专题：网络智能化与生成式人工智能

算网资源智能适配与融合调度方法

张维庭，孙呈蕙，王洪超，代嘉宁
（北京交通大学移动专用网络国家工程研究中心，北京 100044）

摘　要：为有效支撑网络和计算深度融合的发展需求，新型的算力网络架构应运而生。在此背景下，如何实现算网资源的智能感知以及计算任务的高效调度，是当前网络需要解决的关键问题。为此，分析了面向算网融合的新型网络场景，设计了计算任务与算力节点的调度模型，提出了一种基于深度强化学习的资源调度算法。所提算法通过感知用户设备、算网资源可用容量和链路状态等关键信息，能够智能地做出系统成本最小的调度决策。最后，通过仿真实验验证了所提算法在节约系统成本方面的有效性。

关键词：算力网络；资源调度；深度强化学习
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2023174

Intelligent adaptation and integrated scheduling method for computing and networking resources

ZHANG Weiting, SUN Chenghui, WANG Hongchao, DAI Jianing
National Engineering Research Center of Advanced Network Technologies,
Beijing Jiaotong University, Beijing 100044, China

Abstract: To effectively support the development needs of the deep integration of networking and computing, a novel computing and network convergence architecture has emerged. In this context, how to realize the intelligent perception of computing and networking resources and the efficient scheduling of computational tasks are key problems. To this end, the new network scenario for computing and networking convergence were analyzed, a scheduling model for computing tasks and nodes was designed, and a deep reinforcement learning-based resource scheduling algorithm was proposed. The proposed algorithm was able to intelligently make scheduling decisions that minimize the system cost by sensing key information such as user devices, available capacity of computing and networking resources, and link status. Finally, the effectiveness of the proposed algorithm in saving system cost was verified by simulation experiments.

Key words: computing and network convergence, resource scheduling, deep reinforcement learning

收稿日期：2023−07−31；**修回日期**：2023−09−04
通信作者：王洪超，hcwang@bjtu.edu.cn
基金项目：国家自然科学基金资助项目（No.62201029）；中国博士后科学基金资助项目（No.2022M710007，No.BX20220029）
Foundation Items: The National Natural Science Foundation of China (No.62201029), China Postdoctoral Science Foundation (No.2022M710007, No.BX20220029)

0　引言

当前信息时代高速发展，数字化和智能化应用的迅猛增长导致数据量呈指数级增加。海量数据不仅需要可靠传输，而且需要实时智能分析，使得传统网络在时延、带宽、可靠性等方面面临挑战。在此背景下，面向算网融合的新型网络架构被提出，如算力网络[1-2]。算力网络旨在深度融合计算资源与网络资源，集中式或分布式管理云、端、边 3 层架构的算力资源。其目标是通过智能化调度和适配，高效利用有限的算力资源，以满足不断增长的数据处理需求，提供优质的服务质量和个性化的用户体验。北京交通大学提出了名为“智算融合网络”的新型网络体系。该智算融合网络不同于简单的算力资源和网络分离式协同，而是致力于实现二者的深度融合，将算网资源和服务资源紧密结合，为用户提供高质量、差异化的个性化服务[3]。2023 年，北京交通大学进一步推出了“三层三域”的智算融合网络架构，旨在以算力和网络的深度融合为目标，建立高效、智能、差异化的算力服务平台，满足用户对算力按需调度和分配的需求[4]。

在算力网络框架下，管理和适配有限算力资源成为亟待解决的问题。目前的研究，如文献[5-8]，集中在传统网络中对资源进行调度，如何在复杂多变的新型网络环境中，合理配置和分配算力资源，确保数据的快速传输、高效处理和稳定存储，是当前网络研究的热点问题之一。为了实现资源的优化利用和高效调度，必须面对资源不对称、网络负载波动、任务多样性等挑战，进一步探索并创新性地提出适应算力网络特点的资源管理策略和调度算法。

1　基于深度强化学习的算网资源融合调度算法研究

算网资源融合调度算法可以动态地感知算力节点资源信息，根据不同任务的计算需求和算网资源的可用性进行智能匹配，在时延和能耗之间取得平衡，最大限度地提高计算资源和网络资源的利用率和性能，从而降低系统成本。网络环境处于不断变化之中，这使得算网资源融合调度算法需要很高的灵活性，而深度强化学习算法可以自动学习和调整系统的决策策略，可以适应环境并对新的情况做出合理的决策。

