

一种面向物联网的网络切片动态资源分配算法

李中捷^{1,2}, 潘麒名^{1a*}, 姜家祥¹

(1 中南民族大学 a. 电子信息工程学院; b. 智能无线通信湖北省重点实验室, 武汉 430074;
2 智能物联技术湖北省工程研究中心, 武汉 430074)

摘要 针对物联网(Internet of Things, IoT)场景下具有差异化资源需求的网络服务资源分配问题, 提出了一种将网络切片技术(Network Slicing, NS)与确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)相结合的动态切片资源分配算法(Dynamic Slicing Resource Allocation, DSRA). 该算法根据切片上不同设备的资源需求, 动态分配虚拟化的无线接入网资源, 以满足设备资源需求并最小化系统总成本. 仿真实验对比分析了所提出的算法与四种基线算法在服务质量(Quality of Service, QoS)满意率和系统总成本的表现. 仿真结果表明: 在具备多种资源与大量设备的场景中, 所提出的算法与基线算法相比, 能够显著提高设备的QoS水平, 并降低系统的总成本.

关键词 物联网; 网络切片; 资源分配; 深度强化学习

中图分类号 TN929.5 文献标志码 A 文章编号 1672-4321(2025)05-0639-08

doi: 10.20056/j.cnki.ZNMDZK.20250509

A network slicing dynamic resource allocation algorithm for the Internet of Things

LI Zhongjie^{1,2}, PAN Qiming^{1a*}, JIANG Jiexiang¹

(1 South-Central Minzu University a. College of Electronic and Information Engineering, b. Hubei Key Laboratory of Intelligent Wireless Communications, Wuhan 430074, China; 2 Hubei Engineering Research Center of Intelligent Internet of Things Technology, Wuhan 430074, China)

Abstract A Dynamic Slicing Resource Allocation algorithm (DSRA) based on Network Slicing (NS) and Deep Deterministic Policy Gradient (DDPG) is proposed to address the resource allocation problem of network services with differentiated resource demands in the Internet of Things (IoT). The algorithm dynamically allocates virtual radio access network resources based on the resource requirements of different devices on the slice, to meet device resource requirements and minimize the total system costs. The simulation experiment compared and analyzed the performance of the proposed algorithm with four baseline algorithms in terms of quality of service (QoS) satisfaction rate and total system cost. The simulation results show that in scenarios with multiple resources and a large number of devices, the proposed algorithm can significantly improve the QoS level of devices and reduce the total cost of the system compared to baseline algorithms.

Keywords IoT; network slicing; resource allocation; deep reinforcement learning

如今无线网络逐渐发展成为规模庞大的异构物联网(Internet of Things, IoT), 新兴的IoT应用使网络中连接的设备数量呈现指数级增长^[1], 应用场景日益多样化. 传统网络只能在一个物理网络基础设

施中实现单一的网络资源配置, 无法为运行在同一物理网络上的不同应用场景提供差异化的网络需求. 为应对物联网设备数量的大规模增长, 满足不同类型的设备或应用提出的服务质量需求, 下一代

收稿日期 2024-04-04 * 通信作者 潘麒名, 研究方向: 网络切片, E-mail: 934438401@qq.com

作者简介 李中捷(1974-), 男, 教授, 博士, 研究方向: 无线通信网络, E-mail: lizhongjie@mail.scuec.edu.cn

基金项目 国家自然科学基金资助项目(61379028); 湖北省自然科学基金资助项目(2022CFB905); 中央高校基本科研业务费专项资金资助项目(CZY23027)

移动网络联盟在2015年首次引入了网络切片技术的概念^[2]。网络切片技术将物理网络中的实际资源虚拟化,在同一物理基础设施中切分出多个包含网络功能、网络资源和特定配置的虚拟网络,每个虚拟网络能依据其所包含的不同业务按需提供网络资源,形成自己独特的容量、带宽、延迟和安全等特性^[3-4]。

