率和 4.78% 的符合时延请求率, 能够灵活适应各种计算业务的独特需求。

关键词: 算网融合; 算力网络; 资源调度; 多目标优化; 深度强化学习

中图分类号: TN915.08

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2024.00446

Multi-tenant computing network resource allocation algorithm based on deep reinforcement learning

HU Yuxiang^{1,2}, FENG Xu¹, DONG Yongji^{1,2}, HE Mengyang^{3,4}, ZHUANG Lei⁵, SONG Yanrui⁵

1. Institute of Information Technology, Information Engineering University, Zhengzhou 450002, China
2. National Key Laboratory of Advanced Communication Networks, Zhengzhou 450002, China
3. School of Cyberspace Security, Zhengzhou University, Zhengzhou 450002, China
4. Song Shan Laboratory, Zhengzhou 450002, China
5. School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

Abstract: With the rapid advancement of intelligent businesses, the pre-existing relationship between traditional network architectures and computing capabilities has made it difficult to meet the current demands, making the implementation of computing-network convergence inevitable. Under the new computing power network framework brought about by the convergence of computing networks, efficient and intelligent resource scheduling strategy has become a key link to improve user experience. However, the existing resource scheduling algorithms have a single optimization objective and cannot meet the differentiated business needs of multi-tenants. To this end, a Multi objective deep reinforcement learning resource scheduling (MODRLRS) was proposed to call the computing resources and network resources in the computing power network. The algorithm performs multi-objective scheduling optimization of computing network resources by con-

收稿日期: 2024-10-21; 修回日期: 2024-12-11

通信作者: 冯旭, uxengf@126.com

基金项目: 国家重点研发计划 (No. 2023YFB2903902); 中原科技创新领军人才项目 (No. 244200510038); 河南省科技攻关-嵩山实验室资助项目 (No. 232102210154)

Foundation Items: The National Key Research and Development Program of China (No. 2023YFB2903902), The Science and Technology Innovation Leading Talents Subsidy Project of Central Plains (No. 244200510038), The Scientific and Technological Project in Henan Province-Pre-Research Project of Songshan Laboratory (No. 232102210154).

structing a Pareto optimal solution set to meet the personalized business needs of different tenants. Simulation experimental results show that compared with other multi-objective resource scheduling algorithms, the proposed algorithm improves the request acceptance rate by 4.9% and the compliant delay request rate by 4.78%, which can flexibly adapt to the unique requirements of various computing services.

Key words: integration of computing and networking, computing power network, resource scheduling, multi objective optimization, deep reinforcement learning

0 引言

随着数字化与智能化应用的持续演进，对算力资源和网络资源的需求呈现全新特征^[1]。一方面，由于算力应用场景的复杂性与多样性日益增强，传统的单一算力模式已难以满足日益增长的业务需求，需要实现异构算力的融合^[2-3]。另一方面，不同租户的业务活动所产生的庞大数据量，对网络的时延、稳定性以及可靠性提出了更严苛的要求，需要实现数据的快速且可靠传输。算网融合将计算资源和网络资源深度融合构建出一张具备卓越性能的算力承载网络，不仅是对当前算力挑战的有效应对，更是迈向未来智能时代不可或缺的必然选择^[4-6]。

我国运营商于2019年首次提出了算力网络的概念^[7]。算力网络，作为一个集成通信、计算与存储功能于一体的综合性信息系统，已经超越了传统通信网络的范畴^[8]。它凭借高速、移动且无处不在的网络连接能力，有效地整合了云端、边缘端及终端等多层次的算力资源。这一网络能够为用户提供从数据感知、传输、存储到运算的全链条一体化服务，不仅显著提升了算力资源的利用效率，还极大地拓宽了算力技术的应用场景^[9]。

算网融合后的算力网络结构如图1所示。其中，服务提供层负责向租户供给网络服务并满足其需求。服务编排层则专注于对各类网络资源的有效调度与配置。网络控制层实现了算网多维度资源的控制管理，包括灵活调配、性能优化等。算力管理层则致力于解决异构算力资源的建模及交易等问题。算网基础设施层实现了算力和网络的融合，为上层服务提供基础的计算和网络资源。算力网络将用户的计算任务分成若干个按照一条逻辑链路链接的子任务，并根据资源调度策略将这些子任务分配到不同的计算节点上执行，以满足计算任务的性能需求。因此，高效的资源调度算法是提升用户体验的关键。在计算任务中，时延和负载均衡是最常被

考虑的两个性能指标。低时延意味着计算任务能够更快地完成，从而提高用户体验。对于工业自动化、车联网等时间敏感型任务而言，时延过高可能导致任务失败，出现安全事故。而负载均衡能够降低单个节点的压力，提高计算任务请求的响应速度。

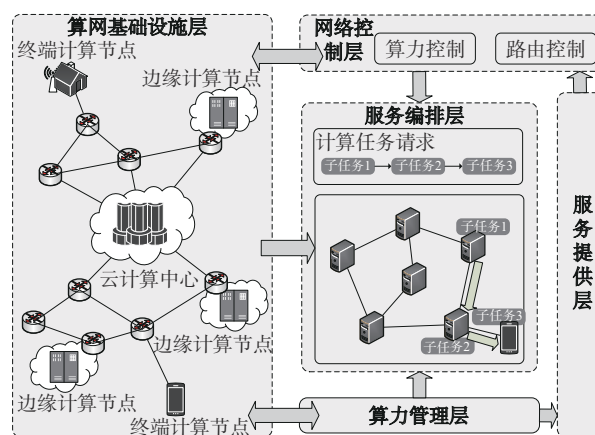


图1 算网融合后的算力网络结构

不同租户的计算任务有着差异化的性能要求，为这些计算任务精准调配网络资源，以满足其独特的性能需求，构成了算力网络面临的一项艰巨挑战^[10]。一方面，算力网络要应对不同租户种类繁多、特性各异的计算任务，这些计算任务各自拥有独特的属性与资源需求^[11-12]。然而基础设施中的计算和网络资源是有限的^[13]，不能统一按照最高的性能标准来为不同计算任务分配资源，这就要求调度算法必须深入挖掘任务特征，精准地将具体的工作负载与适配的资源相匹配，这极大地提升了调度算法的复杂程度。另一方面，算力网络中的资源调度工作往往牵涉多个优化目标^[14-15]，如任务成功率、时延、能耗及负载均衡等^[16]。这些目标之间可能存在着潜在的冲突与矛盾，例如，追求更低的时延可能会导致任务过度集中于某一特定的边缘节点，而负载均衡则要求各计算节点的负载尽可能达到均衡状态。因此，多租户资源调度算法需要在这些相互交织的目标之间巧妙权衡，力求实现整体性能的最优化。