首先介绍了智算融合网络的网络场景，设计了该场景下的通信模型和计算模型；其次提出一种针对时延和能耗综合优化的资源调度算法——基于深度 Q 网络的资源调度（deep Q network-based resource scheduling，DQNRS）算法；最后通过设计实验验证了 DQNRS 算法的收敛性，并将 DQNRS 算法与其他调度策略进行对比，证明了该算法可以在变化的网络场景中产生更小的系统开销。

1.1　系统模型

介绍了应用算网资源融合调度算法的网络场景，构建了处理用户任务的计算模型和通信模型，将最小化时延和能耗的综合开销作为优化目标，求解优化问题。

1.1.1　网络场景

网络场景示意图如图 1 所示，网络场景由用户、基站、算力节点（computing node，CN）和算网资源控制器组成。其中，用户 U 是计算任务的发起者，C_{local} 表示用户 U 本地的计算能力。基站通过光纤与算力节点相连，建立通信链路。算力节点为包括中央处理器（central processing unit，CPU）、图形处理器（graphics processing unit，GPU）、现场可编程门阵列（field programmable gate array，FPGA）等异构计算能力的高性能服务器，用来处理用户上传的任务。假设网络模型中共有 k 个算力节点，$V=\{V_1,V_2,\cdots,V_k\}$ 表示算力节点的集合，其中，V_j 表示第 j 个算力节点，$j=1,2,3,\cdots,k$。$C=\{C_1,C_2,\cdots,C_k\}$ 表示节点计算

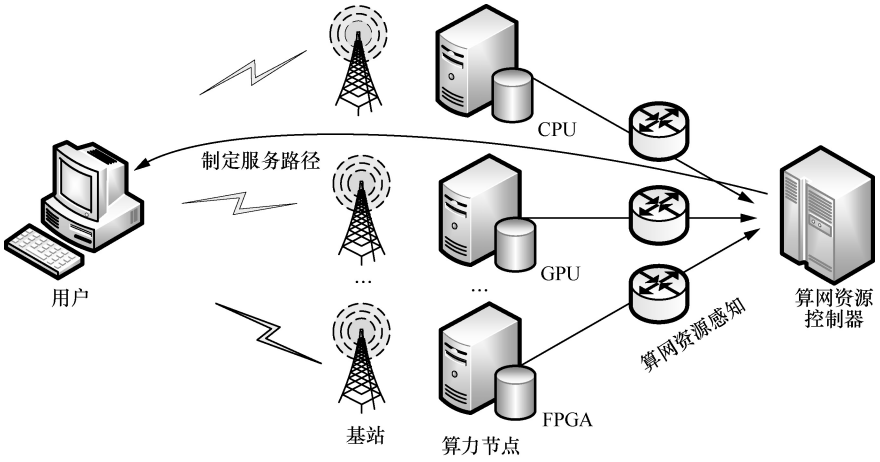


图1　网络场景示意图

能力的集合，其中，C_j 表示第 j 个算力节点 V_j 的计算能力。算网资源控制器运行 DQNRS 算法，通过动态感知网络内资源状态，做出资源调度决策。

1.1.2　通信模型

用户提交的计算任务 A 可以划分成 n 个子任务，即 $A=\{A_1,A_2,\cdots,A_k\}$，A_i^j 表示计算任务 A 中的第 i 个子任务在第 j 个算力节点上执行，其中，$j=1,2,3,\cdots,k$，$j=0$ 表示计算任务在本地执行。将每个子任务 A_i 描述为 $A_i=\{\omega_i,c_i,s_i\}$，其中 ω_i 表示子任务 A_i 的输入数据量；c_i 表示执行子任务 A_i 所需的计算资源，即CPU时钟周期数；s_i 表示完成子任务 A_i 后的输出数据量。为了保证用户服务质量，定义 τ_{MAX} 为完成计算任务 A 所设定的最大允许时延。

当用户发起计算任务时，算网资源控制器需要根据计算任务数据量、网络状态以及各节点处计算能力，综合做出资源调度决策。决策利用布尔变量 $\zeta_i(l)=\{0,1\}$ 进行描述，当 $\zeta_i(l)=0$ 时，子任务 A_i 直接在本地执行；当 $\zeta_i(l)=1$ 时，子任务 A_i 被上传到算力节点处执行。

根据文献[9]中的无线信道模型，忽略干扰因素，上行链路的传输速率为：

$$R_{ij}^{\mathrm{up}}=B_{ij}\mathrm{lb}\left(1+\frac{p_{\mathrm{local}}^{\mathrm{Tr}}h_{ij}}{N_0}\right) \tag{1}$$

