基于人工智能 RL 算法的边缘服务器智能选择模式研究

李小婧[1*]，杨冬冬[1]，韩润东[2]，俞　华[1]，阴崇智[1]

(1. 国网山西省电力公司电力科学研究院　太原　030001；2. 国网山西省电力公司设备部　太原　030001)

【摘要】提出了一种基于人工智能深度强化学习算法的扩展性及智能性较高的智能选择模式。在人工智能深度强化学习算法的基础上，引入了动作抑制、四重 Q 学习 (QQL) 及归一化 Q-value 等机制，研究并实现了在满足业务延迟要求及公平性的原则下，物联终端更智能地选择其接入或切换边缘服务器。该方案减少了业务延迟，提高了响应效率，有助于提高业务安全及运营管理水平。

关　键　词　人工智能；　云边端；　深度强化学习算法；　边缘服务器；　配电站房运检

中图分类号　TM18　　**文献标志码**　A　　doi:10.12178/1001-0548.2022119

Research on Intelligent Selection Mode of Edge Server Based on Artificial Intelligence Deep Reinforcement Learning Algorithm

LI Xiaojing[1*], YANG Dongdong[1], HAN Rundong[2], YU Hua[1], and YIN Chongzhi[1]

(1. Electric Power Research Institute of State Grid Shanxi Electric Power Company　Taiyuan　030001;
2. Equipment Department of State Grid Shanxi Electric Power Company　Taiyuan　030001)

Abstract　Based on the artificial intelligence deep reinforcement learning algorithm, this paper proposes an intelligent selection mode with high fairness, expansibility and intelligence. On the basis of the artificial intelligence deep reinforcement learning algorithm, innovative mechanisms such as action inhibition, quadruple Q-learning (QQL) and normalized Q-value are introduced. With the research results of this paper, the IoT (Internet of Thing) terminal can more intelligently select its access or handover edge server under the principle of meeting the service delay requirements and fairness. This scheme reduces service delay, improves service response efficiency, and has good value significance for improving service security and operation management level.

Key words　artificial intelligence;　cloud edge end;　deep reinforcement learning algorithm;　edge server;　operation inspection of distribution station building

近年来，人工智能和物联网技术得到飞速发展，物联终端接入数量呈现几何级数增长，相应的物联网相关系统仍以终端感知数据采集及分析处理为主，信息化系统由传统的 C/S、B/S 架构，逐渐演进为云端系统结合边缘代理服务器的云−边−端架构模式，以提高业务处理前置化能力[1]。相对传统技术架构而言，云−边−端架构中，部署边缘服务器更有利于大量终端设备的接入，以及结合运检业务进行本地计算和处理，降低业务对云端后台系统的依赖性，因为边缘服务器在物理上比传统云服务器更靠近现场和客户，同时可提供更低的网络延迟[2]。

由于边缘服务器在计算、存储和通信资源等方面存在一定限制，部署服务有限。当海量终端接入后，通讯及业务处理并发量大时，存在着由于数据计算和业务处理需排队导致堵塞延时的问题，甚至宕机等风险[3]。由此提出在现场部署多个边缘服务器或在附近区域租用云服务器做边缘处理，构建边缘服务器集群。由于现场并发通讯的终端数量较多，如何选择最佳边缘服务器进行计算和业务处理是研究的主要问题。以电网某配电站房运检业务为

收稿日期：2022－04－26；修回日期：2022－08－30
基金项目：国网山西省电力公司科技项目 (520530202002)
作者简介：李小婧 (1992－)，女，高级工程师，主要从事人工智能、物联网等方面的研究.
*通信作者：李小婧，E-mail：594122581@qq.com

例，当本地的边缘服务器已发生通讯堵塞或宕机问题时，常见的解决方案会将终端切换到有足够容量且延迟最低的边缘服务器[4]。

由此，本文利用人工智能 (AI)，提出采用深度强化学习 (RL) 算法的边缘服务器智能选择方法来构建 Q 平衡网络，该网络具备高度动态匹配、高度智能选择、最低延迟等优点。由于 RL 算法特别适合高度变化的环境，本文基于该算法，引入动作抑制、四重 Q 学习 (QQL) 和 Q 值归一化等机制，构建了终端–服务器匹配延迟方差的强化学习模型，提出智能选择边缘服务器的具体解决方案。该方案在终端设备尤其并发数超标或边缘服务器宕机后，终端设备快速选择最优的边缘服务器进行智能切换，同时减轻延时等风险问题，实现了边缘服务器在满足延迟要求下的智能动态选择切换，同时考虑到了公平性策略[5]。本文研究不仅更好地解决了海量终端接入及大并发时导致的边缘服务器堵塞延迟甚至宕机等风险，同时对边缘服务器提供了更加科学合理的部署策略，以减少服务器的数量。

