专栏：算力网络

基于策略约束强化学习的算网多目标优化研究

沈林江[1]，曹畅[2]，崔超[1]，张岩[2]
（1. 浪潮通信信息系统有限公司，山东 济南 250100；
2. 中国联合网络通信有限公司研究院，北京 100048）

摘　要：算力网络需要在满足用户业务需求的基础上最大化系统性能指标，现有方法主要通过多目标加权进行转换和求解，存在超参数难以确定、跨场景适用性差等问题。在分析算网目标特性的基础上，基于策略约束强化学习，将业务需求作为约束、系统性能指标作为优化目标，通过价值—策略—超参数的多级迭代策略，实现算网对用户业务需求的期望确定性保障和对系统性能的最优化。同时，研究了针对超参数寻优的多尺度步长（multi-scale step length，MSL）方法，进一步提升了系统的稳定性和准确性。仿真结果表明，所提方法在系统架构和负载变化情况下均具有良好的收敛性和稳定性。

关键词：算力网络；多目标优化；强化学习
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2023165

Research on constrained policy reinforcement learning based multi-objective optimization of computing power network

SHEN Linjiang[1], CAO Chang[2], CUI Chao[1], ZHANG Yan[2]
1. Inspur Communication Information System Co., Ltd., Jinan 250100, China
2. Research Institute of China United Network Communications Co., Ltd., Beijing 100048, China

Abstract: The computing power network needs to maximize the system performance index on the basis of meeting user business needs, and the existing methods are mainly based on the multi-objective weighting method, which has problems such as difficult to determine hyperparameters and poor cross-scenario applicability. Based on this, based on the analysis of the characteristics of the computing power network target, the user business requirements were taken as the policy constraints, and the performance indicators of the computing power network was taken as the optimization goal based on constrained policy optimization, and the expectation certainty of user business needs and the optimization of system performance through the value-strategy-hyper-parameter multi-level iterative strategy was realized. At the same time, the multi-scale step length (MSL) method for hyper-parameter optimization was studied, which further improved the stability and accuracy of the system. Simulation results show that the proposed method has good convergence and stability under the conditions of single terminal-single edge server, multi-terminal-multi-edge server and system load change.

Key words: computing power network, multi-objective optimization, reinforcement learning

收稿日期：2023−06−10；修回日期：2023−08−06

0　引言

算力网络通过整合算力、网络、存储、软件等底层基础资源，成为数字经济的新型基础设施[1-3]。与传统的中心云、边缘云资源独立运营模式不同，算力网络的服务对象包括算力提供者、消费者、运营者等，其服务目标也由满足用户业务需求指标进一步扩展到包括业务需求和系统性能在内的多个目标，包括用户侧的时延、成本、安全，以及系统侧的能耗、负载均衡、成本等[4]。算力网络需要采集算力、网络等资源数据和用户需求数据，对当前系统的服务质量、系统性能等进行动态评估，基于此构建决策和调度系统，从而保证系统稳定高效地运行[5]。

算力资源的异构性、架构的复杂性、业务需求的多变性，以及对能耗、成本的要求给算网决策和调度提出了较大的挑战[5-6]。近年来，基于动态控制理论[7]、强化学习[8]、图神经网络[9]等技术的智能决策系统，通过构建闭环优化能力，在边缘计算[10]、车联网[11]等场景下取得了较好的应用效果。文献[7]研究了基于李雅普诺夫优化算法，针对多用户终端—边缘服务器的边缘计算系统，通过超参数控制实现了系统能耗和时延的均衡和优化，但其依赖于对复杂系统的建模；文献[10]基于无模型的深度强化学习，实现了对边缘计算系统能耗和时延指标的联合优化，同时避免了复杂系统建模问题；文献[12]探讨了在算力网络场景下，通过深度强化学习构建编排调度策略，实现系统回报、成本和时延的均衡。上述方法通过加权形式将系统的多个服务目标整合为单个优化目标进行求解，然而根据加权因子的不同，多目标优化问题中往往存在多个最优解[13]。在不同系统架构、参数及业务目标下，需要对加权因子进行反复测试以获得一个相对合适的值，这导致较高的测试成本和有限的优化效果[14]。

从目标形态上看，用户业务指标和算网指标的表现形式不同，用户业务指标通常要求达到某个特定标准即可，而算网内部指标则需要做到尽可能优化，如尽可能将能耗降到最低[12]。从系统优化角度，可以将算网多目标优化转化为含约束的优化问题，其中约束条件为用户业务指标，包括时延、成本、质量、速率、吞吐量等，而优化目标为算网指标，包括能耗、负载均衡指标[7]。在边缘计算领域，文献[7]基于 KKT 条件（Karush-Kuhn-Tucker condition）将网络带宽约束增加到系统时延和能耗联合优化问题中，但其依赖于特定的网络传输模型，难以推广到其他约束条件下；文献[15]通过在深度 Q 网络（deep Q network，DQN）基础上增加约束网络，实现了对任务完成率的约束，但在一定程度上增加了系统的复杂性。在强化学习领域，对于含约束优化问题的求解，文献[14]讨论了一种基于策略约束的强化学习方法，通过构建价值—策略—控制参数的多级迭代策略，从而实现在满足约束前提下的目标最优化，文献[16]基于拉格朗日乘子法，实现了对含不等式约束的强化学习问题的求解方法，取得了较好的局部最优化性能；文献[17]采用信赖域的方法实现策略约束优化，在满足系统对安全性等要求前提下的最优化回报函数。

综上所述，针对算网多目标优化问题，当前缺少一种简洁、高效的方法，在满足业务需求的基础上最优化系统性能指标。基于此，本文在分析算网用户业务指标和系统性能指标的基础上，将算网服务目标转化为含约束的最优化问题，并采用深度强化学习方法进行求解。引入策略约束算法以解决对超参数的选择问题，同时设计多尺度步长优化方法，进一步提升系统的稳定性。通过仿真模型，验证了本文方法在系统参数和系统架构变化下的稳定性和收敛性。最后，对本文方法进行总结，并展望下一步的研究方向。

1 系统模型

本文研究的系统对象主要基于边缘计算模型，其中包括多个端节点、边缘服务器，其中终端与路由节点通过无线链路连接，边缘服务器与基站部署在同一位置。端节点接收用户发起的任务请求，可以选择在端侧本地处理或通过网络将其卸载到边缘服务器。该系统模型主要包括计算模型（包括端节点和边缘服务器节点）和网络模型，系统参数见表 1。

1.1 计算模型

考虑由 R 个边缘服务器（S_e,$1 \leqslant e \leqslant R$）和 L 个终端（M_l,$1 \leqslant l \leqslant L$）组成的边缘计算系统，终端在时间 τ 接收来自用户的计算任务 A，任务属性包括

