

融合边缘计算与区块链的工业互联网资源优化配置研究^①

杨 乐^{②***} 李 萌^{③***} 叶欣宇^{***} 孙恩昌^{***} 张延华^{***}

(* 北京工业大学信息学部 北京 100124)

(** 先进信息网络北京实验室 北京 100124)

摘 要 随着通信网络技术的发展,工业互联网技术及其应用日益成熟。然而,工业互联网中对数据的安全性和隐私性需求为传统工业互联网架构带来前所未有的挑战。区块链技术作为极具发展前景的新技术之一,已被应用于工业互联网系统中。但是,当前基于区块链的工业互联网系统中仍存在一些亟待解决的问题,设备处理计算任务产生能耗较大,区块链中共识过程效率偏低,系统中存在严重的计算开销。针对上述问题,本文在工业互联网架构中引入移动边缘计算(MEC)技术,提升设备处理计算任务的能力和区块链节点的共识效率。同时,充分考虑工业互联网系统的设备能耗和计算开销。综上,本文提出一种融合边缘计算和区块链的工业互联网资源分配优化方法,以减少系统设备能耗和计算开销为目标,并将此优化问题构造为马尔可夫决策过程(MDP),系统中的卸载决策、区块尺寸和计算服务器均可动态调整和选择。根据优化场景的高动态、多维度特点,本文采取深度强化学习方法优化求解所提问题。通过仿真验证,相比于其他现有方法,本文所提方法可有效提升系统性能。

关键词 工业互联网;区块链;深度强化学习(DRL);移动边缘计算(MEC);资源分配

0 引 言

当前,工业互联网的快速发展吸引了工业界和学术界的广泛关注^[1]。工业互联网技术的应用可实现机器与机器^[2]、人与机器的高效便捷的交互。多种类型的工业互联网设备,又名机器类型通信设备(machine-type communication devices, MTCDs),被广泛应用于多种工业领域中,包括环境监测、制造业、视频监控以及智能网络等。预计到 2021 年,将有近 250 亿数量的工业互联网设备接入网络,用于收集和传输工业数据^[3]。

在工业场景下^[4],工业数据的安全性和可靠性至关重要。然而,由于 MTCD 分布极其广泛以及交

互数据种类多样,数据的隐私性和安全性往往难以保证。另一方面,工业互联网中能量资源和计算资源短缺明显,因大多数 MTCD 在脱离人工控制状态下进行工作,且为了延长设备工作寿命,此类设备仅装配有限的电池供给和轻量的计算资源。因此,一些复杂繁重的计算任务难以在这些设备内独立完成。

针对存在于工业互联网系统中的数据安全性问题,新兴的区块链技术被广泛应用于工业领域^[5]。区块链最早是一种服务于比特币的底层技术,用于记录存储比特币的各类交易。相比于传统的由第三方控制的中心化账本,区块链实质上是一种拥有点对点网络结构的分布式账本,可有效保障数据的安全性和可靠性。然而,无论是以比特币、以太坊为首

① 国家自然科学基金(61901011)和北京市教育委员会科技计划(KM202110005021)资助项目。
② 男,1995 年生,硕士生;研究方向:工业互联网,区块链,网络资源管理;E-mail: yangl@emails.bjut.edu.cn
③ 通信作者,E-mail: limeng720@bjut.edu.cn
(收稿日期:2020-03-11)

的公链结构,还是应用拜占庭容错共识机制的私链结构,都需要大量的计算资源用于挖矿和共识。因此,区块链中计算的效率和稳定性极为重要。

为应对设备、节点计算能耗过高、计算资源短缺的问题,计算任务卸载的方式可有效提高设备和节点的计算能力。通过将计算任务从设备卸载到服务器,不仅可明显提升计算效率,还减少了设备的计算能耗。近年来,大量的研究工作聚焦于包含有中心云计算和移动边缘计算^[6-7] (mobile edge computing, MEC)的混合数据计算模型^[8]。其中,相比于中心化的云计算,分布式的 MEC 拥有低传输时延、低计算开销以及低设备能耗等优势。更重要的是,分布式的计算卸载结构可支撑区块链中共识节点的验证计算过程^[9]。因此,将区块链和 MEC 整合至工业互联网场景成为了当下一新的研究发展方向^[10]。对此,Zhang 等人^[11]提出了一种结合了区块链和 MEC 技术的车载自组织网络安全结构。他们应用区块链保证数据的可追溯性,并借助 MEC 解决区块链中的高维计算问题,既保障了数据的安全性,也提升了区块链节点的验证效率。Qiu 等人^[12]提出了一种计算卸载框架用于提升移动设备的计算能力。他们将设备中的挖矿任务和数据处理任务卸载至 MEC 服务器或云计算服务器,有效提升计算任务的处理效率。

近年来,针对工业互联网系统高动态、多维度等特点,深度强化学习 (deep reinforcement learning, DRL) 逐渐成为受高度关注的优化方法^[13-15]。智能体 agent 按照所学习的策略对环境施加动作,并从中获得即时奖励。两者交互过程持续进行,直到环境达到最终状态。在此过程中,agent 不断学习和调整策略以获得最大的长期累积收益。DRL 中加入了深度神经网络结构对 agent 的动作进行估计,可更有效应对高维的状态、动作问题。近年来,基于 DRL 方法对工业互联网系统性能进行优化的研究逐渐增加。Qiu 等人^[16]基于 dueling deep Q-learning 算法对区块链架构下的工业互联网系统吞吐量进行性能优化,有效提升了系统中交易的处理效率。Feng 等人^[17]应用 DRL 算法,针对基于区块链的工业互联网系统,联合优化了 MEC 系统的计算效率和

区块链系统的交易吞吐量。

尽管上述研究均结合区块链和 MEC 技术对系统性能进行了不同程度的优化,但在工业互联网场景下仍将面临一些潜在的问题和挑战。例如,如何选择计算任务的处理方式平衡设备能耗和系统计算开销。同时,在计算任务卸载过程中,考虑到计算服务器的计算资源和计算开销,须选择更适合的目标服务器处理计算任务。此外,由于区块链技术的引入,设备能耗和计算负载的问题也将更加严峻。因此,在设计系统时均须仔细考虑以上问题。