目前,资源分配方案是网络切片领域中的研究热点,通过在切片之间动态调整网络中的资源,能提高网络性能和资源利用率,满足用户的QoS需求^[5]。文献[6]和文献[7]将物联网中资源分配问题建模为混合整数线性规划(Mixed Integer Linear Programming, MILP)问题。但由于资源分配问题的状态空间和动作空间都是非常复杂的,该优化问题通常是一个NP-hard问题,上述基于优化方案只能获得近似最优解,而文献[8-11]则采用启发式算法解决NP-hard问题。文献[8]采用MILP公式描述切片准入控制方案,设计了基于贪婪启发式方法来求解资源分配问题。文献[9]采用启发式方法计算多项式时间内每个切片的可行和次优资源分配。但上述文献侧重于最大化切片接受度和充分利用物理资源,忽视了系统节能问题。最小化总能源成本至关重要,且对最大化服务提供商的净利润起到积极影响。文献[10]针对如何最小化切片资源分配过程中产生的总能耗,提出了基于整数线性规划(Integer Linear Programming, ILP)和启发式算法的解决方案。文献[11]为最大化满足切片资源请求的总回报并最小化切片资源再分配的总惩罚,提出一个贪婪启发式算法进行求解。上述系统本质上都是静态或准静态的,对于5G网络中设备接入的高动态特性,这些资源分配方法在实际应用时需要根据情况的变化而重新求解,从而消耗大量的算力。面对动态的切片资源分配问题,基于机器学习的方法更具优势。

强化学习(Reinforcement Learning, RL)被认为是辅助实现切片资源动态分配的有力工具^[12]。文献[13]和文献[14]采用深度Q网络(Deep Q Network, DQN)的资源分配算法,提升资源利用率与服务水平协议满意率,但文献[13]并未考虑用户或设备的移动性,文献[14]缺乏针对IoT中众多差异化业务类型的适用性与扩展性。文献[15]将长短期记忆(Long Short-Term Memory, LSTM)模型与优势动作评论(Advantage Actor-Critic, A2C)算法相结合,解决动态资源分配问题,但未考虑分配资源时产生的系统成本。

为解决以上方法在应对IoT中庞大的设备数量与资源种类所面临的难题,本文在网络切片的基础上提出了一种基于深度强化学习的动态切片资源分配算法——DSRA。本文创新之处在于DSRA面向资源种类可扩展的海量设备物联网场景,对无线接入网中的切片进行动态资源分配,并以分配资源时产生的系统总成本与QoS满意率作为算法性能的评价指标。DSRA充分考虑了物联网场景中设备数量和位置变化的动态特性,采用深度强化学习的方法解决资源分配问题,在优化系统成本的同时保证设备的QoS需求。本文将DSRA与其他四种基线算法进行对比,并分别在资源受限和资源充足两种场景下,从QoS满意率、系统成本方面进行分析,仿真结果表明,基于深度强化学习的DSRA算法相较于其他算法,在不同的场景下均能发挥出更好的性能。

1 系统模型与问题表述

图1为一个具有多种IoT设备的区域性网络系统模型。在该系统中,基站 b 以星型拓扑分布在区域中,基站集合表示为 $B = \{b_1, b_2, \dots, b_j, \dots, b_j\}$ 。各种网络设备 d 按随机的方式分布在系统中的不同位置,设备集合表示为 $D = \{d_1, d_2, \dots, d_i\}$ 。系统中有 k 类资源 r ,资源集合表示为 $R = \{r_1, r_2, \dots, r_k\}$ 。考虑到物联网的动态性,并模拟设备数量和位置变化,本文将时间流划分为多个时隙 t ,时隙集合表示为 $T = \{1, 2, \dots, t, \dots, T\}$ 。