表 1 系统参数

参数	参数含义	参数	参数含义
R	边缘服务器数量	k_l	终端能耗系数
S	边缘服务器	ε	小尺度衰落信道功率增益
L	终端数量	H	信道增益
M	终端	g_0	路径损耗常数
A	计算任务	θ	路径损耗指数
D	任务数据量	d_0	参考距离
C	任务计算量	N_0	噪声功率密度
σ	任务处理时限	d	终端到边缘服务器距离
Q_e	边缘服务器任务队列	p	终端发射功率
N_e	边缘服务器队列长度	w	总带宽
f_n	主频等分段数量	α	终端占用的带宽比例
f_e	边缘服务器频率	State	强化学习系统状态
T_e^w	边缘服务器队列等待时长	Action	强化学习行为空间
T_e^c	边缘服务器任务计算时长	I_l	终端任务是否按时完成
T_e^p	终端到边缘的传输时长	I_e	边缘服务器任务是否按时完成
T_e	边缘服务器任务总处理时长	I_p	任务卸载位置标志位
E_e	边缘服务器能耗	π	智能体的策略函数
E_e^p	终端到边缘服务器传输能耗	V_R^π	策略价值函数
k_e	边缘服务器能耗系数	γ	回报折扣因子
Q_l	终端任务队列	μ	系统初始状态分布
N_l	终端队列长度	J_R^π	系统优化目标函数
f_l	终端频率	J_R^c	系统约束目标函数
T_l^w	终端队列等待时长	ϵ	超时率阈值
T_l^c	终端队列任务计算时长	λ	约束权重因子
T_l	终端任务总处理时长	$\hat{V}_R^\pi$	折扣策略价值函数
E_l	终端能耗	η	迭代步长

其数据量 $D(D\in[0,D_{\max}])$、处理所需的计算量 $C(C\in(0,C_{\max}])$，以及任务要求的处理时限 σ，即 $A\triangleq\{D,C,\sigma\}$。本文认为粗粒度的任务拆分由上层调度系统完成，因此定义任务 A 为最小服务粒度，即任务 A 只能由某一终端或某一边缘服务器处理，而不能由多个计算节点协同处理。

以队列形式描述终端和边缘服务器中的计算任务。设边缘服务器 S_e 的任务队列 Q_e 中当前任务数量为 N_e，即 $Q_e=\{A_{e,1},A_{e,2},\cdots,A_{e,N_e}\}$，假设边缘服务器为单核服务器，定义其在处理第 i 个任务的运行频率为 $f_{e,i}(0<f_{e,i}\leqslant f_{e,\max})$，其中 $f_{e,\max}$ 为边缘服务器的最大工作频率，则第 i 个任务所需的计算时长为：

$$T_{e,i}^{c}=\frac{C_{e,i}}{f_{e,i}} \tag{1}$$

边缘服务器按照先进先出顺序处理队列中的计算任务，在时间 t 后，其队列可以表示为 $Q_e^t=\{A'_{e,j_e},A_{e,j_e+1},\cdots,A_{e,N_e}\}$，其中 $A'_{e,j_e}=\left\{D'_{e,j_e},C'_{e,j_e},\sigma'_{e,j_e}\right\}$ 为时间 t 后边缘服务器正在处理的任务所剩余的算力需求对应的任务，由于任务数据量不影响后续任务的处理时长等，为了表达简洁，本文令 $D'_{e,j_e}=D_{e,j_e}$，$\sigma'_{e,j_e}=\sigma_{e,j_e}$，则有：

$$j_e=\arg\max_i\left(\sum_{i=1}\frac{C_{e,i}}{f_{e,i}}\leqslant t\right)+1 \tag{2}$$

$$A'_{e,j_e}=\left\{D'_{e,j_e},C_{e,j_e}-\left(t-\sum_{i=1}^{j_e-1}\frac{C_{e,i}}{f_{e,i}}\right)f_{e,j_e},\sigma'_{e,j_e}\right\} \tag{3}$$

则当前队列中待处理任务数量为 $N'_e=N_e-j_e+1$。对于时间 t 卸载到该边缘服务器的计算任务 A_{e,N_e+1}，其在队列中的等待时长为：

$$T_{e,N_e+1}^{w}=\frac{C'_{e,j_e}}{f_{e,j_e}}+\sum_{i=j_e+1}^{N_e}\frac{C_{e,i}}{f_{e,i}} \tag{4}$$

则任务 A_{e,N_e+1} 卸载到边缘服务器后的总处理时长为：

$$T_{e,N_e+1}=T_{e,N_e+1}^{w}+T_{e,N_e+1}^{c} \tag{5}$$

服务器功率可近似与CPU运行频率的三次方成正比关系，设服务器能耗系数为 k_e，则在时间 t 内服务器能耗为：

$$E_e=\sum_{i=1}^{j_e-1}k_e f_{e,i}^2C_{e,i}+k_e f_{e,j_e}^2C'_{e,j_e} \tag{6}$$

同理，设终端 M_l 的任务队列 Q_l 中当前任务数量为 N_l，即 $Q_l=\{A_{l,1},A_{l,2},\cdots,A_{l,N_l}\}$，定义其在处理第 i 个任务的频率为 $f_{l,i}(0<f_{l,i}\leqslant f_{l,\max})$，能耗系数为 k_l，可以得到任务卸载在终端的情况下，任务 $A_{l,i}$ 的处理时长为：

$$T_{l,i}^{c}=\frac{C_{l,i}}{f_{l,i}} \tag{7}$$

在时间 t 后，其队列可以表示为：

$$Q_l^t=\{A'_{l,j_l},A_{l,j_l+1},\cdots,A_{l,N_l}\} \tag{8}$$

Q_l^t 长度为 $N'_l=N_l-j_l+1$，其中：

$$j_l=\arg\max_i\left(\sum_{i=1}\frac{C_{l,i}}{f_{l,i}}\leqslant t\right)+1 \tag{9}$$

$$A'_{l,j_l}=\left\{D'_{l,j_l},C_{l,j_l}-\left(t-\sum_{i=1}^{j_l-1}\frac{C_{l,i}}{f_{l,i}}\right)\cdot f_{l,j_l},\sigma'_{l,j_l}\right\} \tag{10}$$

则卸载到终端的任务 A_{l,N_l+1} 的处理总时长为：

$$T_{l,N_l+1}=T_{l,N_l+1}^{w}+T_{l,N_l+1}^{c} \tag{11}$$

其中：

$$T_{l,N_l+1}^{w}=\frac{C'_{l,j_l}}{f_{l,j_l}}+\sum_{i=j_l+1}^{N_l}\frac{C_{l,i}}{f_{l,i}} \tag{12}$$

在时间 t 内终端能耗为：

$$E_l=\sum_{i=1}^{j_l-1}k_l f_{l,i}^2C_{l,i}+k_l f_{l,j_l}^2C'_{l,j_l} \tag{13}$$