其中，B_{ij} 表示用户到算力节点 V_j 之间的链路带宽，单位为 Hz；$p_{\mathrm{local}}^{\mathrm{Tr}}$ 表示用户终端设备的发射功率，单位为 W；h_{ij} 表示上行链路的信道增益；N_0 表示信道中的噪声功率。

1.1.3　计算模型

（1）本地执行模型

对于子任务 A_i，当 $\zeta_i(l)=0$ 时，计算任务在本地设备执行，本地 CPU 处理子任务 A_i 所消耗的时间为：

$$T_{i,\mathrm{local}}=\frac{c_i}{C_{\mathrm{local}}} \tag{2}$$

本地设备对子任务 A_i 进行处理时的能耗为：

$$E_{i,\mathrm{local}}=p_{\mathrm{local}}^{\mathrm{comp}}\cdot T_{i,\mathrm{local}} \tag{3}$$

其中，$p_{\mathrm{local}}^{\mathrm{comp}}$ 表示本地设备的计算功率。

子任务 A_i 在本地设备执行所需要的成本为：

$$\begin{aligned}&\mathrm{Cost}_{i,\mathrm{local}}=\lambda^{\mathrm{T}}\cdot T_{i,\mathrm{local}}+\lambda^{\mathrm{E}}\cdot E_{i,\mathrm{local}}\\&\text{s.t. }\lambda^{\mathrm{T}}+\lambda^{\mathrm{E}}=1\end{aligned} \tag{4}$$

其中，$0\leqslant\lambda^{\mathrm{T}}$、$0\leqslant\lambda^{\mathrm{E}}\leqslant 1$ 分别表示执行时延和能耗的权重因子。

（2）远程计算模型

对于子任务 A_i，当 $\zeta_i(l)=1$ 时，计算任务被传输到算力节点执行。总时延包括上行链路的传输时延和节点处理时延，即：

$$T_{i,\mathrm{cn}}^{j}=T_{ij}^{\mathrm{up}}+T_{ij}^{\mathrm{comp}} \tag{5}$$

其中，T_{ij}^{up} 表示子任务 A_i 通过无线链路上传到节

点 V_j 的传输时延。T_{ij}^{comp} 表示算力节点 V_j 对计算任务 A_i 进行计算的处理时延。所以：

$$T_{ij}^{\text{comp}}=\frac{c_i}{C_j} \tag{6}$$

由式（1）可得：

$$T_{ij}^{\text{up}}=\frac{w_i}{R_{ij}^{\text{up}}}=\frac{w_i}{B_{ij}\text{lb}\left(1+\dfrac{p_{\text{local}}^{\text{Tr}}\cdot h_{ij}}{N_0}\right)} \tag{7}$$

用户通过上行链路到节点的传输能耗和节点处理数据时本地设备等待结果回传时产生的能耗为：

$$E_{i,\text{cn}}^{j}=E_{ij}^{\text{up}}+E_{ij}^{\text{comp}} \tag{8}$$

其中，E_{ij}^{up} 表示子任务 A_i 传输到算力节点 V_j 过程中，通过上行通信链路所产生的传输能量消耗；E_{ij}^{comp} 表示 V_j 节点执行子任务 A_i 时，本地设备等待结果回传时产生的能耗。所以：

$$E_{ij}^{\text{up}}=p_{\text{local}}^{\text{Tr}}\cdot T_{ij}^{\text{up}}=\frac{p_{\text{local}}^{\text{Tr}}\cdot w_i}{B_{ij}\text{lb}\left(1+\dfrac{p_{\text{local}}^{\text{Tr}}\cdot h_{ij}}{N_0}\right)} \tag{9}$$

在算力节点执行计算任务的过程中，本地设备等待结果返回的功率记作 $p_{\text{local}}^{\text{wait}}$，则任务在节点处进行计算的处理能耗可以表示为：

$$E_{ij}^{\text{comp}}=p_{\text{local}}^{\text{wait}}\cdot T_{ij}^{\text{comp}}=\frac{p_{\text{local}}^{\text{wait}}\cdot c_i}{C_j} \tag{10}$$

由式（5）和式（8）可得到，综合时延和能耗，计算任务 A_i 在算力节点执行的总成本 $\text{Cost}_{i,\text{cn}}$ 为：

$$\text{Cost}_{i,\text{cn}}=\lambda^{\text{T}}T_{i,\text{cn}}^{j}+\lambda^{\text{E}}E_{i,\text{cn}}^{j} \tag{11}$$

（3）问题建模

子任务只能选择本地计算或算力节点计算两种方式中的一种，用户完成某个子任务所消耗的总成本为：

$$\text{Cost}_i=(1-\zeta_i(l))\text{Cost}_{i,\text{local}}+\zeta_i(l)\text{Cost}_{i,\text{cn}} \tag{12}$$