针对上述问题和挑战,本文提出一种融合边缘计算和区块链技术的工业互联网资源分配优化方法。首先,联合考虑工业互联网场景下的控制器和计算服务器状态。其次,动态调整卸载决策、区块链的区块尺寸和计算服务器的选择。最后,通过设置状态空间、动作空间和奖励函数,建立一个马尔可夫决策过程 (Markov decision process, MDP) 用以描述此动态联合优化问题,并基于 DRL 方法进行求解,不仅有效减少设备计算能耗,还显著降低系统的整体计算开销,并提高数据共识效率。相比于已有相关研究,本文方法更适应高动态、多维度的工业互联网场景,在设备能耗、系统开销上可取得更好的优化效果。

本文的结构安排如下,第 1 节介绍本文的系统模型及网络场景;第 2 节将建立一个包含卸载决策、区块尺寸调整和计算服务器选择的联合优化问题;第 3 节基于 DRL 方法对所提优化问题求解;第 4 节介绍仿真实验环境和参数,并根据仿真结果进行讨论和分析;第 5 节是对本文工作的总结和展望。

1 系统模型

如图 1 所示,本文的系统模型包含 3 部分,分别为用户层、控制器层和区块链层,本章将对这 3 部分的模型详细描述。

1.1 用户层

本文假设在所述网络场景中存在 u 个小区。在每个小区中,都分布着大量的工业互联网设备,与本地控制器连接。工作期间这些设备将收集的本地数

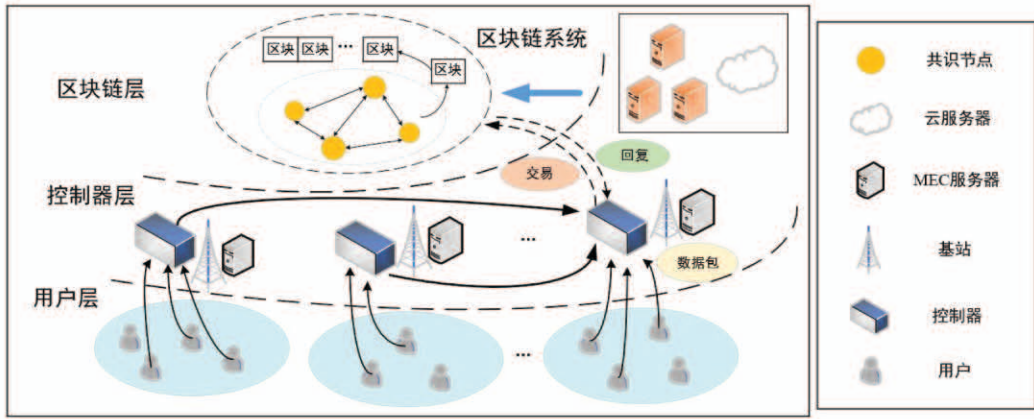


图1 系统模型图

据传输至本地控制器中,用于数据共识和记录。以智能电表设备为例,各小区内智能电表定时向其本地控制器传输耗电量数据。在每个时刻 $t = \{0, 1, \dots, T-1\}$, 控制器系统指定一个控制器作为主控制器,接收各本地控制器的工业数据,并对这些数据处理、打包,上传到区块链系统。

1.2 控制器层

在每个小区内,都包含一个本地控制器和一个部署于基站旁的本地 MEC 服务器。本地控制器不仅可通过基站收发数据,还可对数据进行处理和打包。小区可由集合 $U = \{1, 2, \dots, u\}$ 表示,并假设控制器与小区一一对应,表示为 $U^* = U = \{1, 2, \dots, u\}$, 控制器 u 在其关联小区 u 收集本地工业数据。

此外,每个控制器均有独立的能量状态和算力状态,分别表示为 $G(t) = \{g_1(t), g_2(t), \dots, g_u(t)\}$ 和 $C(t) = \{c_1(t), c_2(t), \dots, c_u(t)\}$ 。本地 MEC 服务器则用于处理本地控制器卸载的计算任务。

在工作期间,每个本地控制器收集 MTCD 的不同类型工业数据,且为节省控制器与区块链系统间的数据传输损耗,各本地控制器将数据首先传输至主控制器。主控制器对数据处理、打包,并将打包好的数据作为一批交易传输至区块链系统。其中,主控制器由此产生的计算任务既可在设备本地处理,也可卸载至本地 MEC 服务器处理,来提升处理效率并减少本地能耗。任务处理的能耗和计算开销如下,若主控制器选择在设备本地处理,时刻 t 时的处理能耗 $E_c(t)$ 表示为

$$E_c(t) = z_u \cdot q(t) \quad (1)$$

其中, $q(t)$ 为时刻 t 时的计算任务复杂度, z_u 为与设备算力状态相关的能耗系数,表示为

$$z_u = 10^{-27} \cdot (c_u)^2 \quad (2)$$

其中, c_u 表示控制器 u 的算力状态。

若主控制器选择卸载计算任务,其在时刻 t 产生的数据传输能耗 $E_s(t)$ 和由本地 MEC 服务器收取的计算开销 $M_s(t)$ 分别表示为

$$E_s(t) = p_t \cdot \frac{d(t)}{r_{cs}} \quad (3)$$

$$M_s(t) = \xi(t) + \mu \cdot q(t) \quad (4)$$

其中, $d(t)$ 表示时刻 t 时计算任务的数据量, p_t 为主控制器的传输功率, r_{cs} 表示控制器与本地 MEC 服务器间的数据传输速率, $\xi(t)$ 和 μ 分别为本地 MEC 服务器的固定计算开销和开销系数。