1　主要问题及研究思路

1.1　主要问题

对于服务器的选择和配置方法，国内外开展了较多研究，如文献 [6] 提出优化切换镜像服务器后减少整体用户延迟；文献 [7] 提出了用户–服务器匹配算法，利用系统最大化处理能力使用户切换到不同的服务器上达到整体延迟优化；文献 [8] 提出中值优化方法减少视频流端到端传输延迟优化问题；文献 [9] 提出在客户端使用域名系统 (DNS) 代理，并采用负载均衡功能选择延迟最低的内容分发网络 (content delivery network, CDN)，以供客户端选择合适的 CDN 服务器；文献 [10] 结合遗传算法进行移动边缘计算中的服务选择，以减少时延。但上述常见方法只考虑了减少延迟，并未实现高度动态化、智能化、公平化[11]。

以图 1 终端–服务器拓扑图为例，U、EN 和 DEN 分别为终端、边缘节点 (单个边缘服务器) 和委托的边缘节点[12]。EN 和 DEN 是配电站房运检系统中的一部分，DEN 是终端首次链接的边缘节点 EN，当发生终端设备切换 EN 时，服务器选择算法在 DEN 中运行，并由 DEN 给出终端该链接或切换的 EN。

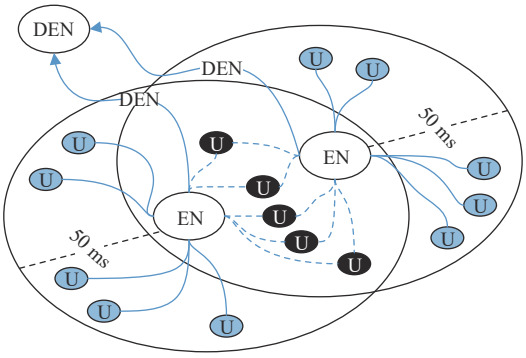

图 1　终端–服务器拓扑图

一个 EN 可处理多个终端，但其容量有限，尤其高并发时系统必须确保终端和 EN 之间的延迟不超过本设备指定的延迟阈值。如若有一些终端一起同时上报数据，并且 EN 同时针对终端进行业务处理，假设其端到端的延时阈值为 100 ms，那么任何终端与其 EN 之间的单向延迟不能超过 50 ms，因此给出 50 ms 的半径到每个 EN。此外，系统须确保终端在会话中所经历的延迟彼此尽可能接近，即终端间延迟变化最小，以提供公平的处理能力和竞争环境[13]。从图中可以看出，黑色终端可以连接到本地及相邻的多个 EN，考虑到多 EN 选择和延迟阈值、最大容量等约束条件，需充分考虑如何有效解决多变量优化问题，如终端设备会因上电\关机\重启动态接入，每类终端设备需满足延迟阈值要求，EN 达到处理瓶颈时引发堵塞甚至宕机等问题；动态部署或删除 EN 时，系统能否灵活、合理地将终端设备分配到合适的 EN 成为挑战[14]。

本文中的终端–边缘服务器匹配的公平性原则主要指早接入的终端拥有更高的优先权以及更低的延迟分配资源进行数据和业务处理，相对后接入的终端存在相对意义上的“不公平”。为此，本文使用 RL 并提出了一种智能选择方案，该方案提供了更高的公平性，同时也能更好地减少延迟。

1.2　研究思路

针对上述问题描述，为在满足延迟的条件下，实现动态化、智能化、公平化的终端–边缘服务器选择，本文主要围绕以下两项内容展开研究。

1) 边缘服务器选择模式研究。当终端切换边缘服务器时，公平性问题就突显出来。已接入的终端与后接入的终端相比具有较大的优势，因为存储、计算及网络资源有限，已接入的终端比后者能更快地接收在线实时数据，并占用更多的资源进行本地计算和业务处理，且当警告产生时，能更快与声光