完成整个计算任务所消耗的总成本为：

$$\text{Cost}_{\text{all}}=\sum_{i=1}^{n}(1-\zeta_i(l))\text{Cost}_{i,\text{local}}+\zeta_i(l)\cdot\text{Cost}_{i,\text{cn}} \tag{13}$$

考虑最小化时延和能耗综合开销的资源调度问题，以最小化系统成本为目标，将问题建模为：

$$\begin{aligned}&\min \text{Cost}_i\\&\text{s.t.}\quad \text{C1}:\zeta_i(l)\in\{0,1\},\forall A_i\in A\\&\qquad \text{C2}:\left(\sum_{i=1}^{n}T_{i,\text{local}},T_{i,\text{cn}}\right)\leqslant\tau_{\text{Max}}\end{aligned} \tag{14}$$

优化目的是最小化用户处理计算任务的时延和能耗的加权系统成本。其中，约束 C1 是对任务调度位置进行约束，约束 C2 是对任务处理总时延进行约束。

1.2　资源调度算法设计

在智算融合网络场景中，算力网络控制器涉及多个变量和状态，如算力节点资源利用率、计算任务请求量、网络负载等，这导致状态空间非常庞大和复杂。深度强化学习中的深度 Q 网络（deep Q-network，DQN）算法能够通过深度神经网络逼近 Q 函数，有效地处理高维、非线性的状态空间，根据复杂的状态信息，做出准确的调度策略。算网资源控制器使用 DQN 算法能够实现自适应学习，采用经验回放技术提高训练效率，并在大规模网络下高效地进行决策。这些特性使得 DQN 算法适用于优化网络资源调度任务。本文在 DQN 算法的基础上融入智算融合网络场景，设计了 DQNRS 算法。算法具体流程为首先将算网资源状态建模为马尔可夫决策过程，然后使用已经配置好超参数的深度神经网络，将设定好的状态输入深度神经网络不断优化 Q 函数，经过一定轮次训练好资源调度决策模型。在本算法中，将计算任务 A 从发起请求到获得最终结果的时间分割为 V 个时间片，每一个时间片都要决定对应子任务 A_i 的执行方式，记变量 t 为当前时刻，$t\in\{1,2,3,\cdots,V\}$，在每个时间片内系统的状态不发生改变。

1.2.1　马尔可夫决策过程建模

（1）状态空间 S

将计算任务调度到算力节点时，当前时刻 t 的系统状态由 3 个部分组成，分别是本地设备状态 s_t^{local}、信道状况 s_t^{channel} 和算力节点的状态 s_t^{nodes}：

$$s_t=\left(s_t^{\text{local}},s_t^{\text{channel}},s_t^{\text{nodes}}\right) \tag{15}$$

本地设备状态 s_t^{local} 包括设备当前的网络质量、当前 CPU 负载率和计算任务的相关信息：

$$s_t^{\text{local}}=(n_t,\delta_t,A_i) \tag{16}$$

其中，n_t 表示当前时刻 t 下本地智能终端的网络连接质量，一般用信号强弱表示，信号强度反映了当前时刻设备与基站之间的信号传输质量[10-11]；δ_t 表示当前时刻 t 下本地智能终端的 CPU 负载率；A_i 表示需要进行调度决策计算任务的相关信息。

网络场景下算力节点传输计算任务的基站与节点是一一对应的，那么在当前时刻 t 下，用户可用的基站数和算力节点数均为 $k\in\{1,2,3,\cdots,K\}$，第 j 个基站为用户提供的上行通信链路的信道状况记作 σ_j，第 j 个算力节点的计算能力记作 C_j，那么信道状况 s_t^{channel} 和算力节点的状态 s_t^{nodes} 可以表示为：

$$s_t^{\text{channel}}=(\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_k) \tag{17}$$

$$s_t^{\text{nodes}}=(C_1,C_2,C_3,\cdots,C_k) \tag{18}$$

综上，整个系统的状态空间 S 定义为：

$$S=(s_1,s_2,\cdots,s_t,\cdots,s_V) \tag{19}$$

（2）行为空间 A

在当前时刻 t，计算任务 A_i 的调度决策总集合可表示为：

$$a_t(l)=[a_{i,0}(l),a_{i,1}(l),\cdots,a_{i,j}(l),\cdots,a_{i,k}(l)] \tag{20}$$

其中，$a_{i,j}(l)=\{0,1\}$，$a_{i,0}(l)=1$ 表示服务请求 A_i 在本地终端侧执行，$a_{i,j}(l)=1(1\leqslant j\leqslant k)$ 表示服务请求 A_i 调度到节点 V_j 处理。因此，行为空间 A 可以表示为：