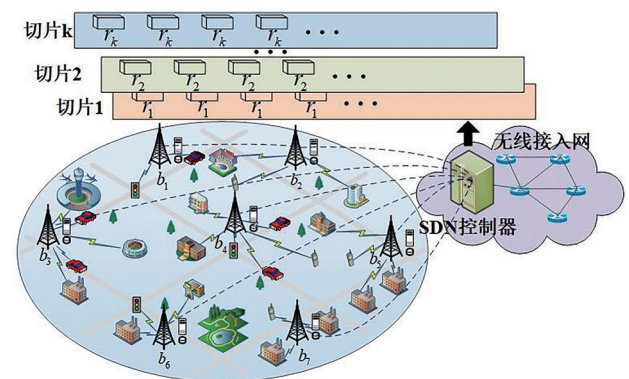


图1 系统模型

Fig. 1 System model

系统在初始化时,所有的设备均分布在固定位置并基于就近原则通过基站与网络连接。此外,系统在不同时隙下会随机激活基站上的设备,即同一时隙中活跃设备的数量保持不变,不同时间隙中活跃设备的数量发生改变,以此模拟更接近实时情况的场景。

系统运行期间,处于活跃的设备向系统请求一定数量的资源,用于满足设备运行期间的资源需求.为便于分析,本文假设每个设备仅向系统请求一类资源,且本工作可扩展到一个设备请求多类资源的情况.位于云端的集中式SDN控制器收集所有基站的运行信息,包括连接的活跃设备数量、系统的资源总量以及设备资源需求量等.根据收集的数据将无线接入网(Radio Access Network, RAN)中的资源进行划分,并模拟分配到不同的基站,以此建立多种不同类型的资源切片,供连接到该基站的设备使用.由此,整个系统的资源分配问题可表示为:

$$A^t = (A_{b,r}^t \in \mathbb{R}^+; t \in T, b \in B, r \in R), \quad (1)$$

式中: $A_{b,r}^t$ 为在时隙 t 下基站 b 所分配到的资源 r 的数量.

资源分配完成后,计算设备的QoS满意率和系统总成本,由此评估系统的性能.若资源分配量能满足设备的资源需求则表明系统满足了设备的QoS需求,QoS满意率为满足QoS需求的设备数量与活跃的设备数量的比值.本文引入的系统成本模型包括资源分配成本、资源调度成本、QoS违反成本以及系统收益,以此衡量资源切片与分配策略对系统性能的影响.

当系统向基站中的切片分配资源时会产生相应的开支,由此,可将资源分配成本 C_{op}^t 定义为:

$$C_{op}^t = \sum_{b \in B} \sum_{r \in R} W_{r,op} \cdot A_{b,r}^t, \quad (2)$$

式中: $W_{r,op}$ 为分配单位 r 资源时所需的成本.

当网络中的处于活跃状态的设备数量发生变化时,SDN控制器需要对切片的资源进行调整并重新配置,由此产生的资源调度成本 C_{re}^t 定义为:

$$C_{re}^t = \sum_{b \in B} \sum_{r \in R} W_{r,re} \cdot A_{b,r}^t, \quad (3)$$

式中: $W_{r,re}$ 为重新配置单位 r 资源时所需的成本.

当设备无法从系统中获取足够的资源来满足需求时,会造成一定的损失,将该损失即QoS违反成本 C_{vi}^t 定义为:

$$C_{vi}^t = W_{vi} \cdot \left(1 - \frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t} \right) \cdot g_{b,r}^t, \quad (4)$$

式中: W_{vi} 为单个设备由于所需资源得不到满足而产生的QoS违反成本; $y_{d,b}^t \in \{0, 1\}$ 为二进制变量,当 $y_{d,b}^t=1$ 时表示设备 d 接入了基站 b 并处于活跃状态,当 $y_{d,b}^t=0$ 时则反之; $\lambda_{d,r}^t$ 为设备 d 请求的资源 r 的数量; $g_{b,r}^t$ 为连接到基站 b 且处于活跃状态的设备中,对资源 r 有需求的设备总数.

系统总成本的大小与资源分配策略密切相关,

较多的资源能够满足物联网设备的QoS需求,但是过度的资源分配也会产生很高的系统成本.因此,为了能在最小化系统总成本的同时尽可能满足设备的QoS需求,本文将设备的QoS水平作为一种奖励机制加入到系统成本中.当系统能为设备提供充足的资源时,即设备的QoS需求得到了满足,将产生相应的收益,该系统收益 R^t 定义为:

$$R^t = V \cdot \sum_{b \in B} \left(\frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t} \cdot g_{b,r}^t \right), \quad (5)$$

式中: V 为单个设备的QoS需求得到满足而产生的收益.