设备进行联动报警。而后接入的终端会因为排队等机制，会将延迟效应进行积累。因此需研究更智能更公平的新型选择模式。

2) RL 强化学习算法研究。RL 算法提出将奖励的方差最小化，旨在充分考虑了系统性能、合理性及安全性后，在强化学习过程中将奖励最大化[15]。文献 [16] 提出了一种 RL 算法分类：具有安全系数折现的机制，并通过外部知识和风险进行指导纠正，该方法虽然预期的效果有可能很好，但由于风险值与环境的随机性有关，可能会出现较大的负面结果，因此不适合本文场景。

对此，本文使用终端和边缘服务器之间的地理距离作为延迟的参考指标，提出了两组次优匹配概念，设计了减少距离方差的方法[17]。针对公平性问题，本文借鉴了动作消除网络 (action elimination network, AEN)，即一种利用两个神经网络的系统：一个包含 Q 函数的近似值、另一个同时学习消除动作，提出了减少延迟变化的方法[18]。该方法输出一个线性上下文模型，并以高概率值来抑制动作。在该方式基础上，本文改进了 Q-table 技术，通过线性向量来指示动作是否可用，文章用固定的或学习的矢量值来处理动作的可用性。

2 主要研究内容

针对前述问题，本文采用 RL 技术，尤其是 Q-learning 算法开展模型研究，由于其具有较好的灵活性和动态性，也称为 Q 平衡网络。常规的 Q-learning 模型通常会选择最近的边缘服务器，即使其容量已到上限。本文将采用新技术应用于 Q-learning 算法，通过设计一个带奖励值的 Q 学习模型以实现终端设备智能公平选择边缘服务器。使用 RL 将终端设备加入网络这一过程称为“状态”，将接入这些终端设备的可用边缘服务器称为“动作空间”，本文将“动作”与“选择服务器”和“状态”与“终端”同义地关联起来[19]。

2.1 动作抑制

2.1.1 原理

由于服务器计算能力有限，一段时间后，边缘服务器处理能力将达到上限，一些预期动作在服务器某一时间点后将不可用。因此，必须考虑动作发生后更新 Q-values 时服务器的可用性，否则若奖励产生后仍使用即将不可用的服务器将导致消极恶性结果，影响后续终端的接入。区别于现有的动作屏蔽算法，当动作发生后，奖励也随即产生。动作抑制算法能将预期动作及奖励完全移除，该技术不仅能将不可用的服务器从备选状态中过滤掉，也有能力预测并选择期望的动作哪怕其 Q-values 不是最高。算法 1 对该技术进行了相关地说明。

算法 1 典型代码 1：Q-learning 动作抑制算法

initialize $Q(s, a)$ for all $s \in S$, $a \in A(s)$, arbitrarily.

define Aavailable for all possible $a \in A(s)$

set Q(terminal state,•)=0

for each episode do

initialize s

repeat (for each state of the episode)

choose highest a for s using policy derived from Q(e.g., −greedy)

while $a \notin$ Aavailable do

choose next highest a using the same policy

take action a, observe r, s' (according to a reward function)

remove a from Aavailable if limit is reached

$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \bullet \max a Q(s', a) - Q(s, a)]$

$s \leftarrow s'$

until s is terminal and Aavailable isn't empty

2.1.2 技术实现

技术实现的主要思想是在学习过程中抑制不可用情况。通过设计并使用可用操作集表来引导下一个动作，并跟踪在下一次迭代中可能采取的动作，选择最高的动作 Q-value，同时算法会分析服务器是否可用，若不可用会选择下一个最高的动作 Q-value。一旦服务器的容量达到最大值，在下一次迭代之前将其从操作集中删除。另外，算法将根据概率值 $1-\varepsilon$选择动作。在某些情况下，抑制动作会加强 Q-learning 中的探索。由于被淘汰的动作可能是最常被选中的动作之一，在后续迭代中被选中的概率较高。因此，需设计从可用动作中删除高概率已选择过的动作而选择次优的算法，即根据预期回报总和探索其他次优动作，既不采取随机动作也不使用最佳动作，而是介入两者之间的动作。

需要注意的是 RL 实体首先将进行离线训练，以获得有关环境的知识和训练策略参数以更接近最优策略。在这个阶段，算法将简单获得有关如何匹配和如何获得服务器与终端公平分配的知识。当模型部署并在线训练时，算法可能会惩罚某个终端−服务器的分配，在不好的状态下不再采取匹配动作。如果匹配时处于一个好的状态，则可以减轻处罚，这样“惩罚”或 Q-value 将逐步调整并收敛到