$$A=[a_1(l),a_2(l),\cdots,a_V(l)] \tag{21}$$

（3）奖励函数 R

将在时刻 t 下，环境状态 s_t 所采取的行为 a_t 记作一组状态—动作对 (s_t,a_t)。每个 (s_t,a_t) 都有一个奖励值，用 $R(s_t,a_t)$ 表示。优化目标通过降低时延和能耗达到系统成本最小化，将奖励函数设置为系统成本的倒数，即：

$$R(s_t,a_t)=\frac{1}{\text{Cost}(s_t,a_t)} \tag{22}$$

其中，$\text{Cost}(s_t,a_t)$ 表示当前状态下的系统成本。奖励函数 $R(s_t,a_t)$ 越大，说明当前系统成本越小。整个过程中的长期奖励值可以表示为：

$$R_{\text{long}}=\sum_{i=0}^{V}\gamma^i R(s_{t,i+1},a_{t,i+1}) \tag{23}$$

其中，$0\leqslant\gamma\leqslant1$，表示折扣因子。

1.2.2 算法具体设计

DQNRS 算法基于典型深度强化学习算法——DQN 算法实现[12-13]，旨在找到不同系统状态 s_t 下的最优动作决策，实现长期累计回报的最大化。在 DQN 算法中，为了降低数据之间的相关性，引入经验回放机制和目标网络机制优化模型的训练效果[14]。经验回放机制可以将历史样本进行重复利用，避免过度依赖最新获取的数据。目标网络机制采用双重神经网络方式，即评估 Q 网络和目标 Q 网络，目标 Q 网络和评估 Q 网络结构相同，但目标 Q 网络的参数固定一段时间再进行更新。通过这种关系，DQN 算法能够解决训练过程中的反馈循环问题，提高算法的稳定性和训练效率。DQNRS 算法整体训练流程如图 2 所示。

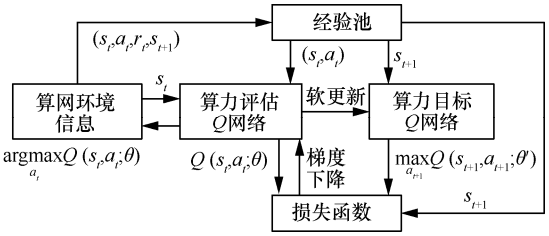


图 2 DQNRS 算法整体训练流程

评估 Q 网络的参数为 θ，目标 Q 网络的参数为 θ'。在 DQN 算法中，通过参数为 θ 的神经网络拟合状态—动作值函数 $Q(s_t,a_t)$[15]，即：

$$Q(s_t,a_t;\theta)\approx Q(s_t,a_t)=E\left[\sum_t\gamma^t R(s_t,a_t)\,|\,s,a\right] \tag{24}$$

其中，$\gamma\in[0,1]$ 表示折扣因子。

为防止算法陷入局部最优解，动作选择策略依据贪婪（$\varepsilon-\text{greedy}$）策略，即在当前状态下，

智能体有$1-\varepsilon$的概率执行最优动作，而有ε的概率随机选取一个任意动作执行。最优动作为：

$$a_t = \underset{a_t}{\operatorname{argmax}} Q(s_t, a_t; \theta) \tag{25}$$

依据$\varepsilon-$greedy 策略选择动作a_t后，将其作用于当前环境状态s_t，获得实时奖励值r_t和下一状态s_{t+1}，将s_t、a_t、r_t、s_{t+1}组成的四元组放入记忆库存储。同时，目标Q网络依据从记忆库中随机选取的样本生成一个目标Q值，即：

$$Q(s_t, a_t; \theta') = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta') \tag{26}$$

通过最小化损失函数 Loss 对当前评估Q网络的参数进行更新，函数通常采用均方误差函数，即：

$$\text{Loss}(\theta) = E[(Q(s_{t+1}, a_{t+1}; \theta') - Q(s_t, a_t; \theta))^2] \tag{27}$$

DQNRS 算法流程设计如算法 1 所示。

算法 1　DQNRS 算法

初始化　回放记忆库M的大小：N

评估Q网络、目标Q网络的参数：θ、θ'

目标Q网络的更新步长：U

回合 episode$=1$、步数 step$=1$

for　episode$=1,2,\cdots,n$ do

获取本地移动设备状态、通信链路信道状况、算力节点状态的信息，形成初始状态s_i

for　step$=1,2,\cdots,k$ do

$$a_t = \begin{cases} \text{random action}, & p = 1-\varepsilon \\ \underset{a_t}{\operatorname{argmax}} Q(s_t, a_t; \theta), & p = \varepsilon \end{cases}$$