算网融合场景中的资源调度问题已经被证实为一个NP难问题^[17-20]，研究者们提出的解决方案主要有两类，一类是启发式算法^[21-22]，这类算法虽然能够在合理的执行时间内提供一个可行的解决方案，但不能保证总是找到最优解；另一类是深度强化学习算法^[23-26]，虽然这类算法取得了很好的结果，但其通过一个加权的线性相加的价值函数来处理多目标问题，当优化目标冲突时，其结果会变得很差。

本文针对上述问题设计了算网融合网络中最小化时延和负载均衡的数学模型，并提出一种基于深度强化学习的多目标资源调度（MODRLRS, Multi objective deep reinforcement learning resource scheduling）算法，重点优化时延和负载均衡以满足不同计算任务的特定需求，为多租户提供端到端的个性化计算服务。

1 算网融合场景中资源调度问题建模

为了更深入地剖析算网融合场景下的资源调度问题，本文致力于构建一个旨在实现最小化时延与负载均衡的数学模型。

算网融合网络的基础设施架构主要由计算节点与连接这些节点的链路两大组件构成，该架构可通过无向图 $G = (V, L)$ 进行数学化表示。其中， V 代表整个网络中计算节点的集合，可用 $V = \{v_1, v_2, \dots, v_n\}$ 表示， L 代表网络中链路的集合，两个相邻节点的链路 $l \in L$ 。将计算节点的总计算资源数量用参数 $R_{v_n}^{\text{cpu}}$ 表示。每个计算节点剩余可利用的计算资源数量使用集合 $E^{\text{cpu}} = \{e_{v_1}^{\text{cpu}}, e_{v_2}^{\text{cpu}}, \dots, e_{v_n}^{\text{cpu}}\}$ 表示， t^{cpu} 表示进行一单元计算所需时间。对于计算节点间通信链路的定义，本文引入链路集合 L ，其中链路 $l \in L$ 拥有的总带宽资源量由参数 b_l 表示，而数据在该链路上传输的时延则由参数 d_l 表示。

不同租户的业务需求展现出显著的差异性，其中自动驾驶与扩展现实等应用场景对时延有着严苛的要求，它们强调实时性以确保操作的精准和流畅；而大数据处理及大型模型训练等业务则对计算资源提出了极高的需求，这些业务依赖于强大的计算能力来高效处理海量数据及完成复杂计算任务。尽管租户对于计算任务的性能需求多样化，但其计算任务请求都可以形式化为一个五元组 $A = \{a, q_s^{\text{cpu}}, q_s^{\text{bw}}, d^a, a^a\}$ ，其中各元素的具体含义如下： $a = \{s_1, s_2, \dots, s_m\}$ 代表该计算任务请求所包含的若

干个子任务的集合，值得注意的是在处理子任务时需要按任务的逻辑顺序先处理前置的子任务。 $q_a^{\text{cpu}} = \{q_{s_1}^{\text{cpu}}, q_{s_2}^{\text{cpu}}, \dots, q_{s_m}^{\text{cpu}}\}$ 表示计算任务中每个子任务所需计算资源的集合，其中 $q_{s_m}^{\text{cpu}}$ 代表计算任务 A 中第 m 个子任务所需的计算资源数量； $q_a^{\text{bw}} = \{q_{s_1}^{\text{bw}}, q_{s_2}^{\text{bw}}, \dots, q_{s_m}^{\text{bw}}\}$ 表示计算任务中每个子任务所需计算资源的集合，其中 $q_{s_m}^{\text{bw}}$ 代表计算任务 A 中第 m 个子任务所需的带宽资源数量， d^a 代表该计算任务所能容忍的最大时延，即任务完成时间的上限， a^a 则记录了该计算任务请求的到达时间，即任务请求被系统接收的时间点。

决策变量用于表示资源调度算法生成的策略，它代表了计算任务的部署状态以及算力网络中的资源使用情况。本文采用了 3 个主要的决策变量： $x_{v_n}^{s_m}$ 、 y_{v_n} 和 z_l 。其中， $x_{v_n}^{s_m}$ 用于表示计算子任务 s_m 是否被放置在特定的计算节点 v_n 上， y_{v_n} 则反映了计算节点的资源使用情况， z_l 刻画了链路的资源使用情况。 $x_{v_n}^{s_m}$ 、 y_{v_n} 和 z_l 的具体表现形式分别如下

$$x_{v_n}^{s_m} = \begin{cases} 1, s_m \text{ 被部署在节点 } v_n \text{ 上} \\ 0, \text{其他} \end{cases} \quad (1)$$

$$y_{v_n} = \begin{cases} 1, \text{节点 } v_n \text{ 被占用} \\ 0, \text{其他} \end{cases} \quad (2)$$

$$z_l = \begin{cases} 1, \text{链路 } l \text{ 被占用} \\ 0, \text{其他} \end{cases} \quad (3)$$

对于“计算任务 A 所占用的计算资源不能超过计算节点 v_n 可利用计算资源”这一原则，可以得到计算资源有关的约束条件 C1，具体如下

$$\text{C1: } \sum_{s_j \in a} q_{s_j}^{\text{cpu}} \cdot x_{v_n}^{s_j} \leq e_{v_n}^{\text{cpu}} \cdot y_{v_n} \quad (4)$$

在部署过程中，若将被计算任务 A 在数据传输过程中占用的带宽表示为 O_l^{bw} ，将链路可利用的带宽表示为 A_l^{bw} ，则根据占用的带宽不能超过可利用的带宽的原则，可以得到带宽资源的约束条件 C2，具体如下

$$\text{C2: } \sum_{s_j \in a} q_{s_j}^{\text{bw}} \cdot z_l \leq b_l \cdot z_l \quad (5)$$