1.2　通信模型

为了将部分算力任务卸载到边缘服务器，终端需要将部分数据传输到边缘服务器，设从终端 M_l 到边缘服务器 S_e 的小尺度衰落信道功率增益为 $\varepsilon_{l,e}$，且 $E\left[\varepsilon_{l,e}\right]<\infty$，则信道增益可以表示为 $H_{l,e}=\varepsilon_{l,e}g_0(d_0/d_{l,e})^{\theta}$，其中，$g_0$ 表示路径损耗常数，θ 表示路径损耗指数，d_0 为参考距离，$d_{l,e}$ 为终端到边缘服务器的距离。根据香农定理，在

时隙 t 内，终端到边缘服务器的传输带宽为：

$$W_{l,e}=\begin{cases}\alpha_{l,e}w\mathrm{lb}\left(1+\dfrac{H_{l,e}p_{l,e}}{\alpha_{l,e}N_0w}\right),\alpha_{l,e}>0\\0,\alpha_{l,e}=0\end{cases} \quad (14)$$

其中，$p_{l,e}$ 表示终端 M_l 的发射功率，N_0 代表噪声功率密度，w 表示总带宽，$\alpha_{l,e}$ 表示终端 M_l 所占用的带宽比例，对于任意边缘服务器 S_e，其带宽分配满足 $\sum_{l=1}^{L}\alpha_{l,e}\leqslant 1$。则任务 A_{e,N_e+1} 由终端到边缘服务器的传输能耗为：

$$E^p_{e,N_e+1}=p_{l,e}T^p_{e,N_e+1} \quad (15)$$

其中，T^p_{e,N_e+1} 表示数据从终端侧传输到边缘服务器所需的时长：

$$T^p_{e,N_e+1}=\frac{D_{e,N_e+1}}{W_{l,e}} \quad (16)$$

2 求解方法

系统架构如图 1 所示，终端接收算力需求任务后，系统需要结合网络信息、终端和边缘服务器的负载信息等，决定将相关需求卸载到终端或者边缘服务器处理，满足业务发起者对时延等指标的要求。另外，从算网系统优化的角度，需要对边缘服务器和端侧的频率等控制参数进行优化，以实现能耗等系统性能指标的最优化。

为实现上述目标，本文首先通过深度强化学习算法，在边缘计算系统外构建决策优化智能体，实现对任务的卸载及对算力资源的控制。然后在保障业务需求的基础上，进一步研究基于策略约束强化学习技术，从而优化算网资源分配等，实现最大化算网系统性能。

2.1 深度强化学习

图 1 中，决策优化智能体在边缘计算系统收到服务请求后，将感知系统状态作为决策输入。设当前任务 A_t 来自终端 M_l，则决策优化智能体感知信息包括：

$$\mathrm{State}=\{A_t,\alpha_{l,e},N_l,T^w_l,f_{l,\max},N_e,T^w_e,f_{e,\max}\}(1\leqslant e\leqslant R) \quad (17)$$

其中，A_t 为当前任务信息，$\alpha_{l,e}$ 为终端到各个边缘服务器的带宽信息，N_l、T^w_l、$f_{l,\max}$ 分别为当前终端的队列长度、等待时长及最大频率信息，N_e、T^w_e、$f_{e,\max}$ 分别为各个边缘服务器的队列长度、等待时长及最大频率信息。

决策优化智能体在每次接收任务后进行一次最优动作决策和执行。智能体动作包括决定任务的卸载位置（终端或某一边缘服务器）以及确定相应终端或边缘服务器处理任务 A_t 的频率。为了将频率与智能体的决策空间对应，本文将终端或边缘服务器主频等分为 f_n 段，则智能体决策空间

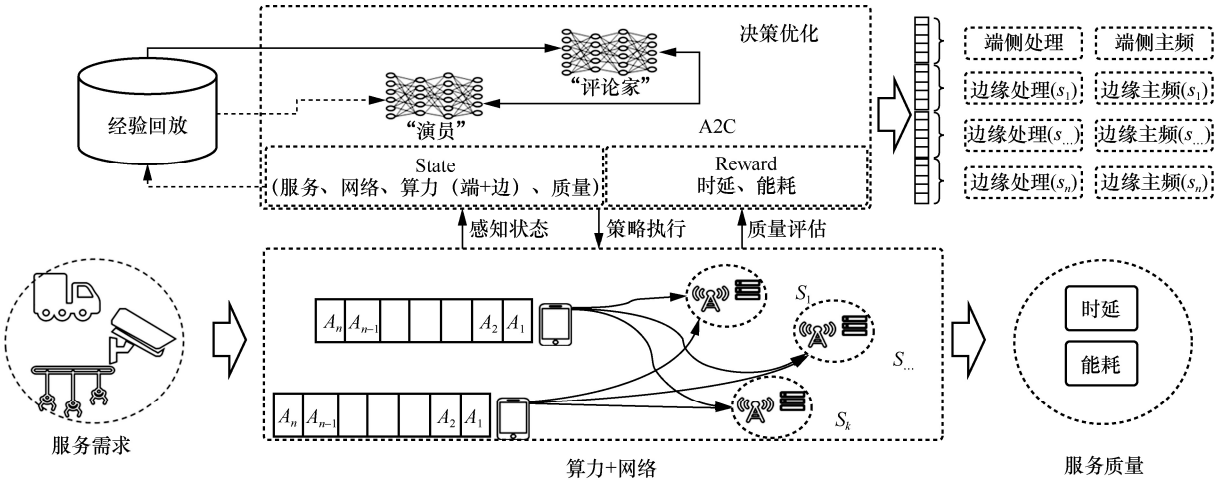


图 1 系统架构

大小为 $f_n(R+1)$，可以表示为：

$$\begin{aligned}\text{Action}=\{&a_{l,1},a_{l,2},\cdots,a_{l,f_n},a_{e_1,1},a_{e_1,2},\cdots,\\&a_{e_1,f_n},a_{e_2,1},a_{e_2,2},\cdots,a_{e_2,f_n},\cdots,a_{e_R,1},a_{e_R,2},\cdots,a_{e_R,f_n}\}\end{aligned}\tag{18}$$

其中，$a_{l,i}$ 表示将任务需求卸载到本地终端，并设置终端频率为 $if_{l,\max}/f_n$，$a_{e,i}$ 表示将任务需求卸载到第 e 个边缘服务器，并设置边缘服务器频率为 $if_{e,\max}/f_n$。

在两个任务到达的时间窗内，智能体优化目标为在保证业务需求按时完成率（时延）的前提下，降低系统能耗。对于卸载到终端的任务仅考虑其计算时延，以 I 终端队列中任务是否能够按时完成，即：

$$I_{l,i}=\begin{cases}0,T_{l,i}\leqslant\sigma_{l,i}\\1,T_{l,i}>\sigma_{l,i}\end{cases}\tag{19}$$

同理，对于卸载到边缘服务器的任务，其时延包括计算时延和传输时延，即：

$$I_{e,i}=\begin{cases}0,T_{e,i}+T_{e,i}^{p}\leqslant\sigma_{e,i}\\1,T_{e,i}+T_{e,i}^{p}>\sigma_{e,i}\end{cases}\tag{20}$$