其真实值，以反应为未来训练中的预期奖励。

2.1.3　正则化效果

神经网络的 Dropout 概念和方法，通过随机神经元的输出乘以零以防止过度拟合。Q-learning 中的抑制动作与神经网络中的 Dropout 方法类似。由于在 Dropout 中某些神经元被丢弃并且其输出被忽略，删除动作类似于将 Q-value 与零相乘并忽略其对选择下一步操作的影响。通常，这会在训练中执行某种形式的正则化，在动作和状态空间很大的情况下是难以避免的[20]。

2.1.4　动作抑制的影响

消除网络可能会对后续构建更多的复杂算法并在网络上产生后续影响，被淘汰的动作表明其在某些场景的需求量很大，这些重要信息可以保留下来，用来对边缘服务器的部署提供建议，或构建一个更智能的代理来学习和预测网络的下一个状态。当所在的边缘服务器发生过载时其剩余容量可供终端选择。此外，需重要考虑的是参数“有效性”应被设计为可固定的，也可以是能学习的。在程序中，使用了服务器的固定容量来标识可用性，但当服务器的任务和请求接近容量上限时，也可以学习该值以消除过载的服务器。

2.2　Q 模型

本文将公平性定义为按终端设备与之接近的平均地理距离进行边缘服务器匹配。因此，本文在 Q-learning 模型中，需定义一个全局变量 D 代表距离的集合。该全局变量会在每次迭代时更新，并反映在动作的奖励中。为了在公平选择的背景下给出奖励，设计了 4 个不同的奖励函数以及 6 个模型。为将距离最小化，对所有的模型对距离的求和值给出负奖励。负奖励将驱使所有代理选择能最小化距离集总和值的动作，并影响其动作的 Q-value 值。模型设计如下。

1) 模型 1 (奖励 = −1 ×d)：d 是已匹配的终端与边缘服务器的距离，为训练代理尽可能减少地理距离，将奖励值设为 d 的负值，即为−1×d。地理距离最低时，Q-value 最高。关于终端 (U) 与边缘服务器 (E) 之间的地理距离计算以经纬度计算，公式为：

$$d = \sqrt{(\text{elong} - \text{ulong})^2 + (\text{elat} - \text{ulat})^2} \tag{1}$$

2) 模型 2 (奖励 = −1× stdv (D))：通过将奖励设置为当前连接距离的当前方差值 STDV 的负值来减少距离的全局平均值。由于代理的选择会影响平均值的变化，因此将奖励设置为将训练实体做出使距离的全局 STDV 值最小化的选择，从而使匹配的终端和服务器地理距离更接近平均值。

3) 模型 3(奖励 = −1× Δ stdv (D))：为促使距离 STDV 波动的发生，模型 3 给出与其采取的动作引起 STDV 变化成正比的负奖励。当代理采取动作时，模型会对延迟的 STDV 造成的变化给出惩罚。这将鼓励采取过多会导致 STDV 波动较小。

4) 模型 4 (奖励 = −1× |Δ 标准差 (D)|)：区别于模型 3，STDV 采用了变化的绝对值，即可以是负值或正值，但目标是将 STDV 的变化最小化。

5) 模型 5(组合模型 QQL)：为了减少距离方差及变化，组合上述所有模型方法。受多表 Q-learning 学习工作的启发[21]，模型 5 查看 4 个 Q-tables 的各自状态，从中选择最高的 Q-value，该想法类似于双 Q-learning，区别为双 Q-learning 是单目标，使用相同的奖励模型，而本文是多目标，并且是从多个奖励模型中进行选择。4 个模型中的每一个都会为代理所处的状态选择一个动作，QQL 将选择带来最高 Q-value 的动作。

6) 模型 6(最小值−最大值归一化模型)：上述模型涉及 4 个奖励函数，在每个表中都有不同的尺度。如 Model 1 使用实际距离作为奖励，而 Model 4 使用 STDV 的变化作为奖励。为达到最佳的公平性效果，模型 6 公平地比较了 QQL 中具有不同尺度的 4 个 Q-table，并进行了最小值−最大值归一化处理。该方法类似于局部响应归一化 (LRN) 理论[22]。LRN 将 4 层 CNN 在 CIFAR-10 数据集上的有效性从 13% 的测试误差减少到 11%。因此对 Model6，通过使用最小值−最大值计算来对 4 个表的 Q-values 按 0～1 的范围归一化。模型 6 为本文研究所推进的最佳模型，具体设计参见图 2，Q-table 中每一行表示每个状态的 Q-values 值，最小值−最大值归一化的计算公式为：

$$x_{\text{norm}_{ij}} = \frac{x_{ij} - x_{\min_i}}{x_{\max_i} - x_{\min_i}} \quad \forall i \in \text{rows},\ \forall j \in \text{columns} \tag{2}$$