计算任务 A 的时延是由计算节点处理子任务 s_m 所需要的时间和计算任务 A 产生的数据在传输过程中所需时间两部分组成。处理子任务 s_m 所需要的时延可由第 n 个计算节点处理每单位计算任务所需的时间 t^{cpu} 、第 m 个子任务请求 s_m 的计算资源数量 $q_{s_m}^{\text{cpu}}$ 和决策变量 $x_{v_n}^{s_m}$ 计算得到。数据传输过程中所消耗

的时间可由计算任务 A 在链路 l 上的传播时延 d_l 和链路占用的决策变量 z_l 计算得到。计算任务时延的计算过程如下

$$D^s = \sum_{v \in V} \sum_{s_i \in a} t^{\text{cpu}} \cdot q_{s_i}^{\text{cpu}} \cdot x_v^{s_i} + d_l \cdot z_l \quad (6)$$

方差是衡量计算节点的负载情况最好的参数之一。方差值越小，代表网络中计算节点使用的资源分布越均匀。方差 W^{load} 具体计算过程如下

$$W^{\text{load}} = \frac{\sum_{v \in V} \left(\frac{R_v^{\text{cpu}} - e_v^{\text{cpu}}}{R_v^{\text{cpu}}} - \frac{\sum_{v \in V} R_v^{\text{cpu}} - e_v^{\text{cpu}}}{|N|} \right)^2}{|N|} \quad (7)$$

其中， $|N|$ 表示计算节点数量。

为充分满足算网融合场景中多租户的差异化需求，本文考虑算网资源与计算任务放置之间的约束，并致力于优化计算任务时延和算力网络负载均衡。基于此，本文所构建的数学模型可以表示如下

$$\min (F_1, F_2), \text{ s.t. } C1, C2 \quad (8)$$

2 基于深度强化学习的多目标资源调度优化算法

本节将详细介绍本文所提出的 MODRLRS 算法的核心原理、算法训练过程以及神经网络模型的详细结构。

2.1 MODRLRS 算法原理

MODRLRS 算法的核心是通过构建时延和负载均衡两个优化目标的帕累托最优解来获取最适合租户需求的资源调度方案。MODRLRS 算法构建帕累托最优解集示意图如图 2 所示， x 轴和 y 轴分别表示两个不同的优化目标。在解的可行域内，帕累托前沿点的集合就是该问题的帕累托最优解集，该解集内的每个解都不存在一个优化方向没有变差而使另一指标变得更好的情况。MODRLRS 算法将这一多目标优化问题分解为若干个子问题，再通过对对应训练完成的神经网络模型对每个子问题分别求解，每个子问题优化方向对应一个神经网络模型。当求解子问题的数量足够多时，每个子问题在可行域下的最优解能够近似为一条线，这些最优解的集合构成了 MODRLRS 算法求解的帕累托最优解集^[27]。

MODRLRS 算法在构建帕累托最优解集后需要将这些解运用到实际场景中，考虑的计算任务请求对最大时延有明确要求，超过该时间很可能会出现事故，因此在时延优化方向的优先级应高于负载均衡。

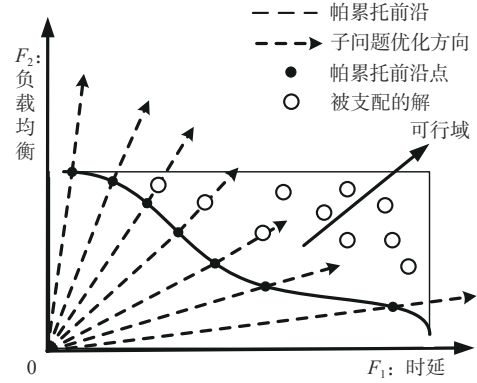


图2 MODRLRS算法构建帕累托最优解集示意图

基于上述原理，MODRLRS 算法的伪代码如下。

算法1 MODRLRS 资源调度算法

输入：状态 S

输出：资源调度方案 P

- 1) for $i=1, 2, \dots$ do
- 2) $p_temp = \pi_{\theta}(a|s)$
- 3) if p_temp 的时延 \leq 最大允许时延 d^s :
- 4) 添加 p_temp 到数组 p_array
- 5) end if
- 6) end for
- 7) $p = \min(\text{Load}(p_array))$
- 8) return p

当有新的计算任务请求到达时，根据步骤 1) 将算力网络中的资源信息作为深度强化学习的策略状态输入并得到多个潜在解；步骤 2) 根据网络现有资源和计算任务请求所需的资源筛选出满足网络资源约束条件的解；步骤 3) 输出符合该计算任务最大时延约束且部署子任务后负载方差最小的解。

2.2 MODRLRS 训练算法

在算法 1 中，策略 $\pi_{\theta}(a|s)$ 代表参数为 θ 的神经网络做出的决策，其含义是在状态为 s 的情况下将当前子任务放置到计算节点 a 上的概率。该策略不能直接使用，需要通过以下反馈不断优化 θ 的值，最终得到不断逼近最优的策略。

$$F = \lambda_1 F_1 + \lambda_2 F_2, \text{ 且 } \lambda_1 + \lambda_2 = 1, \lambda_1, \lambda_2 \in (0, 1) \quad (9)$$

其中， F_1 和 F_2 代表两个优化方向，分别是时延和负载均衡， (λ_1, λ_2) 是两个优化方向的权重，不同的 (λ_1, λ_2) 表示不同的子问题。

从第 2.1 节对 MODRLRS 算法的描述可知，该算法需要训练多个神经网络模型来对相应的多个子问题求解构建帕累托最优解集，如果每个神经网络模

型都从零开始进行训练,那么训练时间过长,会消耗大量计算资源。考虑这个现实问题,本文利用基于邻域参数传递的训练方法来减少计算资源和训练时间的消耗^[24]。MODRLRS算法训练结构如图3所示。在图3中,两个相邻的子问题的权重向量 (λ_1, λ_2) 相邻,表示这两个子问题有着相近最优解,即相邻的两个神经网络模型的参数 θ 相近,所以在训练完第一个神经网络模型后,其他的神经网络模型可以通过其邻近神经网络模型的参数来辅助训练。MODRLR训练算法的伪码如算法2所示,其具体操作为先对第一个神经网络模型进行从零开始的训练,当模型收敛后,将上一个训练好的神经网络模型参数作为当前神经网络模型参数训练的初始值进行再训练。