当切片中的资源能够满足所有设备需求的资源总和时,为保证不产生负的成本且不降低系统收益,(4)-(5)式需要满足以下约束条件:

$$\frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t} = \begin{cases} \frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t}, & \frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t} \leq 1 \\ 1, & \frac{A_{b,r}^t}{\sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t} > 1 \end{cases}. \quad (6)$$

结合上述成本和收益,将系统在时隙 t 的总成本 C_{sum}^t 表示为:

$$C_{sum}^t = C_{op}^t + C_{re}^t + C_{vi}^t - R^t. \quad (7)$$

由于系统中处于激活状态的设备数量在时刻发生变化,每个时隙产生的系统总成本 C_{sum}^t 也随着资源配置的不同而发生改变,因此,整个系统追求的是最小化的长期运行成本,可将其目标函数表示为:

$$\begin{aligned} & \arg \min_{A_{b,r}^t} E \left[\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T C_{sum}^t \right] \\ & \text{s.t. } C1: A_{b,r}^t \in \mathbb{R}^+, \forall b \in B, r \in R, t \in T \\ & \quad C2: A_{b,r}^t \leq A_{b,r}^{\max}, \forall b \in B, r \in R, t \in T \\ & \quad C3: A_{b,r}^t \geq \sum_{d \in D} y_{d,b}^t \cdot \lambda_{d,r}^t, \forall b \in B, r \in R, t \in T, \quad (8) \\ & \quad C4: \sum_{b \in B} A_{b,r}^t \leq MAX_r, \forall r \in R, t \in T \\ & \quad C5: \text{公式(6)} \end{aligned}$$

式中: $A_{b,r}^{\max}$ 为基站 b 中 r 资源的总量; MAX_r 为系统中的 r 资源总量.其中约束条件C1保证了资源分配量不为负值,约束条件C2保证了资源分配量不超过该基站中该资源的可容纳上限,约束条件C3保证了满足设备QoS的条件,约束条件C4保证了资源分配总量不超过系统中的资源总量.

2 DSRA 动态切片资源分配算法

网络切片的资源分配问题实际上就是在变化的环境中不断做出决策的过程,因此,基于强化学习的资源分配算法能够发挥出其他算法所不具备的优势.资源分配是切片网络的核心任务之一,动态高效的资源分配算法对优化(8)式中的目标函数起到决定性的作用.鉴于物联网中的资源种类和设备数量不断增加,系统中的所有状态和资源的所有分配方式都是无穷大的集合,因此在约束条件下优化系统长期运行成本的问题,可以被视为一种无限空间下的马尔科夫决策过程(Markov Decision Process, MDP)^[16].因此,本文将系统中的资源分配过程建模为一个MDP,位于云端的集中式SDN控制器作为智能体,采用深度强化学习方案来执行资源分配的决策.

在MDP中,基于每个时隙 t ,智能体首先通过观测获取系统的当前状态 s^t ,并针对该状态由强化学习算法决策出该时隙下最佳资源分配的动作 a^t ,并将该动作施加到环境中;随后得到系统反馈的信息,包括由该动作完成后得到的奖励 r^t 以及根据状态转移概率 $P(s^{t+1}|s^t, a^t)$ 到达的系统下一状态 s^{t+1} .其中,状态、动作、和奖励的定义分别如下.

在本文的系统模型中,智能体获取环境的状态主要依靠基站汇总的状态信息 $\lambda_{a,r}$ 与 $g_{b,r}^t$,在实际场景中,详细统计每个设备对资源需求的种类和数量

需要花费大量时间,并且设备的资源需求量也可能临时变化,因此用 $g_{b,r}^t$ 作为状态更加合适;此外,基站在上个时隙所分得的资源 $A_{b,r}^{t-1}$ 也会间接影响下一时隙的资源分配决策.因此,将时隙 t 下的状态 s^t 定义为:

$$s^t = \{ \{ g_{b,r}^t \}_{b \in B, r \in R, t \in T}, \{ A_{b,r}^{t-1} \}_{b \in B, r \in R, t \in T} \}. \quad (9)$$