2.3　Q 表扩展性设计

由于 Q-learning 初始设计为处理有限和预定义的动作和状态空间，需对其开展扩展性和实用性改进，因终端和边缘服务器可能会动态的增加或减少，而固定大小的 Q-table 无法动态增加。为解决该问题，通过构建了一个估算函数去获得当前 Q-table 新的状态和新的动作。该估算函数将相邻终端的状态及其服务器的动作所对应的 Q-values 平均化，形成并插入一条新的记录。为进一步提高上述扩展性，将 Q-table 实现为动态哈希图，也称为字典或哈希表，用以实现对表的快速高效地查找、增加和删除[23]。

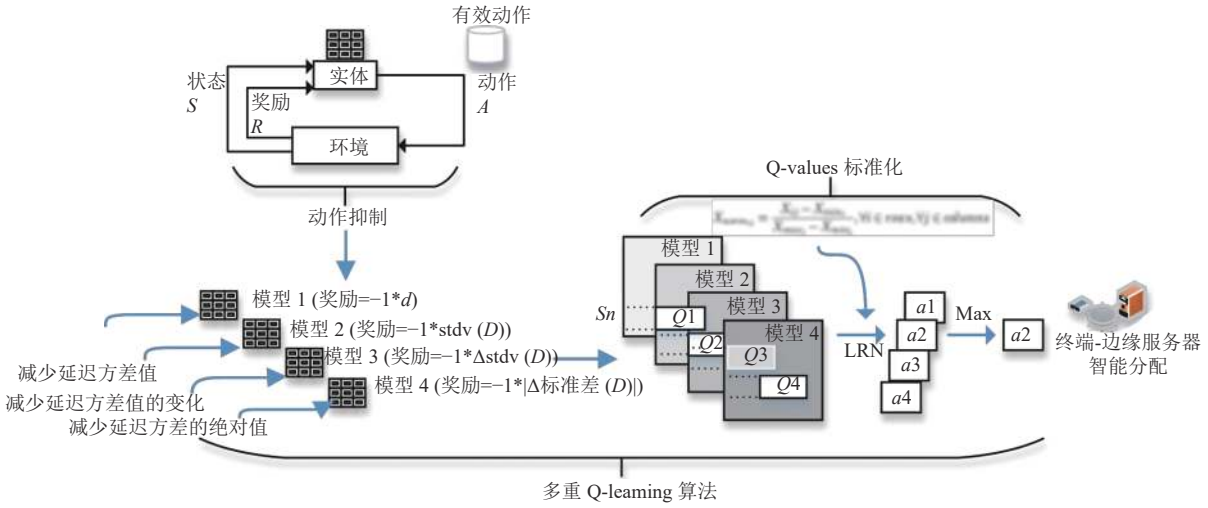

图 2 系统设计概略图

3 实验及性能评估

3.1 数据收集

首先，在每台边缘服务器上运行了一个收集每个终端的延迟、抖动、IP 地址和位置 (经、纬度) 的脚本，并将记录结果存入到 CGCSDD 数据集。在模拟测试中，153 个终端分布在某一个虚拟区域。如图 3 所示，在终端附近随机模拟部署 153 个边缘服务器并开展了仿真实验。

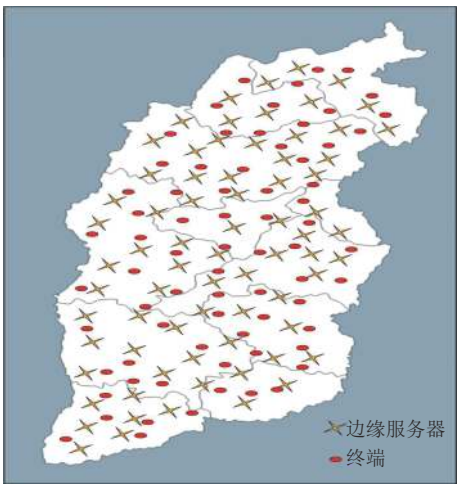

图 3 终端节点及服务器随机分布图

3.2 实验与仿真

通过执行脚本，将 153 个终端分配到 153 台边缘服务器上后，无论每个终端有没有指定到最优的边缘服务器，都会导致会话量大，理论上会产生 153！=2×10 269 个可能匹配的数量。在训练过程中，采用超参数解决 RL 的范围问题，并指定在表查询中使用超参数进行算法优化。本实验和模型中所用的超参数采用以下各值：学习率 $\alpha = 0.1$，奖励折扣因子 $\gamma = 0.6$，探索因子 $\varepsilon = 0.1$，训练持续时期 = 100 000。