执行动作a_t，得到奖励r_t和下一个状态s_{t+1}

将四元组(s_t, a_t, r_t, s_{t+1})存储到记忆回放库M中

从记忆回放库中抽取随机样本(s_i, a_i, r_i, s_{i+1})

if　episode$==n$&&step$==k$

then　预测奖励值$y_i = r_i$

else　$y_i = r_i + \gamma \max\limits_{a_{i+1}} Q(s_{i+1}, a_{i+1}; \theta')$

end if

对 Loss 函数进行梯度下降，更新Q网络

if　step%$U == 0$

$\theta' = \theta$

end if

end for

end for

2　实验与分析

下面介绍仿真参数的设置并分析实验结果。

2.1　仿真参数设计

仿真实验参数的设置见表 1。

表 1　仿真实验参数的设置

参数表示	具体描述	取值
K_{bs}	基站数量	5
K_{nodes}	算力节点数量	5
N	中心算力控制器数量	1
w_i	计算任务输入数据大小/KB	{300, 500, 700, 900, 1 100}
c_i	任务所需要的计算资源/cycle	1×10^9
C	算力节点的计算能力/GHz	{4, 8, 12, 16, 20}
C_{local}	本地设备的计算能力/GHz	{0.5, 0.6, 0.7, 0.8, 0.9, 1.0}
B	上行链路带宽/MHz	15
S/N_0	上行链路信噪比/dB	{15, 20, 25, 30, 35}
$p_{\text{local}}^{\text{Tr}}$	本地设备的发射功率/mW	50
$p_{\text{local}}^{\text{wait}}$	本地设备的静态功率/mW	10
λ^{T}、λ^{E}	用户偏好权重	0.5、0.5

在 DQNRS 算法中，记忆回放库的大小设为 10 000，最大训练回合数为 400，单回合步数最大值为 300，随机抽取样本的步长为 50，学习率为 0.005，奖励折扣因子γ和贪婪系数ε均为 0.9。

2.2　实验结果分析

2.2.1　算法收敛情况

DQNRS 算法的收敛情况如图 3 所示，算法在第 40 个回合开始收敛，这是因为 DQNRS 算法中的经验池需要收集足够的样本才开始训

练。算法在第 60 回合基本稳定，不再发生显著的变化，证明了算法的收敛性。系统成本在 0.1～0.2 区间浮动，说明在网络参数动态变化的情况下，DQNRS 算法相比于本地执行资源调度策略，可以将系统成本降低 85%左右，并有较快的收敛速度。

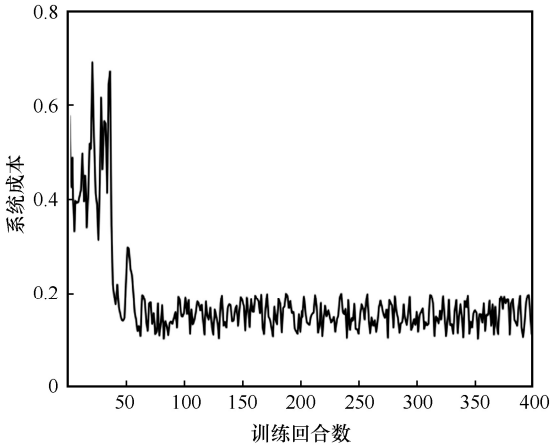


图 3　DQNRS 算法的收敛情况

2.2.2　性能对比

对比执行策略的设置如下。

- DQNRS 算法：深度强化学习算法通过感知算网资源信息，与计算任务相匹配得到的执行策略。
- 全本地执行策略：所有的计算任务全部在本地执行。
- 全节点执行策略：所有的计算任务调度到算力节点执行。
- 随机执行策略：计算任务随机选择在本地或者算力节点执行。
- 基于贪心算法的节点执行策略：基于贪心算法，计算任务通过最佳上行通信信道传输。

系统成本定义为调度执行成本与本地执行成本的比值，对比实验以优化系统成本为目标，分别将计算任务的输入数据大小、算力节点的计算能力和本地设备计算能力的比值、通信链路状况作为自变量，将系统成本作为因变量，对以上几种算法进行分析。