算法2 MODRLRS训练算法

输入: 初始化 n 个神经网络参数 θ , 训练批次 M , 更新参数步长 N

输出: 优化后的所有网络参数 θ

- 1) **for** $i = 1, 2, \dots, n$ **do**
- 2) **if** $i \neq 1$
- 3) $\theta_i \leftarrow \theta_{i-1}$
- 4) **end if**
- 5) **for** $j = 1, 2, \dots, M$ **do**
- 6) $d_\theta \leftarrow 0$
- 7) $s[i\%N] \leftarrow s$
- 8) $P[i\%N] \leftarrow \pi(a|s, \theta_i)$
- 9) $p[i\%N] \leftarrow \text{Sample}(\pi(a|s, \theta_i))$
- 10) $R[i\%N] \leftarrow R(a|s)$
- 11) **if** $\text{length}(P)\%N = 0$
- 12) $\theta = \theta - \text{lr} \cdot \nabla_{\theta} \bar{R}(\theta)$

- 13) 初始化数组 s, P, p, R
- 14) **end if**
- 15) **end for**
- 16) **end for**

在算法2的单个神经网络模型训练中,最核心的便是步骤12)的参数更新式,以下对其原理和 $\nabla_{\theta} \bar{R}(\theta)$ 的推导进行详细介绍。训练的模型如图3所示,模型分成3个部分:智能体、环境和状态。智能体由求解不同子问题的神经网络组成。环境为算网融合的算力网络系统,可根据智能体的动作反馈相应的惩罚值。状态为各个计算节点和链路的资源情况。当计算任务请求到达时,智能体根据状态选择相应动作,在环境中按照智能体的动作部署后,更新新的状态,并且根据部署结果回馈给智能体惩罚值,引导智能体向好的方向学习。

假设在算网融合场景中各类资源的组合有 n 种情况,那么其状态就有 n 种,可用 $s = \{s_1, s_2, \dots, s_n\}$ 表示,智能体在当前状态下有 m 种不同动作,在本文场景下不同动作代表将一个子任务放置在相应的计算节点,可用 $a = \{a_1, a_2, \dots, a_m\}$ 表示。智能体与环境经过数次交互后得到轨迹 τ ,可表示为 $\tau = \{(s_1, a_1), (s_2, a_2), \dots, (s_t, a_t)\}$ 。在动作轨迹 τ 确定后,可得到一个该动作集的惩罚值

$$R(\tau) = F(\tau) + \lambda_3 \cdot C_1(\tau) + \lambda_4 \cdot C_2(\tau) \quad (10)$$

其中, $F(\tau)$ 为通过式(9)计算出的目标值, $C_i(\tau)$ 为不满足约束情况下的惩罚值。最小化时延和负载均衡方差是本文算法优化的两个目标,因此无论是优化目标或是惩罚都需要往0的方向优化,基于此,本文将优化的目标函数定义为计算任务时延、负载方差与惩罚值的和。

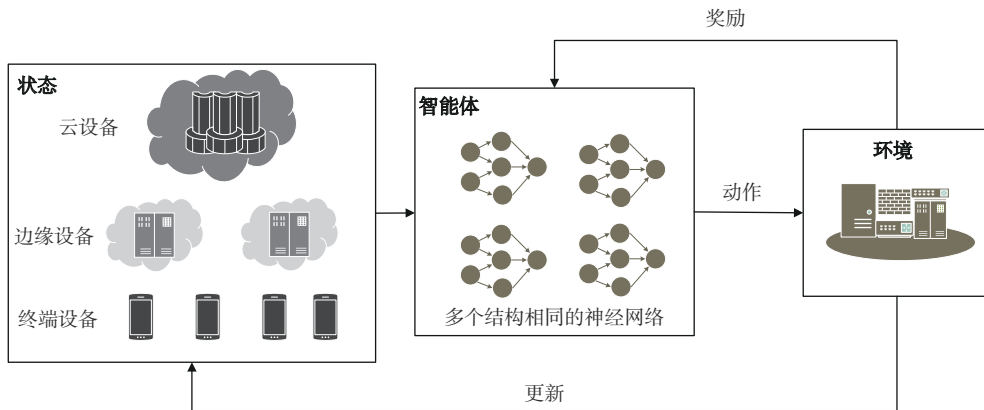


图3 MODRLRS算法训练结构

按照特定的动作轨迹进行参数更新会导致个别数据影响神经网络模型的整体性能，因此采用当前状态惩罚值的期望作为最终的目标函数，该函数的变量为 θ ，如下

$$\bar{R}(\theta) = \sum_{\tau} R(\tau) \pi_{\theta}(\tau) \quad (11)$$

通过这样的数学处理，训练算法能够迭代优化 θ 的参数值，以寻求能够使期望数值最小的参数值。

由于 $R(\tau)$ 的值需要向下优化，因此本文采用梯度下降的方法进行参数优化^[25]，对 $\bar{R}(\theta)$ 求梯度的过程

$$\begin{aligned} \nabla_{\theta} \bar{R}(\theta) &= \nabla_{\theta} \sum_{\tau} R(\tau) \pi_{\theta}(\tau) = \\ & \sum_{\tau} \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau) \end{aligned} \quad (12)$$

根据式(12)的推导，求解 $\bar{R}(\theta)$ 变成求解 $\nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau)$ 的期望，该值不能直接通过计算得到。为了得到 $\nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau)$ 的期望值，本文通过采样 N 个 τ 来近似这个期望值，具体如下

$$\nabla_{\theta} \bar{R}(\theta) \approx \frac{1}{N} \nabla_{\theta} \sum_{\tau} \log \pi_{\theta}(\tau) R(\tau) \quad (13)$$