则任务超时率可以表示为：

$$r_c=\frac{\sum_{e=1}^{R}\left(\sum_{i=1}^{N_e}I_{e,i}+I_{p,e}I_{e,N_e+1}\right)+\sum_{l=1}^{L}\left(\sum_{i=1}^{N_l}I_{l,i}+I_{p,l}I_{l,N_l+1}\right)}{\sum_{e=1}^{R}(N_e)+\sum_{l=1}^{L}(N_l)+1}\tag{21}$$

在两个任务到达的时间窗内，系统能耗包括端侧和边缘侧处理计算的任务能耗，以及从端侧向边缘服务器传输的能耗，即：

$$\begin{aligned}r_e=&p_{l,e}T_{e,N_e+1}^{p}I_{p,e}+\sum_{i=1}^{j_e-1}k_e f_{e,i}^{2}C_{e,i}+\\&k_e f_{e,j_e}^{2}C'_{e,j_e}+\sum_{i=1}^{j_l-1}k_l f_{l,i}^{2}C_{l,i}+k_l f_{l,j_l}^{2}C'_{l,j_l}\end{aligned}\tag{22}$$

其中，$I_{p,e}$ 和 $I_{p,l}$ 为任务卸载位置的标志位，当任务由终端卸载到第 e 个边缘服务器时，$I_{p,e}=1$，$I_{p,l}=0$，当任务卸载到第 l 终端本地进行处理时，$I_{p,e}=0$，$I_{p,l}=1$。

2.2　策略约束强化学习

分析业务需求指标与算网指标的表征区别。对于业务需求指标，用户通常会提出明确的阈值，例如，在时延小于 3 ms 的情况下，任务的平均超时率低于 2%，边缘计算系统只要满足用户的业务指标需求即可，而不是将相关业务指标降到最低。而对于算网系统性能指标，如能耗、负载均衡、成本等，通常没有明确的阈值需求，对于该类指标，边缘计算系统需要在满足业务指标的基础上尽可能进行优化，如将能耗或者成本降到最低等。因此，边缘计算系统优化的目标可以表述为，在满足用户业务需求指标的前提下，尽可能优化算网自身指标，进而将其转化为含约束的最优化问题，其中约束为用户业务需求指标，而最优化的目标为算网系统性能指标。

本文采用回报约束策略优化（reward constrained policy optimization，RCPO）[14]对上述边缘计算系统的任务卸载策略进行优化。设智能体优化的目标为降低系统整体能耗，对于给定状态，设智能体遵循的策略为 π，智能体以最优化系统能耗为目标，则策略价值函数可以表示为：

$$V_R^{\pi}=E\left[\sum_t-\gamma^t r_e\,|\,\text{State}=\text{State}_t\right]\tag{23}$$

其中，γ 表示折扣因子，则在给定初始状态 μ 分布情况下，强化学习优化目标为 $\max_{\pi}J^{\pi}$，其中：

$$J^{\pi}=E_{\text{State}\sim\mu}^{\pi}\left[\sum_t-\gamma^t r_e\right]=\sum\mu(\text{State})V_R^{\pi}\tag{24}$$

将任务超时率 r_c 作为策略约束强化学习的约束条件，即：

$$J_C^{\pi}=E_{\text{State}\sim\mu}^{\pi}\left[r_c\right]\tag{25}$$

为了在保障任务超时率的基础上，最大化算网能耗需求，设业务要求阈值（如超时率）小于 ϵ，含约束的强化学习的优化目标可以表示为：

$$\max_{\pi}J^{\pi},\text{s.t.}J_C^{\pi}\leqslant\epsilon\tag{26}$$

基于拉格朗日乘子法，上述约束计算式可以转化为无约束问题，即：

$$\min_{\lambda}\max_{\pi}\left[J^{\pi}-\lambda(J_C^{\pi}-\epsilon)\right]\tag{27}$$

其中，λ为约束权重因子，λ更新表达式为：

$$\lambda_{k+1}=\lambda_k-\eta_{\lambda,k}\nabla_\lambda(J^\pi-\lambda(J_C^\pi-\epsilon)) \tag{28}$$

其中，η表示约束权重因子迭代步长。在实际应用中，基于拉格朗日乘子法求解含约束的强化学习问题存在对约束条件的适用性差等问题，基于策略约束的强化学习方法引入折扣约束对上述问题进行求解，其中折扣约束可以表示为：

$$V_C^\pi=E\left[\sum_t\gamma^t r_c\,|\,\text{State}=\text{State}_t\right] \tag{29}$$

将折扣约束与回报函数按照加权系数相加，则折扣回报可以表示为：

$$\hat{r}\triangleq -r_e-\lambda r_c \tag{30}$$

折扣策略价值函数可以表示为：

$$\hat{V}_R^\pi=E\left[\sum_t\gamma^t(-r_e-\lambda r_c)\,|\,\text{State}=\text{State}_t\right]=V_R^\pi-\lambda V_C^\pi \tag{31}$$

当权重因子λ与策略π固定时，$\hat{V}_R^\pi$可以通过时序差分法进行求解。λ在强化学习完成一个周期的迭代后，按照式（28）进行更新。

上述算法流程通过寻找最优的λ以实现满足业务约束条件前提下的系统性能最优，在实际应用中，J_C^π与λ存在较强的非线性关系，具体体现在当J_C^π接近ϵ时，J_C^π随λ变化发生剧烈变化。传统采用固定步长方法可能导致λ迭代步长不合理，导致无法实现对式（27）的最优化或者搜索结果无法满足约束条件。基于此，本文根据约束条件满足情况，针对步长η_λ设计一种多尺度步长策略，即在第k次迭代中，当$\nabla_{\lambda,k-1}\nabla_{\lambda,k}<0$，则令$\eta_\lambda=\eta_\lambda/2$、$\lambda_k=\lambda_{k-1}$，当$\eta_\lambda$或者$\text{abs}(\text{d}\lambda_k)$小于给定阈值，即中断算法流程，认为任务查找到较为准确的λ。

基于A2C（advantage actor-critic）算法的策略约束强化学习算法流程如下。

算法1 基于A2C算法的策略约束强化学习算法流程

1. **输入** 业务要求阈值ϵ，初始学习率η_λ、η_{θ_π}、η_{θ_V}，折扣率γ，终止阈值th_η、th_c

2. **初始化** 策略网络权重θ_π、价值网络权重θ_V，权重系数λ，迭代周期$t=0$

3. for $T=1,2,\cdots,T_{\max}$：

4. 初始化梯度，$\text{d}\theta_\pi=0,\text{d}\theta_v=0,\text{d}\lambda=0$

$t_{\text{start}}=t$

5. while State_t非终止状态 and $t-t_{\text{start}}<t_{\max}$：

a. 获取当前动作：$\text{Action}_t\sim\pi(\text{Action}_t\,|\,\text{State}_t,\theta_\pi)$：

b. 获取当前奖励、下一步状态及约束值：$r_{e,t}$、State_{t+1}、$r_{c,t}$

c. $t=t+1$

d. $\text{Reward}=\begin{cases}0,\text{State}_t\text{为终止状态}\\ \hat{V}_R^\pi(\text{State}_t,\theta_V),\ \text{其他}\end{cases}$