动作是由智能体做出的切片资源分配决策,即在时隙 t 下对每个基站各类资源的分配数量 $A_{b,r}^t$,则动作 a^t 定义为:

$$a^t = \{ \{ A_{b,r}^t \}_{b \in B, r \in R, t \in T} \}. \quad (10)$$

奖励用于表征智能体做出动作的优劣,基于(8)式的最小化系统总成本的目标,将奖励设置为负的系统总成本,即鼓励智能体做出能降低系统总成本的动作,因此,时隙 t 下的奖励 $r^t(s^t, a^t)$ 定义为:

$$r^t(s^t, a^t) = -(C_{\text{sum}}^t). \quad (11)$$

在强化学习具体算法的选择上,本文采用了DDPG作为智能体的核心算法.DDPG是一种基于动作价值函数的Actor-Critic框架深度强化学习算法,通过单步更新的方式快速学习输出策略.在其结构中,Actor和Critic分别由各自的评价网络和目标网络构成.Actor负责基于当前状态做出资源分配决策,Critic负责通过计算状态动作价值来评价演员策略的好坏,并指导Actor网络的更新方向.此外,DDPG也借鉴了DQN中的回放经验池Buff,将与环境的互动过程作为经验记录下来,采用离线策略的方式更新网络.图2为本文的算法模型.

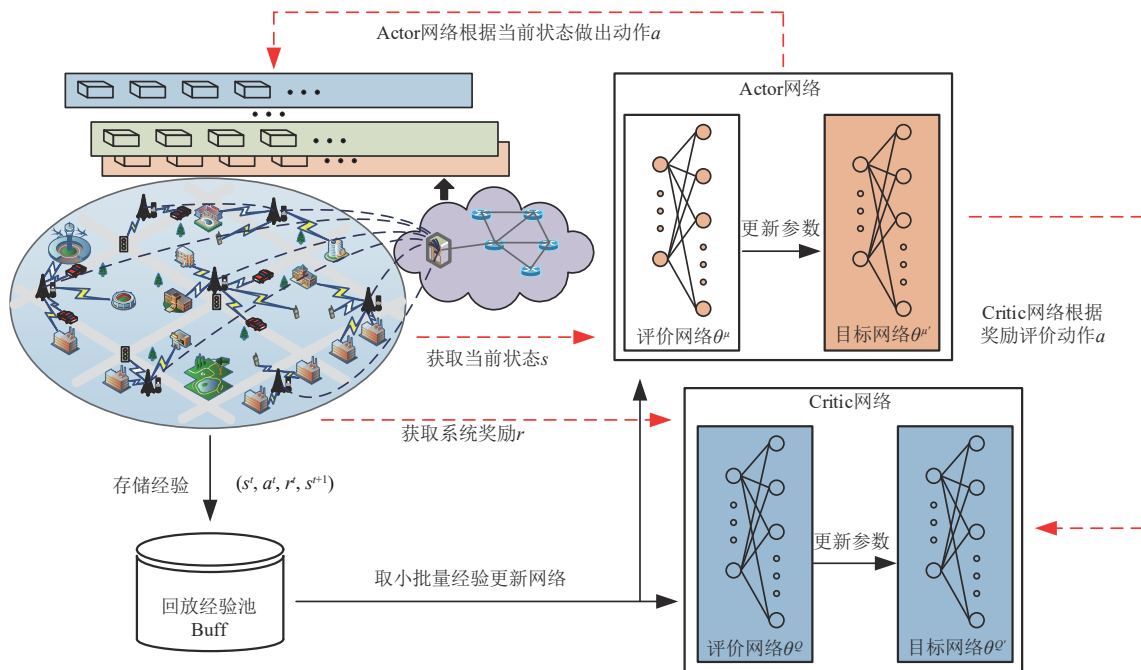


图2 基于DDPG的算法模型

Fig. 2 Algorithmic model based on DDPG

该算法分为三个阶段:初始化阶段、经验积累阶段以及网络更新阶段.