将收集的数据处理、打包后,主控制器将此未验证的区块传输至区块链系统,其传输能耗 $E_n(t)$ 表示为

$$E_n(t) = p_t \cdot \frac{d(t)}{r_{cn}} \cdot n \quad (5)$$

其中, r_{cn} 为控制器与区块链系统间的数据传输速率, n 为区块链节点数量。

综上,在时刻 t ,若主控制器在设备本地执行计算任务,则其产生的总能耗 $E(t)$ 和服务收取的计算开销 $M_c(t)$ 分别为

$$E(t) = E_c(t) + E_n(t) \quad (6)$$

$$M_c(t) = 0 \quad (7)$$

相反,若主控制器将计算任务卸载至本地 MEC 服务器,则其产生的总能耗 $E(t)$ 和服务收取的计算开

销 $M_c(t)$ 分别为

$$E(t) = E_s(t) + E_n(t) \quad (8)$$

$$M_c(t) = M_s(t) \quad (9)$$

1.3 区块链层

在区块链系统中,存在 n 个共识节点,由集合 $N = \{1, 2, \dots, n\}$ 表示。这些节点负责验证来自主控制器发送的交易,并将已验证的区块添加至区块链。

(1) 共识机制。本文的区块链共识机制应用了实用型拜占庭容错机制 (practical Byzantine fault tolerance, PBFT),可在被攻击节点数量低于总节点数量三分之一的情况下保证系统的正常运行。如图 2 所示, PBFT 的共识过程可分为请求、预准备、准备、确认和回复 5 个阶段,具体过程如下。

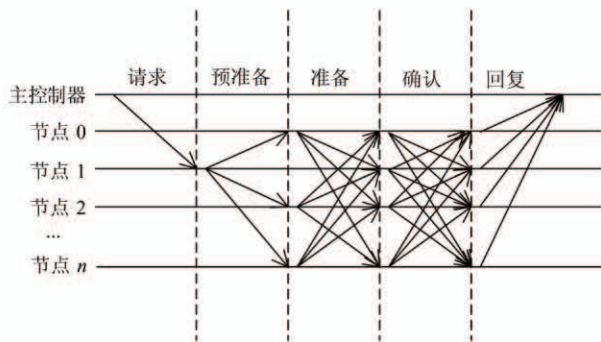


图 2 PBFT 工作流程

第 1 步,主控制器向区块链系统发送数据共识请求和已打包的未验证区块,并由区块链系统随机分配一个主节点,负责验证区块的相关信息。具体地,主节点依次验证区块的签名和消息身份验证码 (message authentication code, MAC),若以上信息验证通过,主节点将继续验证区块中每个交易的签名和 MAC。本文假设节点验证一个签名及生成/验证一个 MAC 所需的计算轮数分别为 θ 和 α 。

因此,在请求阶段,主节点所需的计算轮数为

$$c_1(t) = (1 + \frac{b(t)}{h}) \cdot (\theta + \alpha) \quad (10)$$

其中, $b(t)$ 为时刻 t 时区块中交易的总量, h 为交易的平均尺寸。

第 2 步,主节点生成并发送一个 MAC 至系统中各从节点。各从节点将验证此 MAC,验证通过后再验证区块中各交易的签名和 MAC。因此,在预准备阶段,主节点和从节点所需的计算轮数分别为

$$c_{2p}(t) = (n - 1) \cdot \alpha \quad (11)$$

$$c_{2r}(t) = \alpha + \frac{g \cdot b(t)}{h} \cdot (\theta + \alpha) \quad (12)$$

其中, g 为由主控制器发送的正确交易的比例。

第 3 步,验证通过的从节点均生成并发送一个 MAC 至所有其余节点,证实其成功完成上一步验证过程。每个节点均收到并验证 $n - 1$ 个来自其余节点发送的 MAC。当验证通过的节点数量多于 $2f(f = (n - 1)/3)$ 时,下一阶段过程才能进行。因此,在准备阶段,主节点和从节点所需的计算轮数分别为

$$c_{3p}(t) = 2f \cdot \alpha \quad (13)$$

$$c_{3r}(t) = (n - 1 + 2f) \cdot \alpha \quad (14)$$

第 4 步,收到超过 $2f$ 数量正确验证消息的节点再次向所有其余节点发送一个 MAC,证明其完成了上一步验证。与此同时,每个节点也再次对这些 MAC 进行验证。在确认阶段,所有节点所需的计算轮数均为

$$c_4(t) = (n - 1 + 2f) \cdot \alpha \quad (15)$$

第 5 步,所有收到超过 $2f$ 数量有效确认信息的节点向主控制器发送回复消息。若正确的回复消息数量超过 $2f$,则共识过程视为成功完成,生成的区块顺利被添加至区块链,并向全网络广播。同时,生成区块也根据区块尺寸产生开销,表示为

$$M_b(t) = \varepsilon \cdot s(t) \quad (16)$$

其中, $s(t)$ 表示在时刻 t 生成的区块尺寸, ε 为描述区块尺寸与区块开销的常数。

(2) 服务器计算。在每一步验证过程中,所有节点 (包括主节点和从节点) 都会验证大量的签名和 MAC,由此产生了一批复杂繁重的计算任务。同时,在每次 PBFT 共识过程中还存在与总交易量相关的时延 t_l ,若共识时延超出此时限,生成的区块将无法及时添加至区块链。因此,共识的效率至关重要。

为应对这些复杂繁重的计算任务,并满足共识过程的时限要求,MEC 服务器和云计算服务器被用于支持节点的共识过程,两种服务器拥有不同的计算能力,并产生不同的计算开销。此外,当区块尺寸不足而导致多次共识过程时,服务器将产生额外的计算开销。

由于 MEC 服务器和云计算服务器的计算资源

同时被其他计算任务占用,难以在每时刻准确了解服务器的计算资源和计算开销状态。因此,本文将服务器的计算资源状态和计算开销状态构造为随机变量 γ 和 η ,将 γ 和 η 分别量化为 p 和 q 个离散值,由 $X = \{x_0, x_1, \dots, x_{p-1}\}$ 和 $Y = \{y_0, y_1, \dots, y_{q-1}\}$ 表示。在时刻 t ,服务器的计算资源状态和计算开销状态分别为 $\gamma(t)$ 和 $\eta(t)$,应用 Markov 链对其建模,描述其状态转移关系。 $\gamma(t)$ 和 $\eta(t)$ 分别服从状态转移概率 $P_{x_i x_j}^\gamma = \Pr\{(t+1) = x_j | (t) = x_i\}$ 和 $P_{y_i y_j}^\eta = \Pr\{(t+1) = y_j | (t) = y_i\}$,其分别存在于宽度为 $X \times X$ 和 $Y \times Y$ 的状态转移矩阵 $\mathbf{R}^\gamma(t) = [P_{x_i x_j}^\gamma(t)]_{X \times X}$ 和 $\mathbf{O}^\eta(t) = [P_{y_i y_j}^\eta(t)]_{Y \times Y}$ 中。