6. for $\tau=t-1,t-2,\cdots,t_{\text{start}}$：

a. $\text{Reward}=r_{e,\tau}-\lambda r_{c,\tau}+\gamma\text{Reward}$

b. $\text{d}\theta_\pi=\text{d}\theta_\pi+\nabla_{\theta_\pi}\log\pi\left(\text{Action}_\tau|\text{State}_\tau,\theta_\pi\right)\left(\text{Reward}-\hat{V}_R^\pi\left(\text{State}_\tau,\theta_V\right)\right)$

c. $\text{d}\theta_V=\text{d}\theta_V+\partial\left(\text{Reward}-\hat{V}_R^\pi\left(\text{State}_\tau,\theta_V\right)\right)^2/\partial\theta_V$

7. if State_t为终止状态：

a. $\text{d}\lambda_{T-1}=\text{d}\lambda$

b. $\text{d}\lambda=-(J_C^\pi-\epsilon)$

c. $t=0$

d. 初始化State_0

8. if $\text{d}\lambda_{T-1}\text{d}\lambda<0$：

a. $\eta_\lambda=\eta_\lambda/2$

b. $\lambda=\lambda_{T-1}$

9. if $\eta_\lambda<\text{th}_\eta$ or $\text{abs}(\text{d}\lambda)<\text{th}_\eta$：

a. break

10. 更新策略网络权重θ_π、价值网络权重θ_V、权重系数λ，$\lambda_{T-1}=\lambda$

对算法1流程的解释如下：算法1的流程5、流程6为标准的A2C算法流程，在权重系数λ固定情况下，优化目标分别为折扣策略价值函数$\hat{V}_R^\pi$

和策略函数 π。算法 1 的流程 7 中，当完成一个完整的训练流程，即 State_t 为终止状态时，评估 J_C^{π} 与阈值 ϵ 的差异（流程 7.a），并以此为梯度更新约束权重因子 λ，从而实现系统在一定周期内能够满足业务对超时率要求的基础上，最优化系统性能。除此之外，由于式（25）表示对约束的期望，因此，上述算法 1 能够保证约束的平均值满足要求，而无法保证系统在任意时刻满足约束条件。

3　仿真与分析

3.1　实验设置

为了验证本文方法的有效性，设计如图 1 所示的边缘计算系统，其中包含 R 个边缘服务器及 L 个终端。模拟仿真系统参数见表 2，其中相关网络参数（包括 $\varepsilon_{l,e}$、w、N_0、g_0、θ、d_0 等）及算力主频、能耗系数、端侧发射功率等参考当前已有边缘计算相关文献[7]，为了降低强化学习动作空间维度，将终端和边缘算力主频等分为 10 份，即 $f_n=10$。

表 2　模拟仿真系统参数

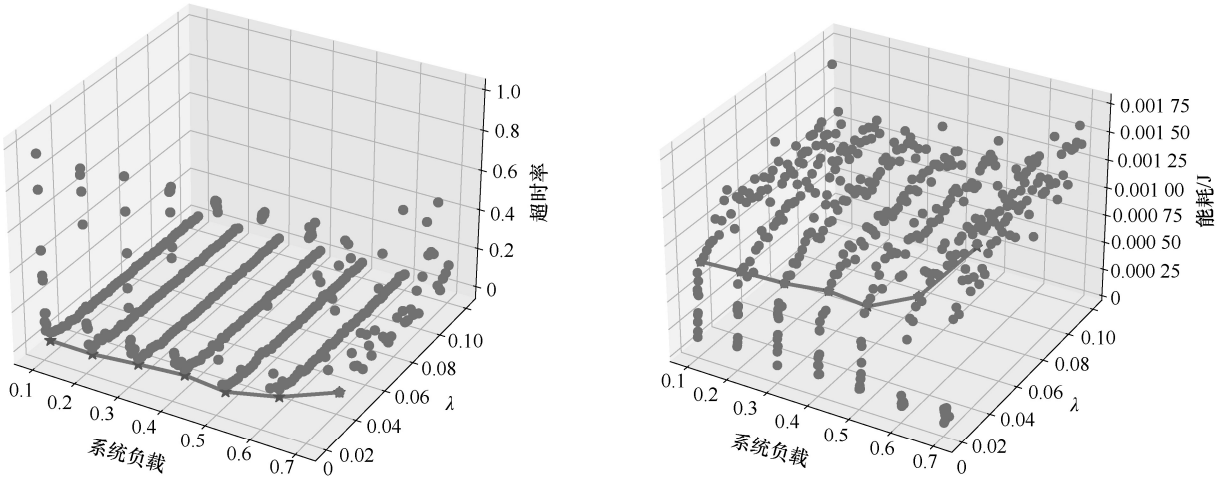
参数	取值范围
$\varepsilon_{l,e}$	Exp(l)
w	10 MHz
N_0	–174 dBm/Hz
g_0	–40 dB
θ	4
d_0	1 m
$d_{l,e}$	[90,150] m
$p_{l,e}$	500 mW
k_l	10^{-27}
k_e	10^{-27}
D	[0,50 000] bit
C	[30 000,50 000] cycle（CPU 转数）
$f_{e,\max}$	2.5 GHz
$f_{l,\max}$	1 GHz
f_n	10

3.2　实验结果与分析

首先分析单终端单边缘服务器系统下（$R=1$、$L=1$），超时率和能耗随系统负载和权重系数 λ 的关系，其中任务时延要求为在 0.3～0.7 ms 内均匀分布。其中系统负载与整个边缘计算系统在单位时间内接收的任务的时间间隔成反比，即单位时间内系统接收的任务数量越多，系统负载越高。

超时率和能耗随权重系数 λ 的变化趋势如图 2 所示，为算法 1 所示的一个训练周期内的平均能耗与平均超时率随系统负载和权重系数 λ 的关系，其中灰色实线为满足超时率阈值 $\epsilon=0.2$ 时的最小 λ。由图 2 可见，在系统负载固定情况下，系统超时率随着 λ 的增加而降低，而能耗随着 λ 的增加增高，说明随着 λ 的增大，智能体更加关注优化超时率指标，从而导致能耗增高。由此可见，系统能耗和超时率目标之间的博弈关系。在给定系统负载和超时率需求情况下，通过对 λ 的寻优，能够在满足超时率指标的基础上得到最优的系统能耗指标。除此之外，灰色实线表明，随着系统负载的增高，需要提升 λ 以保证超时率的稳定性（如图 2（a）所示），在一定程度上说明了 λ 需要随系统负载、参数、解构等动态调整，同时，λ 增高会导致能耗的增高，反映了系统能耗和系统负载的正比关系。另外，系统超时率、系统能耗随着 λ 的增加存在快速增加或降低的过程（在 $\lambda<0.02$ 范围内），在一定程度上证实了本文采用多尺度步长方法的合理性。