本文在传统方法的基础上又构建了 3 种锚方法，力图将延迟最小化而不仅是延迟的方差，同时尽量减少延迟的变化。具体描述如下。

1) 锚点 1 是实践中最常用的方法，通过将终端匹配到最近的边缘服务器 (基于地理距离)，该服务器具有留给新终端的容量。

2) 锚点 2 将终端匹配到第二近的可用边缘服务器。这种方法会将“最佳”服务器信息保存起来以匹配后续需接入的终端。

3) 锚点 3 将前一半的 76 个终端匹配到距离范围 50% 以内的边缘服务器 (即第 75 个最近的边缘服务器)，同时将另一半终端匹配到最近的边缘服务器。

4) 传统方法的主要目的是减少方差，其原理为：①对第一个终端，找到该节点与任何边缘服务器之间的最低和最高延迟，并将这两个值计算平均值 dconv，然后将该节点连接到 dconv 最接近的边缘服务器；②对接下来的每个节点，将其连接到延迟与 dconv 最接近的边缘服务器 (dconv 参见步骤①只计算一次)。

图 4 为采用了锚点方式、传统方式、本文模型 6 方式所测试的终端–边端服务器地理距离分布效果图。表 1 为各模型及方式的延迟测试结果，在锚点 2 和锚点 3 中，稍微降低了方差的同时也增加了平均值，锚点 2 具有最佳方差。

传统方法的结果取决于第一个选择的节点，它决定了 dconv 值。因为算法运行了 153 次，每次都从不同的节点开始，根据 3 个典型 dconv 值：最

低、中间、最高选择了 3 个案例开展测试及测试数据整理，测试结果参见表 1。

从表 1 可看到模型 6 的方差最小，比最好的锚方法 (锚点 2) 效果显著提高 30%。与传统方法相比，模型 6 仍然比最好的传统方法有更好的方差结果，同时有更好的平均延迟。

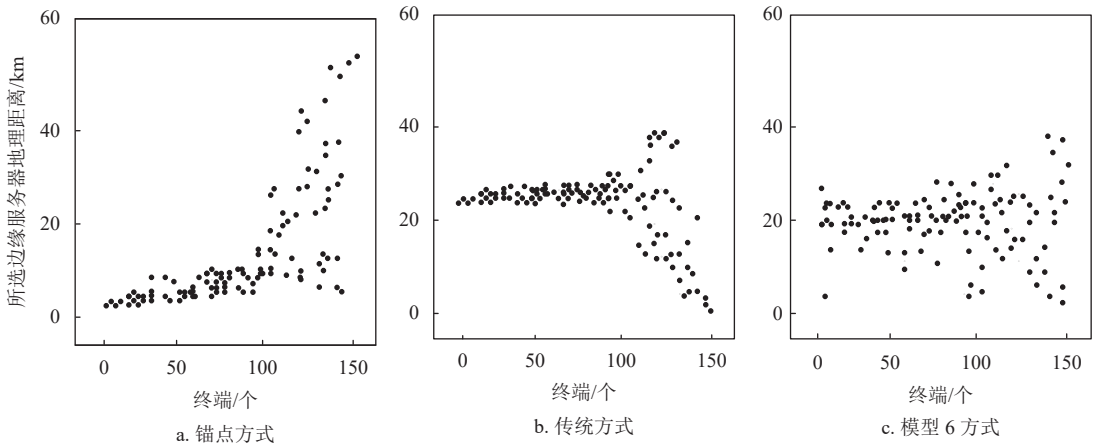

图 4　实验方法及测试结果

表 1　模型延迟测试结果
(各方法测试数据中每列取最小值)

方法	平均值(avg)	方差值(stdv)
锚点 1	10.63	12.67
锚点 2	10.78	12.6
锚点 3	13.13	12.62
传统方法(最小值)	17.89	9.3
传统方法(中间值)	23.38	8.92
传统方法(最大值)	27.2	8.75
模型 1	16.76	14.33
模型 2	19.22	11.95
模型 3	17.19	10.92
模型 4	19.35	9.83
模型 5	18.94	8.67
模型 6	17.4	8.22