根据每一阶段所需计算轮数,共识过程所需总计算轮数为

$$c(t) = \left[\frac{(1+g) \cdot b(t)}{h} + 1 \right] \cdot \theta + \left[\frac{(1+g) \cdot b(t)}{h} + 3n + 4f - 1 \right] \cdot \alpha, \quad b(t) \leq s(t) \quad (17)$$

若总交易量大于区块尺寸,则多次共识所需总计算轮数(本文假设最多为两次共识)为

$$c(t) = \left[\frac{(1+g) \cdot b(t)}{h} + 2 \right] \cdot \theta + \left[\frac{(1+g) \cdot b(t)}{h} + 6n + 8f - 2 \right] \cdot \alpha, \quad b(t) > s(t) \quad (18)$$

因此,共识时延和共识计算开销分别为

$$D_n(t) = \begin{cases} \frac{c(t)}{\gamma(t)} + 3t_b & b(t) \leq s(t) \\ \frac{c(t)}{\gamma(t)} + 3t_b + t_i & b(t) > s(t) \end{cases} \quad (19)$$

$$M_n(t) = \begin{cases} \eta(t) & b(t) \leq s(t) \\ 2\eta(t) & b(t) > s(t) \end{cases} \quad (20)$$

其中, t_b 为节点间的广播时延, t_i 为区块生成间隔。

据此,总开销 $M(t)$ 表示为

$$M(t) = M_c(t) + M_n(t) + M_b(t) \quad (21)$$

当共识时延 $D_n(t)$ 小于共识时限 t_l ,则可视为一次成功的共识过程,所生成的区块被添加至区块链,主控制器也将收到数据共识成功的消息。

综上,本文设置 $W(t)$ 表示时刻 t 时系统总能耗和总开销之和,表示为

$$W(t) = k_e \cdot E(t) + k_m \cdot M(t) \quad (22)$$

其中, k_e 和 k_m 分别为能耗和开销的权重系数,且满足 $0 \leq k_e \leq 1$, $0 \leq k_m \leq 1$ 和 $k_e + k_m = 1$ 。

2 问题建模

基于区块链的工业互联网系统存在高动态性、多维度性等特点。因此,本文通过构建一个离散马尔可夫决策过程来描述此优化问题,旨在联合优化设备能耗和系统计算开销,提高共识效率。通过构建优化问题的状态空间、动作空间和奖励函数,基于深度强化学习的优化方法对问题求解。

2.1 状态空间

在每个决策时刻,智能体 agent 通过观察环境状态,实时地学习经验并升级问题策略。结合本文优化问题,agent 须实时获取各控制器和计算服务器的状态。因此,本文定义时刻 t 时的状态空间为

$$s(t) = \begin{bmatrix} g_1(t) & g_2(t) & \dots & g_u(t) \\ \gamma_1(t) & \gamma_2(t) & \dots & \gamma_v(t) \\ \eta_1(t) & \eta_2(t) & \dots & \eta_v(t) \end{bmatrix} \quad (23)$$

其中, $G(t) = \{g_1(t), g_2(t), \dots, g_u(t)\}$ 为所有控制器的能量状态集合, $\gamma(t) = \{\gamma_1(t), \gamma_2(t), \dots, \gamma_v(t)\}$ 和 $\eta(t) = \{\eta_1(t), \eta_2(t), \dots, \eta_v(t)\}$ 分别为服务器的计算资源状态集合和计算开销状态集合,包含 $v-1$ 个 MEC 服务器和 1 个云计算服务器。

2.2 动作空间

为获取最大长期收益,需动态调节系统的若干部分。本文设置的调节动作包含卸载决策动作 $a_p(t)$,区块尺寸调整动作 $a_b(t)$ 和服务器选择动作 $a_s(t)$,在时刻 t 时的动作空间表示为

$$a(t) = [a_p(t), a_b(t), a_s(t)] \quad (24)$$

其中, $a_p(t) \in (0, 1)$ 为卸载决策, $a_p(t) = 0$ 表示本地处理, $a_p(t) = 1$ 表示卸载。 $a_b(t) \in (1, 2, \dots, w)$ 为所选区块尺寸等级, $a_s(t) \in (1, 2, \dots, v)$ 为所选用于支持共识节点计算的服务器。

2.3 奖励函数

为降低设备能耗和系统计算开销,并提高共识效率,本文据此构建即时奖励函数为

$$r(t) = \begin{cases} \sigma(t) - k \cdot \frac{W(t)}{b(t)} & D_n(t) \leq t_l \\ -k \cdot \frac{W(t)}{b(t)} & D_n(t) > t_l \end{cases} \quad (25)$$

其中, $\sigma(t)$ 为共识成功的额外基础奖励, k 为系统总开销的权重参数。

3 基于 DQL 的工业互联网系统资源分配决策算法

针对工业互联网系统中控制器和服务器状态的高动态性和多维度性的特点,并基于联合优化目标,本文采用深度强化学习方法对系统的资源分配进行优化,以获取最佳长期收益。

深度 Q-learning (deep Q-learning, DQL) 结构中 包含一个用于近似动作-状态值的深度神经网络层, 和一个用于动态网络更新和策略决策的在线深度 Q 网络层。基于 DQL 方法,智能体 agent 与环境交互 并获取经验,最终根据收敛的值函数执行智能策略。 具体地,在时刻 t , agent 观察环境状态 $s(t)$, 包含控 制器能量状态和服务器计算资源、计算开销状态。 随后, agent 选择并执行动作 $a(t)$, 包含卸载决策、 区块尺寸调整和服务器选择。同时,环境反馈 agent 即时奖励 $r(t)$, 并跳转至下一状态 $s(t+1)$ 。此过 程循环进行,直至环境达到最终状态 $s_{terminal}$ 。