2.3 MODRLRS算法预测模型

本文设计了结构相同但训练目标函数各异神经网络模型来实现决策。MODRLRS算法神经网络结构如图4所示，该神经网络模型有4层结构，分

别为输入层、编码器、解码器和输出层。

输入层负责对当前算网融合网络中的信息进行初步处理。具体而言，输入层将计算节点的计算资源、度和相连链路的总带宽资源数据归一化为 $3 \times n$ 特征矩阵，其中 n 为算网融合网络中计算节点的数量。

编码器由多个长短期记忆（LSTM, long short term memory）网络组成，用来处理和分析算网融合网络的数据信息。LSTM网络通过输入门、遗忘门和输出门3个神经网络结构来计算特征矩阵的隐藏状态 c_t 和输出 h_t ^[28]

$$f_t = \sigma(\theta_f \cdot [h_{t-1}, x_t] + b_f) \quad (14)$$

$$i_t = \sigma(\theta_i \cdot [h_{t-1}, x_t] + b_i) \quad (15)$$

$$\tilde{c}_t = \tanh(\theta_c \cdot [h_{t-1}, x_t] + b_c) \quad (16)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (17)$$

$$\tilde{h}_t = \sigma(\theta_o \cdot [h_{t-1}, x_t] + b_o) \cdot \tanh(c_t) \quad (18)$$

式(14)是遗忘门的计算过程，该结构针对当前输入 x 和上一神经元的输出 h_t 进行操作，并经过sigmoid函数，得到[0,1]的向量 f_t ，0代表要遗忘的某部分知识，1代表要留下的知识。式(15)~式(17)是输入门的计算过程，该结构将之前留下来的知识和当前学到的知识进行叠加，得到更新的值 i_t 和状态待选向量 \tilde{c}_t 。新的状态 c_t 如式(17)所示，由遗忘

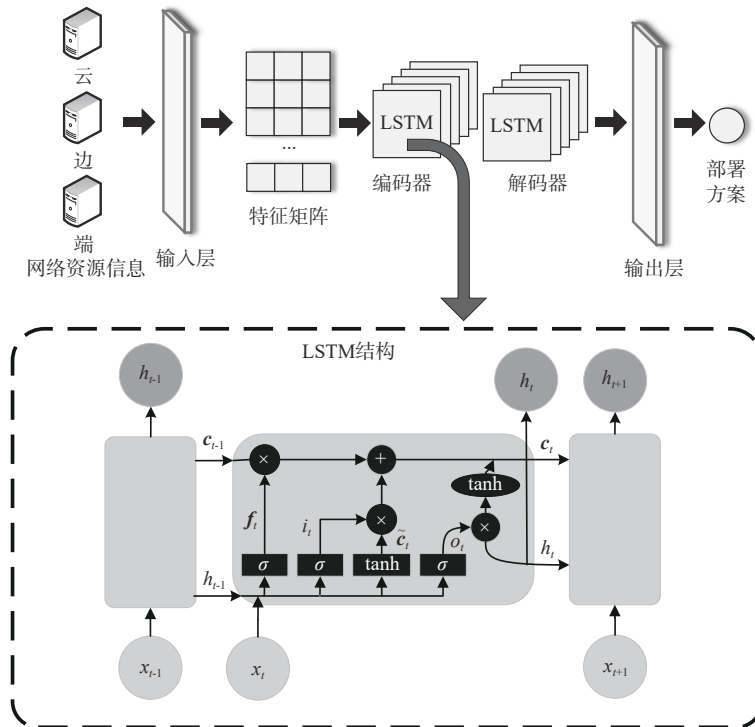


图4 MODRLRS算法神经网络结构

门的输出 f_i 、上一神经元的状态 c_{i-1} 、更新的值 i_i 和状态待选向量 \tilde{c}_i 计算得到。式(18)是输出门的计算过程,该结构通过整合新的状态信息和上一个神经元的输出得到当前输出 h_i 。编码器经过一系列LSTM运算后得到最终的隐藏状态 c_i ,并将 c_i 输入解码器中。

解码器与编码器结构相同,也由多个LSTM组成,用于处理和分析编码器输出的隐藏状态 c_i ,经过一系列与编码器相同的运算后输出计算节点得分 h_i 。

输出层将计算节点得分 h_i 转化为概率,并进行一次抽样,概率越高的计算节点越容易被抽中。

一般而言,神经网络模型的复杂度可用两个指标来描述:一是模型的参数数量,对应模型的空间复杂度;二是模型的计算量,对应模型的时间复杂度。在该模型中,LSTM需要维护4组核心数据集,分别对应输入门、输出门、遗忘门和候选态。据此可得模型总参数数量为 $4h(i+h+n)$,其中 h 为隐藏层数量, i 为输入数据的维度, n 为网络中节点的数量。基于式(14)~式(18),可以确定该神经网络模型在前向传播过程中的计算量为 $4L_i h^2 + 4L_i n h$ 。其中, L_i 表示输入的序列长度。

3 仿真测试

3.1 仿真实验环境

仿真实验参数设置见表1,包括仿真实验网络参数、计算任务请求数据集参数、神经网络训练参数以及对比算法参数。

为了严谨地评估本文所提出算法应用在真实网络的性能,本文采用了开源的Abilene真实网络拓扑作为主干网络。在此基础上,实验增加了6个具备计算处理能力的节点以及计算节点的覆盖面,旨在模拟现实中的算网融合网络环境。

为了模拟真实环境下的计算任务请求,本文根据多租户计算任务的性能需求特征和请求时间特征生成了大小为4 000的数据集。根据租户多样化应用场景的性能需求,数据集包含了具有不同时延约束、计算资源和带宽资源需求的计算任务请求。具体而言,针对车联网与工业物联网等应用场景,计算任务时延需控制在10 ms以内;对于实时竞技类游戏和音视频通话等应用场景,计算任务时延则要求不超过50 ms;对于音视频类业务,计算任务时

延要求在100 ms以内。因此本文模拟的计算任务时延需求设置为10~100 ms。此外,为了贴近真实环境下的请求到达模式,计算任务数据集中的计算任务请求到达时间符合泊松分布。