按公平性定义终端设备与之接近的平均地理距离进行边缘服务器匹配，平均地理距离最优者具有最佳公平性，从图 4a中可看出，对于算法后期连接到系统的终端，无论在哪个位置接入边缘服务器，其延迟阈值都超过了；图 4c 中，无论终端在哪个位置接入，QNetwork都满足延迟阈值距离分配最优，具备最佳公平性。同时可看到锚方法导致用户之间的延迟值差异很大，而这在 QNetwork 中得到了更好的控制。原因如下：锚点 1 首先第一批终端被分配到距离最短的可用服务器。一段时间后，这些服务器的容量已达到极限，系统只能将更远的服务器分配给后续的终端。图 4b 中传统算法成功减少了前半部分终端的方差，但对于后半部分，由于更近的服务器不可用，导致距离值超标。另一方面，模型 6 首先将终端分配到距离接近最终平均值 17.40 的服务器，因此方差值较低。由于模型 6 是 4 个不同模型的组合，并且每个时期都会从 4 个模型中选择一个，因此监控使用了哪个模型将有助于验证这些模型。结果显示，模型 3 使用了 139/153 次，而模型 2 和模型 4 各使用了 7/153 次。这证明了 Model 3 拥有最好的 Q-values，实验结果也证明了同时使用该模型提高了整体结果。

4　结 束 语

本文针对大量终端采用公平性原则并发接入边缘服务器的多元多约束问题，提出了 RL 深度学习解决方案。同时，采用实验仿真数据证明了 RL、特别是 QQL 比现有方法更公平地解决接入问题，实验数据表明终端−服务器延迟的 STDV 值最多减少了 35%。此外，在设计此解决方案时，本文还引入了动作抑制、QQL 和归一化 Q-values 等技术，设计了更具扩展性和可实施性的 RL 新型系统。

由于 Q-table 方法对更复杂场景处理能力扩展性有限，将考虑使用近似函数或相似度函数来处理超量的动作和状态空间。当新终端或边缘服务器上线后，通过使用近似函数或相似度函数来估计和预测 Q-value，使终端节点更快地找到其最优 Q-value 值。另一种是使用更复杂的架构，如 DQN(deep Q-network)，它消除了对用表存储 Q-values 的限制，并将其替换为神经网络以分析评估 Q-values 的状态，研究并使用这种更先进的架构。