由于 DQL 方法是一种基于值函数的优化算法, 在不同状态下准确地评价动作尤为关键。当前,一 种普遍用于评价动作优劣的方法,称为动作-状态值 $Q^\pi(s, a)$, 所示如下:

$$Q^\pi(s, a) = E^\pi \left[\sum_{k=1}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right] \quad (27)$$

其中, $E^\pi[\cdot]$ 表示数学期望, $\gamma \in (0, 1)$ 为奖励折扣 因子, r_{t+k+1} 为在时刻 $t+k+1$ 执行策略 π 获得的奖 励。

据此, agent 不断通过下式迭代动作-状态值, 表达式为

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (28)$$

其中, $\alpha \in (0, 1]$ 为学习率, $\max_{a'}$ 为 $Q(s', a')$ 为下一

状态的最大动作-状态值。

在所提优化问题中,状态空间和动作空间均复 杂高维,难以探索到所有状态以及获取每一组动作- 状态值。而深度神经网络通过调整网络中的参数, 具有从高维数据中获取低维特征的能力。据此,本 文应用深度神经网络生成近似值 $Q(s, a, \theta)$ 用以近 似动作-状态值 $Q(s, a)$ 。agent 将观察到的环境特 征输入深度神经网络,随后从网络中获取该环境状 态下所有动作的近似 Q 值作为网络输出。

DQL 方法的工作流程如图 3 所示,估计网络输 出近似值 $Q(s, a, \theta)$,并由损失函数 $L(\theta)$ 训练网 络中参数,通过调整网络层的权重和偏置 θ 以最小化 $L(\theta)$,从而达到反向训练。损失函数 $L(\theta)$ 表达式 为

$$L(\theta) = E[(r(s, a) + \gamma \max_{a'} Q(s', a', \theta) - Q(s, a, \theta))^2] \quad (29)$$

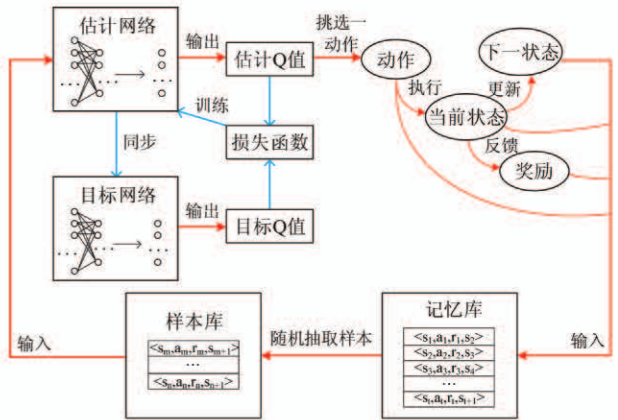


图 3 DQL 工作流程

然而,深度 Q 网络在训练中仍会存在不稳定的 情况,这有可能导致训练效果不佳,甚至训练失败。 针对此问题,本文增加了经验回放和固定目标网络 两种措施。

(1) 经验回放。为打破经验学习的时间相关性, agent 将探索经验 $\langle s_t, a_t, r_t, s_{t+1} \rangle$ 存入经验池 中。待学习时,从经验池中随机抽取样本输入估计 网络中,替代输入存在关联性的实时经验。

(2) 固定目标网络。目标网络和估计网络拥有 相同的网络结构,但相比估计网络中参数的实时更 新,目标网络将其网络中参数暂时冻结,以消除估计 网络输出的估计 Q 值和目标网络输出的目标 Q 值

间的相关性。当经过一定的训练步数后,目标网络将其参数与估计网络中参数同步。

因此,改进的损失函数 $L(\theta)$ 表达式为

$$L(\theta) = E[(r(s,a) + \gamma \max_{a'} Q(s', a', \theta^-) - Q(s,a,\theta))^2]$$

(30)

其中, θ^- 为目标网络中的权重和偏置参数。

综上,基于 DQL 的工业互联网系统资源分配决策算法的具体工作流程如算法 1 所示。

算法 1 基于 DQL 的工业互联网系统资源分配决策算法

1: 初始化:

初始化估计网络中的参数 θ

初始化目标网络中的参数 θ^-

初始化记忆库容量 M 和样本容量 B

初始化贪婪系数 ε

2: for episode = 1, K do

3: 随机重置控制器和服务器初始状态 $s_{initial}$, 并使 $s(t) = s_{initial}$

4: while $s(t) \neq s_{terminal}$ do

5: 根据概率 ε 随机选择动作 $a(t)$

6: 否则选择指定动作 $a(t) = \operatorname{argmax}_a Q(s(t), a, \theta)$

7: 执行动作 $a(t)$ 并获取即时奖励 $r(t)$, 观察下一状态 $s(t+1)$

8: 将经验 $\langle s_t, a_t, r_t, s_{t+1} \rangle$ 存入记忆库

9: 从记忆库中随机抽取样本容量 B 大小的经验 $\langle s_i, a_i, r_i, s_{i+1} \rangle$

10: 在估计网络中计算并输出估计 Q 值 Q_{eval}

11: 在目标网络中计算并输出目标 Q 值 Q_{target} :
if $s(t+1) = s_{terminal}$:
 $Q_{target} = r(t)$
else:
 $Q_{target} = r(t) + \gamma \max_{a'} Q(s(t+1), a', \theta^-)$

12: 训练估计网络中的参数以最小化损失函数
 $L(\theta) = E[(Q_{target} - Q_{eval})^2]$

13: 固定训练步数后将目标网络中参数同步为估计网络参数

14: 每隔一定训练步数,减小贪婪系数 ε

15: $s(t) \leftarrow s(t+1)$

16: end while

17: end for

其中 5~7 行是 agent 与环境的交互过程;8~9 行应用经验回放机制抽取训练样本;随后,10~13 行为估计网络的训练过程和目标网络的同步过程;5 和

13 行应用贪婪系数 ε 以平衡探索动作和偏好动作选择。

4 仿真实验及结果分析

本节首先介绍仿真实验的环境和参数,随后展示并讨论不同框架和不同方法下的相关仿真结果。

为更直观评估本文所提方法,将对下列框架、方法进行仿真分析:(1) 无卸载决策框架,计算任务全部在控制器本地执行;(2) 无区块尺寸调整框架,生成的区块尺寸固定;(3) 无服务器选择框架,随机挑选服务器支持节点的共识过程计算;(4) Q-learning 方法,基于 Q-learning 算法优化。