初始化阶段主要对网络参数进行初始化,包括 Actor 的评价网络 $\mu(s|\theta^\mu)$ 和 Critic 的评价网络 $Q(s, a|\theta^Q)$ 会被初始化,其中 θ^μ 和 θ^Q 代表网络的参数, Actor 和 Critic 对应的目标网络分别用 $\mu(s|\theta^\mu)$ 和 $Q(s, a|\theta^Q)$ 表示,其中的网络参数 θ^μ 和 θ^Q 也需要被初始化.

经验积累阶段主要由 Actor 与环境交互并积累历史数据.在这一过程中,对于每个时隙 t , Actor 会先基于当前状态 s^t 做出资源分配决策,即动作 $a^t = \{A^t\}$.由于 DDPG 输出的是确定性策略,为尽可能多地对策略进行探索,本文为动作 a^t 加入加性高斯白噪声,表示为:

$$a^t = \mu(s|\theta^\mu) + \varepsilon, \quad (12)$$

式中: $\varepsilon \sim N(0, \sigma^2)$ 为加性高斯白噪声.随后将动作施加到环境当中,得到奖励 r^t ,同时环境进入下一状态 s^{t+1} .最后,智能体将状态、动作和奖励组合成一个多维元组 $\{s^t, a^t, r^t, s^{t+1}\}$ 作为一条经验存储到系统的回放经验池 *Buff* 中.

网络更新阶段采用离线策略更新网络参数.当有一定数量的经验后,首先从回放经验池 *Buff* 中随机选取 N_m 条经验组成 mini-batch,再计算 Critic 网络的损失函数并更新其评价网络 θ^Q 的参数, Critic 网络的损失函数定义为:

$$L(\varphi) = \frac{1}{N_m} \sum_{n=1}^{N_m} (y_n - Q(s^n, a^n|\theta^Q))^2, \quad (13)$$

式中: $y_n = r^n + \gamma Q(s^n, a^n|\theta^Q)$, 为由 Critic 的目标网络得到的更新目标; $\gamma \in (0, 1)$, 为折扣系数.

随后在 Critic 的指导下,通过梯度上升的方式对 Actor 的评价网络 θ^μ 进行参数更新,梯度更新公式表示为:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N_m} \sum_{n=1}^{N_m} \nabla_a Q(s, a|\theta^Q)|_{s=s^n, a=\mu(s|\theta^\mu)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s^n}. \quad (14)$$

最后,通过软更新的方式更新 Critic 与 Actor 的目标网络参数,更新公式表示为:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}, \quad (15)$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}, \quad (16)$$

式中: $\tau \in [0, 1]$ 为目标网络的更新率.

基于 DDPG 的 DSRA 算法具体流程如表 1 所示.

表1 DSRA算法

Tab. 1 DSRA algorithm

算法1: DSRA算法

输入: $\sigma, \gamma, \tau, M, N_m, \text{Critic}$ 网络与 Actor 网络的学习率;
 输出: 资源分配策略 $\{A^t\}_{t \in \mathcal{T}}$;

初始化: Actor 网络 $\mu(s|\theta^\mu)$ 和 $\mu(s|\theta^\mu)$, Critic 网络 $Q(s, a|\theta^Q)$ 和 $Q(s, a|\theta^Q)$, Buff;

- 1.
2. for episode = 1: M do
3. 重置环境并获取初始状态 s^0 ;
4. for $t = 1, 2, \dots, T$ do:
5. 经验积累:
6. 统计 $g_{b,r}^t$, 获取环境的当前状态 s^t ;
7. Actor 做出动作并加噪, 得到实际动作 a^t ;
8. 与环境交互, 得到奖励 r^t 与下一状态 s^{t+1} ;
9. 将经验 $\{s^t, a^t, r^t, s^{t+1}\}$ 存入经验池 Buff;
10. 网络更新(积累一定数量的经验后):
11. 在经验池 Buff 中随机选取 N_m 条经验组成 mini-batch;
12. 最小化(13)式更新 Critic 评价网络参数 θ^Q ;
13. 通过(14)式更新 Actor 评价网络参数 θ^μ ;
14. 通过(15)–(16)式对 Actor 和 Critic 的目标网络参数 $\theta^{Q'}$ 和 $\theta^{\mu'}$ 进行软更新;
15. end for;
16. 每隔固定数量的 episodes 保存模型与参数;
17. end for;