表 1 仿真实验参数设置

参数	单位	数值
云计算节点数量		3
边缘计算节点数量		14
云计算节点计算资源数量	GHz	10
边缘计算节点计算资源数量	GHz	8~10
链路带宽资源数量	Gbit/s	800~1 000
计算任务分成计算子任务的数量		2~6
子任务要求的计算资源数量	CU	1~2
数据传输所需的带宽资源数量	Gbit/s	10~100
计算任务数量		4 000
计算任务最大允许时延	ms	10~100
计算任务请求到达时间		按照泊松分布时间到达
每个事件片段平均任务请求数量		8
神经网络模型数量		100
神经元层数		3
更新步长		10
初始学习率lr		0.02
学习率变化轮数		30
学习率变化式		$lr = 0.7 lr$
遗传算法种群数量		10
遗传算法迭代次数		100

本文实验使用邻域参数传递训练算法训练了共计100个神经网络模型。这些模型在时延与负载方差权重 (λ_1, λ_2) 的配置各不相同。具体而言,权重配置从 $(1,0)$ 逐步过渡到 $(0,1)$,相邻模型间的权重变化量精细设定为0.01,从而形成了一个序列:即 $(1,0)$, $(0.99,0.01)$ 直至 $(0,1)$ 。首个神经网络模型,即权重配置为 $(1,0)$ 的模型,其训练过程始于零参数状态;而后续的神经网络模型则通过继承前一个模型的参数来进行再训练,实现了参数的有效传递与利用。在训练过程中,所有模型的初始学习率均设定为0.02,且为了优化训练效果,学习率每经过30轮训练周期后根据公式 $lr = 0.7 lr$ 即可进行更新调整。

3.2 对比实验

为了验证本文所提出训练算法加速模型的收敛情况,本文将首次训练和迭代训练的训练过程进行了对比。首次迭代与迭代训练对比如图5所示,神经网络模型在从零开始进行首次训练时经过120轮

学习进入收敛状态，复用相邻神经网络模型参数进行再训练仅需50轮可以进入收敛。本文实验设置的神经网络模型数量为100，因此迭代训练次数为99次。实验结果显示通过总计5 070轮训练能够确保所有神经网络模型达到收敛状态，而采用从零开始的训练策略则需要12 000轮的训练才能实现所有神经网络模型的收敛。该结果表明利用邻域参数传递算法进行训练能够减少57.75%的训练成本。增加神经网络模型的训练数量通常能够显著提升其性能表现，并且邻域参数传递算法的应用能大幅度降低训练过程中的计算成本，然而在实际应用于不同场景时，需根据各场景的具体需求，谨慎选择训练神经网络模型的数量，以期在训练的计算开销与性能提升之间找到一个恰当的平衡点。

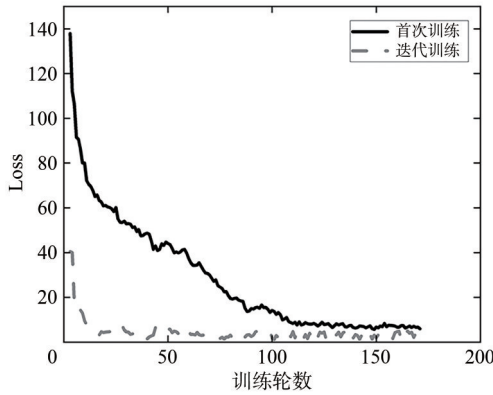


图5 首次迭代与迭代训练对比

为验证本文所提出算法的效果，将所提出的MODRLR算法与第二代非支配排序遗传算法（NSGA-II, non-dominated sorting genetic algorithm II）^[29]、第三代非支配排序遗传算法（NSGA-III, non-dominated sorting genetic algorithm III）^[30]、基于分解的多目标进化算法（MOEAD, multi-objective evolutionary algorithm based on decomposition）^[31]、策略梯度（PG, policy gradient）算法^[32]进行对比。

不同算法计算任务请求接受率对比如图6所示。在整个实验中，本文所提出算法的计算任务请求接收率稳定在98%左右，对比NSGA-II提升了4.90%的请求接受率。在多租户的场景中，多个租户共享有限的计算和网络资源，且工作负载和资源需求往往是动态变化的，这增加了为计算任务分配资源的难度。NSGA-II、NSGA-III和MOEAD这3种算法在解决约束问题时有着搜索空间的不连续性、

局部最优解的陷阱、不能在不可行解中利用有效信息的局限性。这些局限性使其在优化目标时难以充分满足计算资源和网络资源的约束，直接导致了多租户计算任务请求接收率变低。而单纯的PG算法将时延和负载均衡仅通过简单线性相加来进行多目标优化，但时延和负载均衡不是相近的优化方向，在训练过程中会相互冲突，导致其策略的计算任务请求接收率变低。

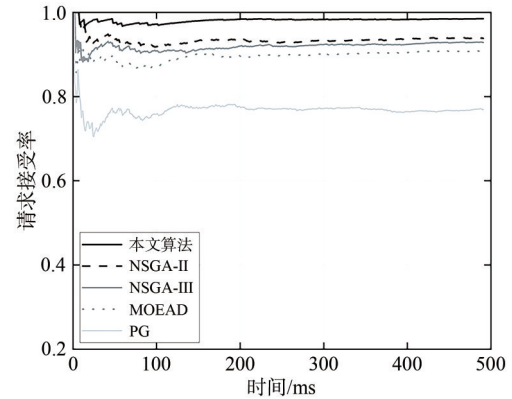


图6 不同算法计算任务请求接受率对比

为验证本文算法在满足多租户计算任务最大允许时延方面的效果，将5个算法分别处理2 000个计算任务的结果进行对比，不同算法时延对比如图7所示。在平均时延这个指标上本文算法不如NSGA-II、NSGA-III和MOEAD这3种算法，但本文算法的符合时延约束率高于其他算法，比其中最好的算法提升了4.78%。这是由于本文算法在分配算网融合网络资源时，当能满足计算任务对于时延的需求后，节省边缘的优质资源以能更好地满足其他计算任务的请求，这导致了MODRLR算法的平均时延偏高。然而，在处理多租户连续且高度个性化的计算任务请求场景中，MODRLR算法在达成计算任务时延要求后虽作出了一定程度的性能妥协，但这种策略使之能够并行满足更多计算任务对于时延的严格要求。