单终端—单边缘服务器情况下的系统超时率、能耗、总回报、终端频率和边缘服务器频率随权重系数 λ 和训练迭代次数的变化趋势如图 3 所示，其中任务时延要求为 0.3～0.7 ms，系统负载为 0.5。整体而言，随着强化学习训练迭代，系统超时率、能耗水平趋于稳定，而系统的总回报（图 3（c）、图 3（h）、图 3（m））在前 1 000 次快速增加且趋于稳定。随着 λ 降低，系统超时率



(a) 超时率随权重系数λ的变化趋势　　(b) 能耗随权重系数λ的变化趋势

图 2　超时率和能耗随权重系数 λ 的变化趋势

逐渐增高（图 3（a）、图 3（f）、图 3（k））、能耗逐渐降低（图 3（b）、图 3（g）、图 3（l）），与图(2)呈现的趋势相符。观察终端频率(图 3(d)、图 3（i）、图 3（n））和边缘服务器频率（图 3（e）、图 3（j）、图 3（o））可见，当λ较大（$\lambda=0.005$）时，系统更加注重降低超时率，因此算力任务多数被从边缘服务器上卸载，并且边缘服务器以较高频率（约为 1.4 GHz）运行，任务超时率小于 0.01；随着λ降低（$\lambda=0.002$），系统逐渐趋于降低能耗，尽管该过程中算力任务仍被卸载于边缘服务器，边缘服务器频率有了明显降低（约为 1.1 GHz），任务超时率约为 0.2；随着λ进一步降低（$\lambda=0.000\,5$时），系统更加注重能耗而忽略超时率影响，因此大量算力任务被卸载到终端本地进行处理，此时系统超时率接近 1。上述分析说明，本文采用的深度学习算法，能够随着系统权重系数λ的变化，自动决定将算力任务卸载到边缘侧和终端本地处理，同时对卸载对象的频率进行控制，从而实现能耗和超时率的均衡。

单终端—单边缘服务器情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 8）如图 4 所示，其中任务时延要求为 0.3～0.7 ms，系统负载为 0.5，超时率阈值$\epsilon=0.2$。结合多尺度步长迭代策略约束强化学习搜索所获得的最优λ=0.001 8。随着训练迭代，系统的整体超时率稳定在 0.2 左右，搜索结果与图 2 和图 4 呈现的趋势相符。值得注意的是，如针对算法 1 的分析，λ的优化体现在保障平均超时率，而不是各个迭代步骤的超时率（图 4（a)），因此存在部分时刻超时率大于ϵ。由于强化学习所具备的根据环境状态自适应调节能力，因此系统通过调节边缘服务器频率能够稳定到ϵ，反映了本文方法整体的稳定性。

单终端—单边缘服务器情况下的策略约束强化学习超时率、能耗随λ的变化趋势如图 5 所示，为系统平均能耗和平均超时率随λ变化趋势。由图 5 可见，在λ由 0.1 逐步收敛到 0.001 8 的过程中，超时率和能耗最初变化较为平缓（$0.02<\lambda<0.1$），随着λ的降低，超时率迅速增加，而能耗迅速降低，由于本文采用多尺度步长的迭代策略，策略约束强化学习能够在约束接近ϵ时逐渐降低迭代步长，从而保证较高的迭代效率和搜索精度。