本文仿真实验环境为 Python3.6 和广泛用于部署机器学习算法的 Tensorflow1.31.1 环境。在网络场景中,考虑存在 6 个小区对应 6 个本地控制器,以及 3 个用于支持节点共识过程计算的服务器,包括 2 个 MEC 服务器和 1 个云计算服务器。此外,在区块链系统中部署有 4 个共识节点用于数据共识,场景中其他参数具体如表 1 所示。

表 1 实验参数设置

变量	参数
共识节点数量, N	4
MEC 服务器和云计算服务器数量, X, Y	2, 1
小区和本地控制器数量, U	6
验证签名和生成/验证 MAC 所需 计算轮数, θ, α	10 Mcycles, 0.1 Mcycles
固定区块生成间隔, t_i	0.25 s
交易平均尺寸, h	200 B
每个小区平均交易量, b	3 MB
本地控制器平均计算轮次, c_u	1 GHz/s
本地 MEC 服务器平均计算轮次, c_l	20 GHz/s
支持节点共识过程计算的 MEC 服务器和云计算服务器平均计算 轮次, c_m, c_c	20 GHz/s, 100 GHz/s
能耗和开销的权重系数, k_e, k_m	0.5, 0.5

同时,本文将用于支持节点共识过程计算的 MEC 服务器和云计算服务器的计算资源状态设置为 4 个等级,分别为很高、高、中等和低,其状态转移

矩阵表示为

$$R = \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0.2 & 0.1 & 0.1 & 0.2 \end{bmatrix} \tag{31}$$

此外,支持节点共识过程计算的 MEC 服务器和云计算服务器的计算开销状态分为 3 个等级,分别为昂贵、普通和廉价,其状态转移矩阵表示为

$$O = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \tag{32}$$

图 4 展示了不同方法优化效果的收敛性比较。在图中,随着迭代次数的增加,各方法优化曲线都在大约 2 500 次迭代后达到收敛,且相比于其他方法,本文所提方法优化后可得到更高的总奖励。主要原因是,合理的卸载决策和适当的区块尺寸有助于减少设备能耗和计算开销,且选择更高效廉价的服务器也可节省系统开销。

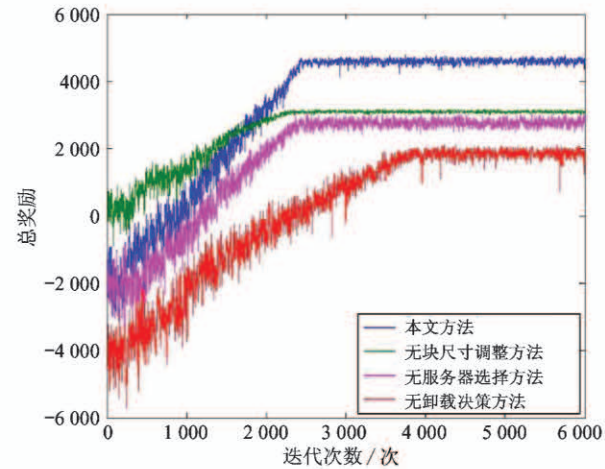


图 4 不同框架下曲线收敛性对比

此外,图 5 展示了本文所提出的 DQL 方法与 Q-learning 方法的优化曲线收敛性比较。如图所示,所提方法可达到更高的总奖励并具有更稳定的收敛性。其主要原因是,基于区块链的工业互联网系统中的状态动态高维,Q-learning 方法难以探索评估所有动作-状态对,其不完整的探索使 agent 很难获得最佳策略。相反,所提 DQL 方法通过深度神经网络评估不同状态下动作优劣,更加适应系统的高维复杂环境。因此,所提方法可取得更好的优化效果。