为验证使用MODRLR算法后的网络资源负载情况，本文将使用5个算法的负载方差进行对比，不同算法负载均衡对比如图8所示。本文算法、NSGA-II、NSGA-III、MOEAD和PG算法这5个算法的平均负载方差分别是0.265、0.472、0.363、0.154、1.054。本文所提出的MODRLR算法在负载均衡方面相较于MOEAD算法有微弱的劣势，这种差距源于实验设计的特定要求，即需针对连续性的多租户计算任务请求进行资源分配。MOEAD算法

在这种场景下的请求接受率低于本文所提出的 MODRLR 算法，在某一个时间片段内，本文算法能够同时容纳更多租户的计算任务请求，因此更难进行负载均衡。但本文算法优于除 MOEAD 算法外的其他算法，这证明 MODRLR 算法在处理计算任务时能够在一定程度上平衡计算节点的负载，使计算任务处理地稳定、高效。

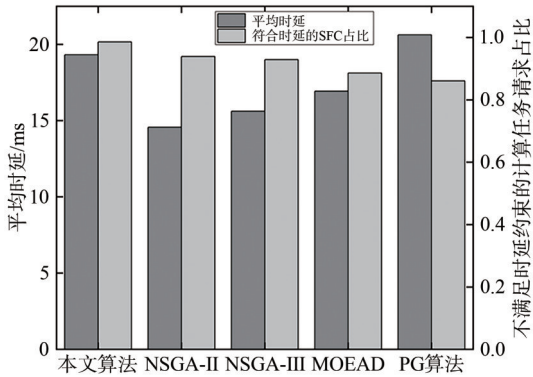


图7 不同算法时延对比

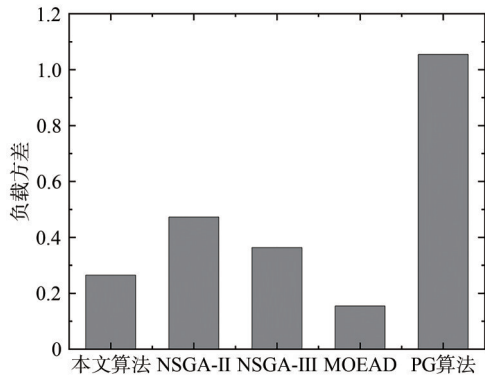


图8 不同算法负载均衡对比

4 结束语

算网融合是未来信息化社会发展的必经之路。针对算网融合场景中不同租户的个性化的服务需求，本文提出了 MODRLR 算法来同时优化计算任务时延和算网资源负载均衡，并与 NSGA-II、NSGA-III、MOEAD 和 PG 算法进行对比实验。实验结果表明，所提出的 MODRLR 算法的计算任务请求接受率能够稳定在 98% 左右，对比 NSGA-III 算法提升了 4.90% 的请求接受率；在时延优化上，MODRLR 算法提升了 4.78% 的符合时延的计算任务请求率，能够满足不同租户的时延需求；在负载均衡优化上，MODRLR 算法与最好的 MOEAD 算

法相差不大，能够高效、稳定地执行计算任务。下一阶段，笔者将对算网融合场景下计算任务的安全性和隐私性进行研究。

参考文献:

- [1] 雷波, 刘增义, 王旭亮, 等. 基于云、网、边融合的边缘计算新方案: 算力网络[J]. 电信科学, 2019, 35(9): 44-51.
LEI B, LIU Z Y, WANG X L, et al. Computing network: a new multi-access edge computing[J]. Telecommunications Science, 2019, 35(9): 44-51.
- [2] 郭凤仙, 孙耀华, 彭木根. 6G 算力网络: 体系架构与关键技术[J]. 无线电通信技术, 2023, 49(1): 21-30.
GUO F X, SUN Y H, PENG M G. Computing force networks in 6G: architecture and key technologies[J]. Radio Communications Technology, 2023, 49(1): 21-30.
- [3] 刘宇航, 张菲. 计算概念谱系: 算势、算力、算术、算法、算礼[J]. 中国科学院院刊, 2022, 37(10): 1500-1510.
LIU Y H, ZHANG F. Computing concept genealogy: potential, power, arithmetic, algorithm and ritual of computation[J]. Bulletin of Chinese Academy of Sciences, 2022, 37(10): 1500-1510.
- [4] 张宏科, 权伟, 刘康. 算力网络研究与探索[J]. 中兴通讯技术, 2023, 29(1): 1-5.
ZHANG H K, QUAN W, LIU K. Research and exploration of computing power network[J]. ZTE Technology Journal, 2023, 29(1): 1-5.
- [5] 陈晓红, 许冠英, 徐雪松, 等. 我国算力服务体系构建及路径研究[J]. 中国工程科学, 2023, 25(6): 49-60.
CHEN X H, XU G Y, XU X S, et al. Computing power service system of China and its development path[J]. Strategic Study of CAE, 2023, 25(6): 49-60.
- [6] 梁芳, 佟恬, 马贺荣, 等. 东数西算下算力网络发展分析[J]. 信息技术与政策, 2022(11): 79-83.
LIANG F, TONG T, MA H R, et al. Analysis of the development of the computing power network under east-data-west-computing project[J]. Information and Communications Technology and Policy, 2022(11): 79-8.
- [7] ITU-T. Computing power network-framework and architecture: Y.2501[S]. 2021.
- [8] 杨帆, 宋闻萱, 许方敏, 等. 工业互联网算网一体技术研究[J]. 无线电通信技术, 2023, 49(1): 63-71.
YANG F, SONG W X, XU F M, et al. Research on the application of computing force network technology in industrial Internet of things[J]. Radio Communications Technology, 2023, 49(1): 63-71.
- [9] 王少鹏, 邱奔. 算网协同对算力产业发展的影响分析[J]. 信息技术与政策, 2022(3): 29-33.
WANG S P, QIU B. Analysis on the impact of computing network collaboration on the development of computing power industry[J]. Information and Communications Technology and Policy, 2022(3): 29-33.