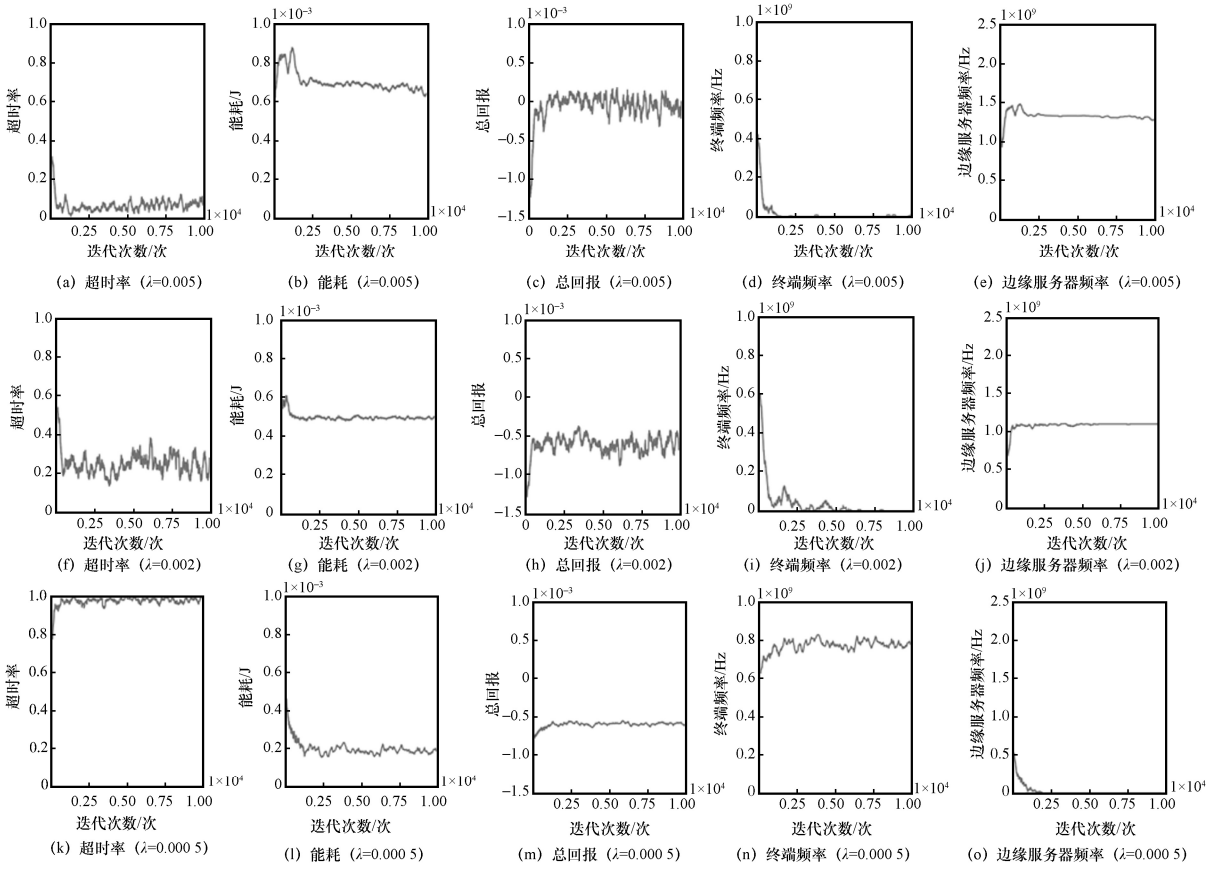


图 3　单终端—单边缘服务器情况下的系统超时率、能耗、总回报、终端频率和边缘服务器频率随权重系数 λ 和训练迭代次数的变化趋势

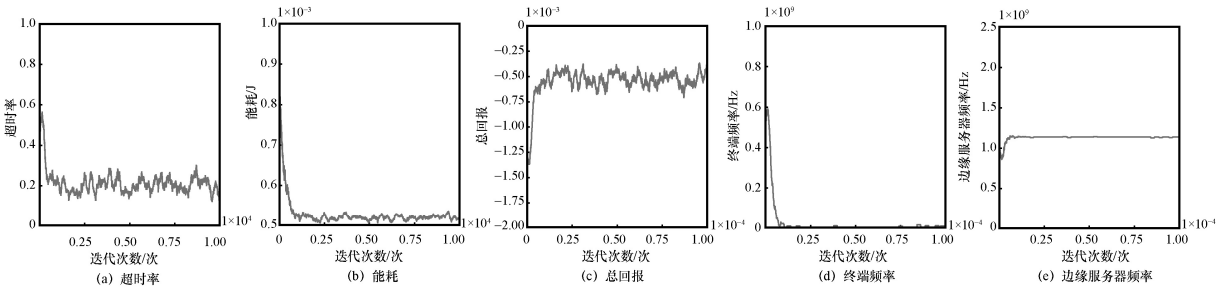


图 4　单终端—单边缘服务器情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 8 时）

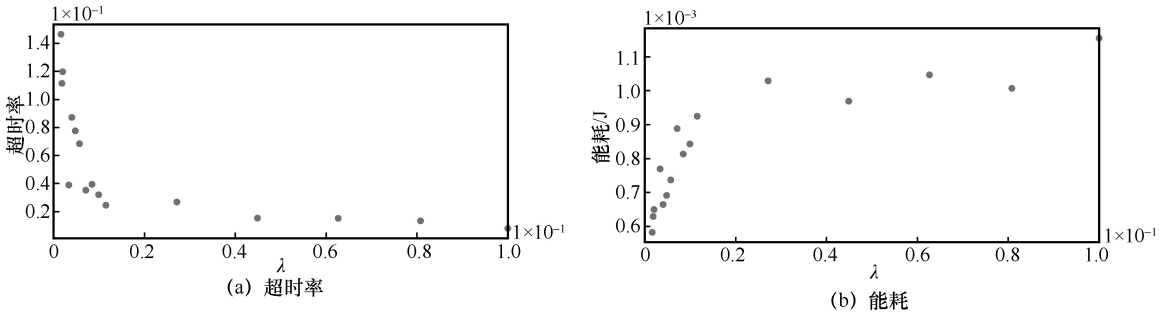


图 5　单终端—单边缘服务器情况下的策略约束强化学习超时率、能耗随 λ 的变化趋势

为了验证本文方法对系统负载的稳定性，单终端—单边缘服务器情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势如图 6 所示，其中任务时延要求为 0.24～0.56 ms，系统负载为 0.2，ϵ=0.2。结合多尺度步长迭代策略约束强化学习搜索所获得的最优 λ=0.002 1。由此可见，在系统负载发生变化的情况下，策略约束的强化学习能够在有限次（次数<2 500）迭代下趋于稳定，且系统超时率能够稳定在 0.2 左右。

为了验证本文方法对于跨系统架构的适用性，多终端（2 个）—多边缘服务器（2 个）情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 78，系统负载=0.75，任务时延要求为 0.3～0.7 ms 时）如图 7 所示。多终端（3 个）—多边缘服务器（5 个）情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 12，系统负载=0.75，任务时延要求为 0.3～0.7 ms 时）如图 8 所示。首先，在多终端多边缘服务器系统下，含策略约束的强化学习均具有较好的收敛性，系统超时率稳定在 0.2 左右（如图 7（a）、图 8（a）所示）。除此之外，由图 7 和图 8 可见，策略约束强化学习在多终端—多边缘服务器系统下的稳定性有所降低，超时率和能

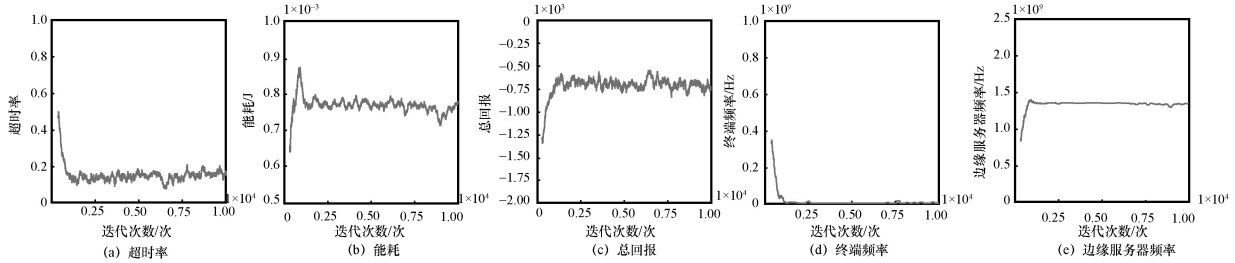


图 6　单终端—单边缘服务器情况下的策略约束强化学习超时、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.002 1，系统负载=0.2，任务时延要求为 0.24～0.56 ms 时）

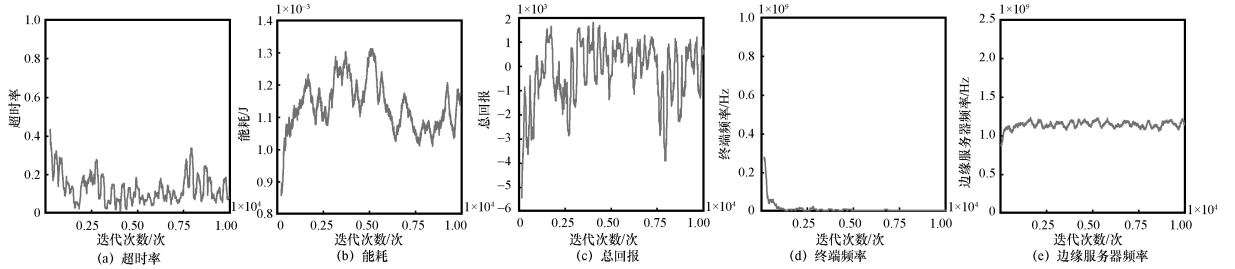


图 7　多终端（2 个）—多边缘服务器（2 个）情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 78，系统负载=0.75，任务时延要求为 0.3～0.7 ms 时）