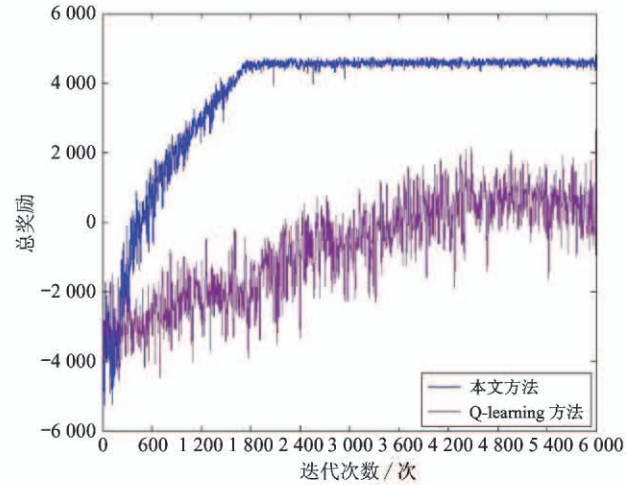


图 5 本文方法和 Q-learning 方法曲线收敛性对比

图 6 和图 7 展示了在不同框架、方法优化下,设备能耗与小区数量的关系对比图。如图所示,当小

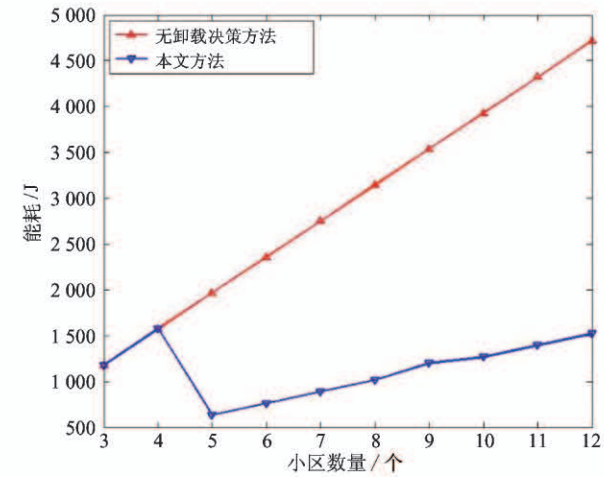


图 6 不同框架下设备能耗与小区数量关系图

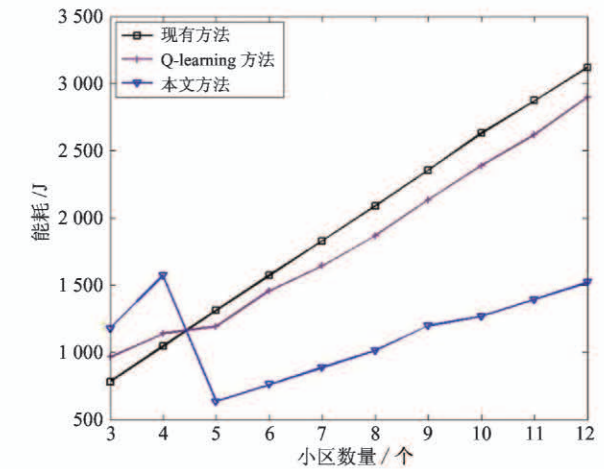


图 7 不同方法下设备能耗与小区数量关系图

区数量增加时,设备能耗也随之增加。同时,本文所提方法优化的设备能耗在小区数量大于4时,均明显低于其他方法。主要原因是,相比于其他方法,所提方法的卸载决策更合理,在平衡设备能耗和计算开销后决策是否卸载计算任务,有效缓解了设备在面临复杂计算任务时的能耗压力。

图8和图9展示了不同框架、方法下计算开销与小区数量的关系图。在图中,随着小区数量增加,计算开销也随之增加,但本文所提方法下的计算开销始终低于其他方法。主要原因是增加的计算任务量占有了更多的服务器计算资源,衍生出更多计算开销,且为承载更多的交易量,调整后生成了更大尺寸的区块。因此,计算开销明显增加。然而,本文方法适时地调整区块尺寸以及恰当地选择更廉价高效的服务器,有效节省了额外的计算开销。

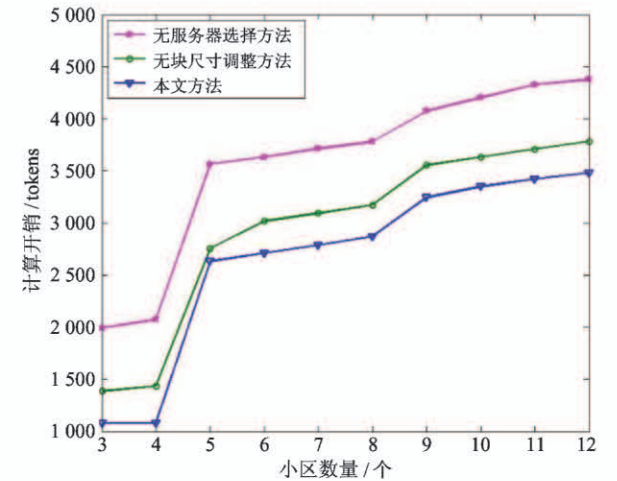


图8 不同框架下计算开销与小区数量关系图

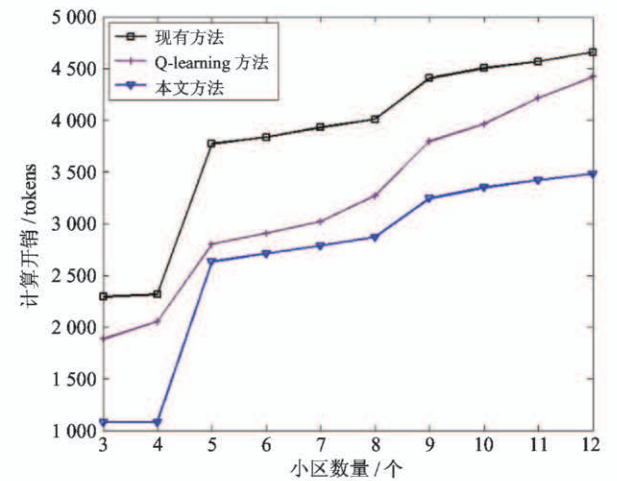


图9 不同方法下计算开销与小区数量关系图

图10和图11展示了不同框架、方法下系统能耗开销与小区数量的关系图。如图所示,在所有方法下,系统能耗开销均随着小区数量的增加而增长。系统能耗开销包含设备能耗和计算开销,由于存在对卸载决策、区块尺寸调整和服务器选择的联合优化,相比于已有方法,本文所提方法可取得更好效果。

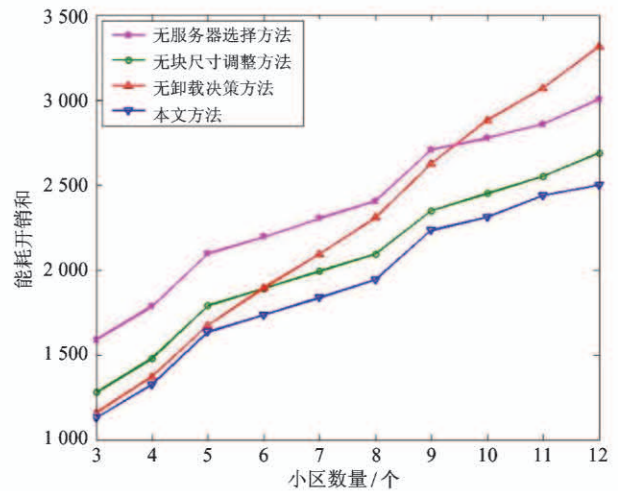


图10 不同框架下系统能耗开销与小区数量关系图

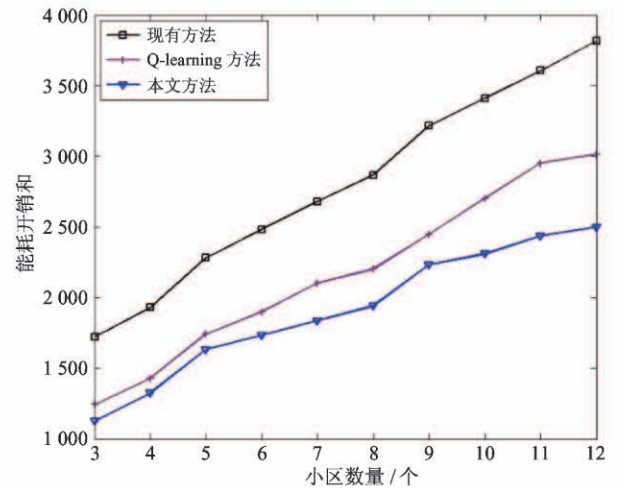


图11 不同方法下系统能耗开销与小区数量关系图

5 结论与展望

本文针对基于区块链的工业互联网系统,提出了一种设备能耗和系统计算开销联合优化的决策方法。本文将移动边缘计算和区块链技术整合应用于工业互联网场景中,联合考虑控制器和服务器的状态。同时,基于区块链协议和计算服务器资源构建