（1）系统成本与计算任务输入数据大小

系统成本与输入数据大小的关系如图4所示，给出了输入数据大小对系统成本的影响。由图 4 可知，DQNRS 算法实现了最优的系统性能，同时系统成本最小。随着输入数据大小的增加，所有算法的系统成本都在增加，其原因在于计算数据需要通过通信链路传输到算力节点，数据量增加导致传输的时延和能耗增加，进而导致系统成本增加。基于贪心算法的节点执行策略虽然每次调度时通过最佳上行通信链路传输，但其没有考虑算网资源的适配，因此没有达到最佳的效果；DQNRS 算法能够动态感知算力资源和网络状态，综合做出资源调度决策，进而实现最优的系统性能。

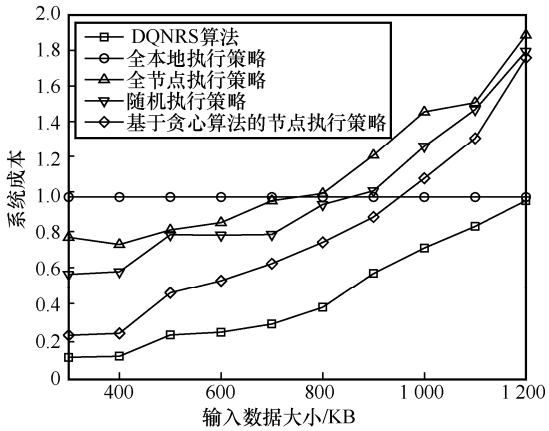


图 4　系统成本与输入数据大小的关系

（2）系统成本与算力节点和本地设备的计算能力的比值

算力节点与本地设备计算能力的比值对系统成本的影响如图 5 所示，DQNRS 算法产生的系统成本最小。在本地设备计算能力与算力节点计算能力相差较小时，基于贪心算法的节点执行策略产生的系统成本小于全节点执行策略产生的系统成本，其原因在于采用贪心策略能够有效降低传输过程中的时延和能耗。在算力节点计算能力远高于本地设备计算能力时，将任务调度到算力节点执行所产生的系统开销更小。

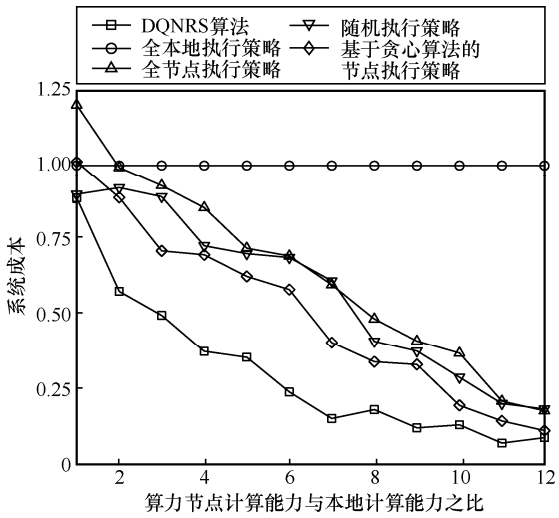


图 5　算力节点与本地设备计算能力的比值对系统成本的影响

（3）系统开销与通信链路的状况

上行通信链路信噪比对系统成本的影响如图 6 所示，随着信噪比的增大，各个算法所产生的系统成本在减少。其原因在于任务需要通过信道传输到算力节点执行时，较好的通信链路能够有效节省传输过程中的时延和能耗。通过分析得出，在通信链路较差时，将计算任务调度到算力节点执行减少的处理时延无法抵消传输时延增加带来的影响，所以在网络资源调度时需要重点考虑通信链路情况。

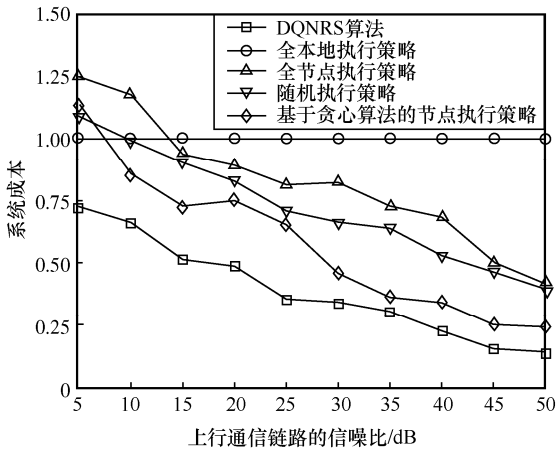


图 6　上行通信链路信噪比对系统成本的影响

3　结束语

本文提出了一种基于深度强化学习的算网资源智能适配与融合调度方法，将用户的计算任务和算力节点相适配，然后把计算任务调度到相应的算力节点实现有效的算力资源分配。通过实验可以验证，本文提出的调度方法可以完成算网资源分配的优化，且在调度过程中的总成本和网络性能参数上，相对于其他常规调度策略有更优的表现。本文方法为实现智算融合网络的调度机制提供了一种可行方案。未来将聚焦于解决面向用户移动性的算网资源融合调度问题，旨在为高速移动场景下的算力用户提供更稳定、高效的计算服务体验。

参考文献：

[1] IETF, Routing Area Working Group (RTGWG). A report on compute first networking (CFN) field trial[R]. 2019.