- [10] 王淑玲, 孙杰, 王鹏, 等. 云边协同中的资源调度优化[J]. 电信科学, 2023, 39(2): 163-170.
WANG S L, SUN J, WANG P, et al. Resource scheduling optimization in cloud-edge collaboration[J]. Telecommunications Science, 2023, 39(2): 163-170.
- [11] HAZRA A, RANA P, ADHIKARI M, et al. Fog computing for next-generation Internet of things: fundamental, state-of-the-art and research challenges[J]. Computer Science Review, 2023, 48: 100549.
- [12] WU Y L, DAI H N, WANG H Z, et al. A survey of intelligent network slicing management for industrial IoT: integrated approaches for smart transportation, smart energy, and smart factory[J]. IEEE Communications Surveys & Tutorials, 2022, 24(2): 1175-1211.
- [13] XIONG Z H, ZHANG Y, LUONG N C, et al. The best of both worlds: a general architecture for data management in blockchain-enabled Internet-of-things[J]. IEEE Network, 2020, 34(1): 166-173.
- [14] ITU-T. Signalling requirements for service deployment in computing power network: Q.4140[S]. 2023.
- [15] CUI Y Y, ZHANG D G, ZHANG T, et al. A novel offloading scheduling method for mobile application in mobile edge computing[J]. Wireless Networks, 2022, 28(6): 2345-2363.
- [16] JAMIL B, IJAZ H, SHOJAFAR M, et al. Resource allocation and task scheduling in fog computing and Internet of everything environments: a taxonomy, review, and future directions[J]. ACM Computing Surveys, 2022, 54(11s): 1-38.
- [17] ADDYA S K, SATPATHY A, GHOSH B C, et al. CoMCLoud: virtual machine coalition for multi-tier applications over multi-cloud environments[J]. IEEE Transactions on Cloud Computing, 2023, 11(1): 956-970.
- [18] FARHADI V, MEHMETI F, HE T, et al. Service placement and request scheduling for data-intensive applications in edge clouds[J]. IEEE/ACM Transactions on Networking, 2021, 29(2): 779-792.
- [19] GAO Y F, YAN Z B, ZHAO K L, et al. Joint optimization of server and service selection in satellite-terrestrial integrated edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73(2): 2740-2754.
- [20] BAHRAMI B, KHAYYAMBASHI M R, MIRJALILI S. Multi-objective placement of edge servers in MEC environment using a hybrid algorithm based on NSGA-II and MOPSO[J]. IEEE Internet of Things Journal, 2024, 11(18): 29819-29837.
- [21] ABDELMONEEM R M, BENSLIMANE A, SHAABAN E. Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures[J]. Computer Networks, 2020, 179: 107348.
- [22] NI L N, ZHANG J Q, JIANG C J, et al. Resource allocation strategy in fog computing based on priced timed petri nets[J]. IEEE Internet of Things Journal, 2017, 4(5): 1216-1228.
- [23] 张维庭, 孙呈蕙, 王洪超, 等. 算网资源智能适配与融合调度方法[J]. 电信科学, 2023, 39(9): 12-20.
ZHANG W T, SUN C H, WANG H C, et al. Intelligent adaptation and integrated scheduling method for computing and networking resources[J]. Telecommunications Science, 2023, 39(9): 12-20.
- [24] ZHAO X Y, ZONG Q, TIAN B L, et al. Fast task allocation for heterogeneous unmanned aerial vehicles through reinforcement learning[J]. Aerospace Science and Technology, 2019, 92: 588-594.
- [25] WU Q, WANG W H, FAN P Y, et al. Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks[J]. IEEE Transactions on Network and Service Management, 2024, 21(4): 4179-4196.
- [26] WANG X B, WU Q, FAN P Y, et al. Vehicle selection for C-V2X mode 4-based federated edge learning systems[J]. IEEE Systems Journal, 2024, 18(4): 1927-1938.
- [27] ZHU H B. Pareto improvement: a GRA perspective[J]. IEEE Transactions on Computational Social Systems, 2023, 10(3): 1241-1251.
- [28] FAN Y Q, LIU J H, YE H, et al. TA-LSTM: a time and attribute aware LSTM for deep flight track clustering[J]. IEEE Transactions on Aerospace and Electronic Systems, 2023, 59(5): 7047-7060.
- [29] CAO K, WENG J. REPFS: reliability-ensured personalized function scheduling in sustainable serverless edge computing[J]. IEEE Transactions on Sustainable Computing, 2024, 9(3): 494-511.
- [30] PENG K, HUANG H L, ZHAO B H, et al. Intelligent computation offloading and resource allocation in IIoT with end-edge-cloud computing using NSGA-III[J]. IEEE Transactions on Network Science and Engineering, 2023, 10(5): 3032-3046.
- [31] WANG H Z, TANG L X, XIAO M, et al. Multi-objective optimization for joint communication and computing resource allocation in NOMA-based MEC system[C]//Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC). Piscataway: IEEE Press, 2024: 1-8.
- [32] WANG S M, SONG X Q, XU H, et al. Joint offloading decision and resource allocation in vehicular edge computing networks[J]. Digital Communications and Networks, 2023.

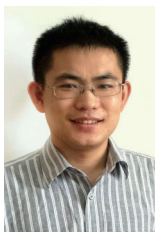
[作者简介]



胡宇翔 (1982-), 男, 信息工程大学信息技术研究所、先进通信网全国重点实验室教授、博士生导师, 主要研究方向为新型网络架构、网络空间安全、智能路由与可编程转发等。



冯旭 (1997-), 男, 信息工程大学信息技术研究所博士生, 主要研究方向为零信任网络、下一代互联网等。



董永吉(1983-), 男, 信息工程大学信息技术研究所、先进通信网全国重点实验室副教授, 主要研究方向为网络可编程数据平面和网络安全。



庄雷(1963-), 女, 郑州大学计算机与人工智能学院教授、博士生导师, 主要研究方向为模型检查、未来网络架构和网络虚拟化等。



和孟佺(1994-), 女, 郑州大学网络空间安全学院、嵩山实验室助理研究员, 主要研究方向为下一代互联网、互联网应用程序等。



宋艳蕊(2000-), 女, 郑州大学计算机与人工智能学院硕士生, 主要研究方向为下一代互联网和服务功能链的安全部署。