区块链共识时延模型,在保证数据安全性的基础上,提升了区块链共识和数据处理的效率。本文将此优化问题构造为马尔可夫决策过程,并根据优化场景的高动态、多维度的特点,采用更适合高维场景的深度强化学习方法予以建模求解。通过仿真结果证明,本文所提方法不仅具有更稳定快速的收敛性,还能更有效地减少设备能耗,节省计算开销,并提升区块链共识效率。

未来的工作将在现有工作基础上,考虑更多的区块链共识协议,并关注工业互联网中数据缓存等问题。

参考文献

- [1] Li J Q, Yu F R, Deng G Q, et al. Industrial Internet: a survey on the enabling technologies, applications, and challenges[J]. *IEEE Communications Surveys and Tutorials*, 2017, 19(3): 1504-1526
- [2] Li M, Yu F R, Si P B, et al. Green machine-to-machine (M2M) communications with mobile edge computing (MEC) and wireless network virtualization [J]. *IEEE Communication Magazine*, 2018, 56(5): 148-154
- [3] Xu L D, He W, Li S C. Internet of Things in industries: a survey[J]. *IEEE Transactions on Industrial Informatics*, 2014, 10(4): 2233-2243
- [4] Sisinni E, Saifullah A, Han S, et al. Industrial Internet of Things: challenges, opportunities, and directions[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(11): 4724-4734
- [5] Yao H P, Mai T L, Wang J J, et al. Resource trading in blockchain-based industrial Internet of Things[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(6): 3602-3609
- [6] Li M, Yu F R, Si P B, et al. Energy-efficient machine-to-machine (M2M) communications in virtualized cellular networks with mobile edge computing (MEC) [J]. *IEEE Transactions on Mobile Computing*, 2019, 18(7): 1541-1555
- [7] Yang X L, Fei Z S, Zhang J C, et al. Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(11): 11018-11030
- [8] Zhao J H, Li Q P, Gong Y, et al. Computation offloading

and resource allocation for cloud assisted mobile edge computing in vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 7944-7956

- [9] Dai H N, Zheng Z B, Zhang Y. Blockchain for Internet of Things: a survey[J]. *IEEE Internet of Things Journal*, 2019, 6(5): 8076-8094
- [10] Yang R Z, Yu F R, Si P B, et al. Integrated blockchain and edge computing systems: a survey, some research issues and challenges [J]. *IEEE Communications Surveys and Tutorials*, 2019, 21(2): 1508-1532
- [11] Zhang Z, Hong Z C, Chen W H, et al. Joint computation offloading and coin loaning for blockchain-empowered mobile edge computing[J]. *IEEE Internet of Things Journal*, 2019, 6(6): 9934-9950
- [12] Qiu X Y, Liu L B, Chen W H, et al. Online deep reinforcement learning for computation offloading in blockchain empowered mobile edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 8050-8062
- [13] Zhang X Y, Li C S, Shi H C, et al. AdapNet: adaptability decomposing encoder-decoder network for weakly supervised action recognition and localization [J]. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, doi: 10.1109/TNNLS. 2019.2962815
- [14] Zhang X Y, Wang S P, Yun X C. Bidirectional active learning: a two-way exploration into unlabeled and labeled dataset[J]. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2015, 26(12): 3034-3044
- [15] Zhang X Y, Shi H C, Li C S, et al. Multi-instance multi-label action recognition and localization based on spatio-temporal pre-trimming for untrimmed videos[C] // Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020: 12886-12893
- [16] Qiu C, Yu F R, Yao H P, et al. Blockchain based software-defined industrial Internet of Things: a dueling deep Q-learning approach [J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4627-4639
- [17] Feng J, Yu F R, Pei Q Q, et al. Cooperative computation offloading and resource allocation for blockchain-enabled mobile edge computing: a deep reinforcement learning approach[J]. *IEEE Transactions on Vehicular Technology*, 2020, 7(7): 6214-6228

Research on resource optimization and allocation for industrial Internet based on edge computing and blockchain

Yang Le^{***}, Li Meng^{***}, Ye Xinyu^{***}, Sun Enchang^{***}, Zhang Yanhua^{***}

(^{*} Faculty of Information Technology, Beijing University of Technology, Beijing 100124)

(^{**} Beijing Laboratory of Advanced Information Networks, Beijing 100124)

Abstract

Industrial Internet has emerged with the developments of various communication technologies. However, the requirement of data security and privacy in industrial Internet has brought unprecedented challenges to traditional architecture. As one of the promising technologies, blockchain has been applied into industrial Internet systems. Nevertheless, there are still some problems to be solved in blockchain-enabled industrial Internet systems, i. e. unbearable energy consumption for computation tasks, poor efficiency of consensus mechanism in blockchain, serious computation overheads of network systems. To handle the above issues and challenges, this paper integrates mobile edge computing (MEC) and blockchain into the industrial Internet systems to promote the computation capability of devices and improve the efficiency of consensus process. Meanwhile, the energy consumption and the computation overheads are jointly considered. Thus, a resource allocation optimization framework for blockchain-enabled industrial Internet systems is proposed to decrease energy consumption in devices and computation overheads of system. The optimization problem is formulated as a Markov decision process (MDP), and offloading decision, block size as well as computing server can be dynamically adjusted and selected. Accordingly, due to high-dynamic and large-dimensional characteristics of the system, the deep reinforcement learning is introduced to solve the formulated problem. Simulation results demonstrate that the proposed scheme can improve system performance significantly compared with other existing schemes.

Key words: industrial Internet, blockchain, deep reinforcement learning (DRL), mobile edge computing (MEC), resource allocation