[2] IETF, Computing in the Network Research Group (COINRG). Directions for computing in the network[R]. 2020.

[3] 张宏科, 于成晓, 权伟, 等. 融算网络体系基础研究[J]. 电子学报, 2022, 50(12): 2928-2934.
ZHANG H K, YU C X, QUAN W, et al. Fundamental research on computing integration networking[J]. Acta Electronica Sinica, 2022, 50(12): 2928-2934.

[4] ZHANG H K, QUAN W, CHAO H C, et al. Smart identifier network: a collaborative architecture for the future Internet[J]. IEEE Network, 2016, 30(3): 46-51.

[5] GAO J, KUANG Z F, GAO J, et al. Joint offloading scheduling and resource allocation in vehicular edge computing: a two layer solution[J]. IEEE Transactions on Vehicular Technology, 2023, 72(3): 3999-4009.

[6] 刘润滋, 吴伟华, 张文柱, 等. 基于图学习的密集空间网络传输资源调度方法[J]. 电子学报, 2021, 49(11): 2133-2137.
LIU R Z, WU W H, ZHANG W Z, et al. Graph learning based transmission resources scheduling in dense space networks[J]. Acta Electronica Sinica, 2021, 49(11): 2133-2137.

[7] 彭新玉, 周扬, 董振江. 基于车联网远程驾驶的虚拟资源智能协同管理技术[J]. 电信科学, 2020, 36(4): 61-68.
PENG X Y, ZHOU Y, DONG Z J. Intelligent collaborative management technology of virtual resources based on Internet of vehicles remote driving[J]. Telecommunications Science, 2020, 36(4): 61-68.

[8] MUNIR A, HE T, RAGHAVENDRA R, et al. Network scheduling and compute resource aware task placement in datacenters[J]. IEEE/ACM Transactions on Networking, 2020, 28(6): 2435-2448.

[9] GODAVARTI M, MARZETTA T L, SHAMAI S. Capacity of a mobile multiple-antenna wireless link with isotropically random Rician fading[J]. IEEE Transactions on Information Theory, 2003, 49(12): 3330-3334.

[10] 李泽源. 移动边缘计算中计算卸载和资源分配协同优化策略设计与实现[D]. 北京: 北京交通大学, 2021.
LI Z Y. Design and implementation of cooperative optimization strategy for computing offload and resource allocation in mobile edge computing[D]. Beijing: Beijing Jiaotong University, 2021.

[11] CARDELLINI V, DE NITTO PERSONÉ V, DI VALERIO V, et al. A game-theoretic approach to computation offloading in mobile cloud computing[J]. Mathematical Programming, 2016, 157(2): 421-449.

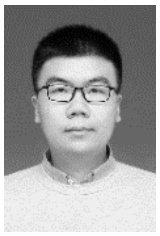
[12] WU M Y, YU F R, LIU P X, et al. A hybrid driving decision-making system integrating Markov logic networks and connectionist AI[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(3): 3514-3527.

[13] LI C L, ZHANG Y, LUO Y L. A federated learning-based edge caching approach for mobile edge computing-enabled intelligent connected vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(3): 3360-3369.

[14] KANG H Y, LI M L. High-dimensional data classification based on dimension reduction of BP neural network[J]. Computer Engineering and Applications, 49(20): 183-7.

[15] ZHANG C, ZHANG Z. Task migration for mobile edge computing using deep reinforcement learning[J]. Future Generation Computer Systems, 2019, 96: 111-118.

[作者简介]



张维庭（1992− ），男，博士，北京交通大学移动专用网络国家工程研究中心副教授、硕士生导师，主要研究方向为工业互联网、算力网络和网络智能。

孙呈蕙（1998− ），女，北京交通大学移动专用网络国家工程研究中心硕士生，主要研究方向为算力网络和边缘计算。

王洪超（1982− ），男，博士，北京交通大学移动专用网络国家工程研究中心副教授、硕士生导师，主要研究方向为信息网络。

代嘉宁（2000− ），男，北京交通大学移动专用网络国家工程研究中心硕士生，主要研究方向为算力网络和深度强化学习。