参　考　文　献

[1] 王友祥, 陈杲, 黄蓉. 云边协同技术发展分析[J]. 邮电设计

技术, 2021(3): 1-6.
WANG Y X, CHEN G, HUANG R. Analysis on cloud-edge collaboration technology development[J]. Designing Techniques of Posts and Telecommunications, 2021(3): 1-6.
[2] 刘远龙, 潘筠, 王玮, 等. 用于泛在电力物联网的配电变压器智能感知终端技术研究[J]. 电力系统保护与控制, 2020, 48(16): 140-146.
LIU Y L, PAN Y, WANG W, et al. Research on intelligent sensing terminal technology of a distribution transformer for ubiquitous power internet of things[J]. Power System Protection and Control . 2020, 48(16): 140-146.
[3] 刘亮, 李卉. 边缘计算网关的功能设计与系统实现[J]. 电测与仪表, 2021, 58(8): 42-48.
LIU L, LI H. Functional design and system implementation of edge computing gateway[J]. Electrical measurement and instrumentation, 2021, 58(8): 42-48.
[4] 蒲世亮, 袁婷婷. 基于云边融合的物联网智能服务架构探讨[J]. 智能物联技术, 2018, 1(1): 1-6.
PU S L, YUAN T T. Research on intelligent architecture for IoT services based on fusion of cloud and edge computing[J]. Technology of IoT & AI, 2018, 1(1): 1-6.
[5] 聂峥, 章坚民, 傅华渭. 配变终端边缘节点化及容器化的关键技术和应用场景设计[J]. 电力系统自动化, 2020, 44(3): 154-161.
NIE Z, ZHANG J M, FU H W. Key technologies and application scenario design for making distribution transformer terminal unit being a containerized edge node[J]. Automation of Electric Power Systrems, 2020, 44(3): 154-161.
[6] WEBB S D, SOH S. Adaptive client to mirrored-server assignment for massively multiplayer online games[C]// Multimedia Computing and Networking 2008. [S.l.]: SPIE, 2008, 6818: 179-186.
[7] FARLOW S, TRAHAN J L. Client-Server assignment in massively multiplayer online games[C]//2014 Computer Games: AI, Animation, Mobile, Multimedia, Educational and Serious Games (CGAMES). [S.l.]: IEEE, 2014: 1-8.
[8] HU Y, NIU D, LI Z. A geometric approach to server selection for interactive video streaming[J]. IEEE Transactions on Multimedia, 2016, 18(5): 840-851.
[9] GOEL U, WITTIE M P, STEINER M. Faster web through client-assisted CDN server selection[C]//2015 24th International conference on computer communication and networks (ICCCN). [S.l.]: IEEE, 2015: 1-10.
[10] QIN F, ZHAO Z, ZHANG H. Optimizing routing and server selection in intelligent SDN-based CDN[C]//2016 8th International Conference on Wireless Communications & Signal Processing (WCSP). [S.l.]: IEEE, 2016: 1-5.
[11] ZHANG S, DI N, HU Y, et al. Server selection and topology control for multi-party video conferences[C]// Proceedings of Network and Operating System Support on Digital Audio and Video Workshop. [S.l.]: ACM, 2014: 43-48.
[12] WU H, DENG S, LI W, et al. Service selection for composition in mobile edge computing systems[C]//2018 IEEE International Conference on Web Services (ICWS). [S.l.]: IEEE, 2018: 355-358.
[13] 徐超, 吴波, 姜丽丽, 等. 云—边缘系统中跨域大数据作业调度技术研究[J]. 计算机应用研究, 2020, 37(3): 754-758.
XU C, WU B, JIANG L L, et al. Task scheduling for geo-distributed data analytics in cloud-edge system[J]. Application Research of Computers, 2020, 37(3): 754-758.
[14] 栾奇麒, 程力涵, 李春鹏, 等. 基于智能边缘计算的物联接入网关容错机制研究[J]. 电子设计工程, 2022, 30(23): 90-93,98.
LUAN Q L, CHEN L H, Li C P, et al. Research on fault tolerance mechanism of IoT access gateway based on intelligent edge computing[J]. Electronic Design Engineering, 2022, 30(23): 90-93,98.
[15] 张凤荔, 赵佳君, 刘东, 等. 基于深度强化学习的边云协同串行任务卸载算法[J]. 电子科技大学学报, 2021, 50(3): 398-404.
ZHANG F L, ZHAO J J, LIU D, et al. Edge cloud collaboration serial task offloading algorithm based on deep reinforcement learning[J]. Journal of University of Electronic Science and Technology of China, 2021, 50(3): 398-404.
[16] GARCIA J, FERNÁNDEZ F. A comprehensive survey on safe reinforcement learning[J]. Journal of Machine Learning Research, 2015, 16(1): 1437-1480.
[17] LIU L, LI B, GUO R. Consensus control for networked manipulators with switched parameters and topologies[J]. IEEE Access, 2021(99): 1.
[18] DENG Y, LI Y, TANG X, et al. Server allocation for multiplayer cloud gaming[C]//Proceedings of the 24th ACM international conference on Multimedia. [S.l.]: ACM, 2016: 918-927.
[19] 庄夏. 基于并行粒子群和 RL 的无人机航路规划算法设计[J]. 西南师范大学学报 (自然科学版), 2016, 41(3): 31-36.
ZHUANG X. On path planning algorithm for ucav based on parallel particle swarm and rl[J]. Journal of Southwest Normal University (Natural Science Edition), 2016, 41(3): 31-36.
[20] 韩忻辰, 俞胜平, 袁志明, 等. 基于 Q-learning 的高速铁路列车动态调度方法[J]. 控制理论与应用, 2021, 38(10): 1511-1521.
HAN X C, YU S P, YUAN Z M, et al. High-Speed railway dynamic scheduling based on Q-learning method[J]. Control Theory and Applications, 2021, 38(10): 1511-1521.
[21] KANTASEWI N, MARUKATAT S, THAINIMIT S, et al. Multi Q-table Q-learning[C]//2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES). [S. l.]: IEEE, 2019: 1-7.
[22] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
[23] 赵梓铭, 刘芳, 蔡志平, 等. 边缘计算: 平台、应用与挑战[J]. 计算机研究与发展, 2018, 55(2): 327-337.
ZHAO Z M, LIU F, CAI Z P, et al. Edge compution: platforms, applications and challenges[J]. Journal of Computer Research and Development, 2018, 55(2): 327-337.

编 辑 税 红