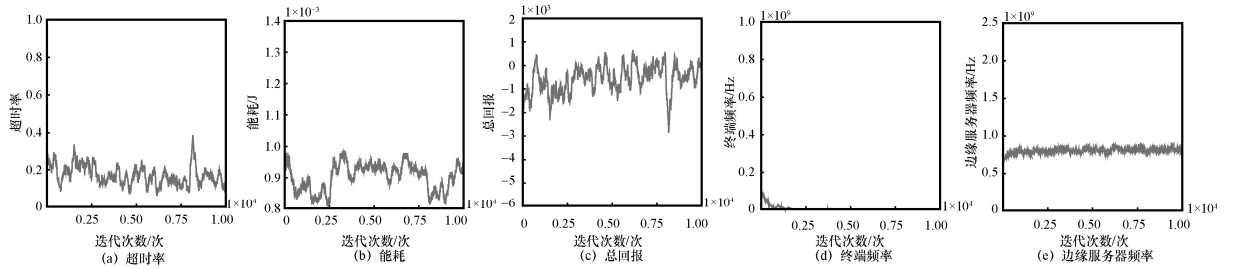


图 8　多终端（3 个）—多边缘服务器（5 个）情况下的策略约束强化学习超时率、能耗、总回报、终端频率和边缘服务器频率随训练迭代次数的变化趋势（λ=0.001 12，系统负载=0.75，任务时延要求为 0.3～0.7 ms 时）

耗的波动性明显增加，可能归因于多终端—多边缘服务器的状态更加复杂，且当前强化学习采集的系统状态数据不足，导致模型性能降低。

4　结束语

以边缘计算系统为代表的算力网络需要在满足用户业务需求的基础上最大化能耗等算网系统性能指标。本文将算网服务抽象为多目标优化问题，针对多目标优化中的约束参数问题，基于策略约束强化学习方法，通过状态空间、行为空间、回报函数设计以及迭代步长优化等进行问题求解。理论分析和仿真模型验证了本文方法理论的正确性，以及在系统负载变化和架构变化的快速收敛和稳定性。

如求解方法部分（第 2 节）所述，本文方法将用户业务需求转化为强化学习的约束条件，所求得的解能够满足用户业务约束期望的稳定性，但难以保证在所有时刻均能满足用户业务需求。针对上述问题，笔者认为：一方面，可以通过设计新的约束方法，保证所有阶段均能满足用户需求；另一方面，可以通过增加系统状态指标、设计探索空间或动作的安全强化学习算法等，降低用户业务需求指标的波动性或方差，以达到对用户业务需求的保证。该部分工作将是笔者下一步的研究内容之一。

算力网络所面临的用户需求复杂，可能存在多个业务约束条件，同时需要满足内部能耗、成本等多个优化指标，可以将其抽象为含有多个约束条件下的多目标优化问题。在多约束、多目标场景下，可以探索通过扩展拉格朗日乘子法、控制参数的嵌套寻优等方法来实现，但同时可能面临求解过程复杂、参数寻优困难等问题，探索多个约束条件下的多目标优化问题的高效求解方法，将是笔者下一步的研究内容之一。

参考文献：

[1] TANG X Y, CAO C, WANG Y X, et al. Computing power network: the architecture of convergence of computing and networking towards 6G requirement[J]. China Communications, 2021, 18(2): 175-185.

[2] 雷波, 赵倩颖, 赵慧玲. 边缘计算与算力网络综述[J]. 中兴通讯技术, 2021, 27(3): 3-6.
LEI B, ZHAO Q Y, ZHAO H L. Overview of edge computing and computing power network[J]. ZTE Technology Journal, 2021, 27(3): 3-6.

[3] 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.

[4] 李建飞, 曹畅, 李奥, 等. 算力网络中面向业务体验的算力建模[J]. 中兴通讯技术, 2020, 26(5): 34-38, 52.
LI J F, CAO C, LI A, et al. Computing power modeling for business experience in computing power network[J]. ZTE Technology Journal, 2020, 26(5): 34-38, 52.

[5] 何涛, 杨振东, 曹畅, 等. 算力网络发展中的若干关键技术问题分析[J]. 电信科学, 2022, 38(6): 62-70.
HE T, YANG Z D, CAO C, et al. Analysis of some key technical problems in the development of computing power network[J]. Telecommunications Science, 2022, 38(6): 62-70.

[6] KHAN W Z, AHMED E, HAKAK S, et al. Edge computing: a survey[J]. Future Generation Computer Systems, 2019, 97(C): 219-235.

[7] MAO Y Y, ZHANG J, SONG S H, et al. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems[J]. IEEE Transactions on Wireless Communications, 2017, 16(9): 5994-6009.

[8] MOUSAVI S S, SCHUKAT M, HOWLEY E. Deep reinforcement learning: an overview[C]//Proceedings of SAI Intelligent Systems Conference (IntelliSys). Heidelberg: Springer, 2016: 426–440

[9] LI Y, ZHANG X, ZENG T, et al. Task placement and resource allocation for edge machine learning: a GNN-based multi-agent reinforcement learning paradigm[J]. arXiv preprint, 2023, arXiv: 2302.00571.

[10] ALE L H, ZHANG N, FANG X J, et al. Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(3): 881-892.

[11] LI M S, GAO J, ZHAO L, et al. Deep reinforcement learning for collaborative edge computing in vehicular networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(4): 1122-1135.

[12] YANG A, WU M, CHENG B, et al. Reinforcement learning in computing and network convergence orchestration[J]. arXiv preprint, 2022, arXiv:2209.10753.

[13] JAIN T, AVANEESH, VERMA R, et al. Latency-memory optimized splitting of convolution neural networks for resource constrained edge devices[C]//Proceedings of 2022 14th International Conference on Communication Systems & Networks

(COMSNETS). Piscataway: IEEE Press, 2022: 531-539.
[14] TESSLER C, MANKOWITZ D J, MANNOR S. Reward constrained policy optimization[J]. arXiv preprint, 2018, arXiv: 1805.11074.
[15] ZHUANG S, GAO C X, HE Y, et al. QC-DQN: a novel constrained reinforcement learning method for computation offloading in multi-access edge computing[C]//Proceedings of 2022 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE Press, 2022: 1-8.
[16] BHATNAGAR S, LAKSHMANAN K. An online actor-critic algorithm with function approximation for constrained Markov decision processes[J]. Journal of Optimization Theory and Applications, 2012, 153(3): 688-708.
[17] ACHIAM J, HELD D, TAMAR A, et al. Constrained policy optimization[J]. arXiv preprint, 2017, arXiv:1705.10528.

[作者简介]

沈林江（1981− ），男，浪潮通信信息系统有限公司副总经理、算力网络研究院院长，主要从事算力网络相关前沿理论分析、技术研究和产品设计等工作。

曹畅（1984− ），男，博士，中国联合网络通信有限公司研究院未来网络研究部总监、高级工程师，主要从事算力网络、IPv6+网络新技术、未来网络体系架构等研究工作。

崔超（1993− ），男，现就职于浪潮通信信息系统有限公司，主要从事算力网络、AI算法等相关研究工作。

张岩（1983− ），男，博士，中国联合网络通信有限公司研究院未来网络研究部主任研究员、高级工程师，主要从事算力网络、云网融合/云计算、未来网络体系架构等研究工作。