3 仿真实验与结果分析

3.1 仿真设置与对比算法

本文分别在资源受限场景与资源充足场景下进行仿真.在资源受限场景中,每一类资源总量能够满足所有设备对该类资源的平均需求,设定系统中每类资源总量为 140.在资源充足场景中,系统则会对所有类别的资源提供充足的资源总量,设定系统中每类资源总量为 280.为体现真实场景下设备数量与资源种类的关系,系统中的设备数量会随着资源种类 k 的增加而增加,且不同 QoS 等级的设备数量都 100.当系统中有一类资源即 $k=1$ 时,设备数量 $i=100$,按设备 QoS 等级划分一类切片.以此递增,当 $k=5$ 时,设备数量 $i=500$,按设备 QoS 等级划分五类切片.两种场景在相同的仿真区域中进行,仿真区域的大小设定为 $1 \text{ km} \times 1 \text{ km}$,其中包含 7 个地面基站与 1 个集中式 SDN 控制器,基站按图 1 分布在仿真区域中,每个基站的信号覆盖半径设置为 300 m ,能够覆盖整个区域.

本文使用 ReLU 作为网络模型的激活函数,Adam 作为网络模型的优化函数.详细的场景设置与仿真参数如表 2 所示,是经过前期实验的调参工

作得到的最佳参数设置.

表2 场景设置与仿真参数
Tab. 2 Scene settings and simulation parameters

参数	值	参数	值
训练轮次 M	500	经验池最大容量 Buff	100000
时隙数 T	24	Batch size N_m	128
基站数量 J	7	折扣系数 γ	0.8
系统中资源 r 的总量 MAX_r	140、280	目标网络更新率 τ	0.01
基站 b 中资源 r 的可容纳上限 $A_{b,r}^{max}$	20、40	Critic 学习率	0.001
设备 d 请求资源 r 的数量 $\lambda_{d,r}$	1~2	经验池最大容量 Buff	100000
单位资源分配成本 $W_{r,op}$	1	Actor 隐藏单元	(200, 100)
单位资源调度成本 $W_{r,re}$	0.3	Actor 学习率	0.0001
QoS 违反的单位损失 W_{vi}	2	更新率 σ	0.01
QoS 满足的单位收益 V	0.5		

本文仿真对比了如下四种基线算法:

(1) 理论最优 (Optimal). 该算法基于网络切片架构, 在不超过各类资源的系统总量与基站容量的前提下, 严格按照每个设备对特定资源的需求量对基站进行资源分配, 以实现最高的 QoS 满意率与最低的系统的总成本. 但在实际场景中, 由于基站容量的限制, 且难以实现每时每刻精确统计所有设备的资源需求, 因此只是理论上的最优.

(2) 数量优先 (AFA). 该算法在无网络切片架构下实现资源分配, 系统资源按照基站各类资源的总需求量进行分配. 可以避免由于资源的过度分配而产生的资源浪费, 但会造成某类或多类资源的需求量与分配量不相符, 导致 QoS 满意率降低.

(3) 需求优先 (RFA). 该算法基于网络切片架构, 不考虑系统成本, 始终按所有设备的最大需求量对各类资源进行分配, 优先满足设备对各种资源的需求. 但该算法不可避免产生资源浪费.

(4) 随机 (Random). 在满足系统限制的前提下, 随机分配资源.

3.2 仿真结果分析

在资源受限场景中, 采用 DSRA 算法, 不同资源种类数量 k 的系统总成本的下降过程如图 3 所示. 系统中的资源种类数量不同, 系统的总成本会有较大的差异, 因此本文采用归一化的成本, 将训练过程中出现的最高系统总成本作为归一化范围的上限, 将算法稳定后的多轮训练平均总成本作为归一化范围的下限, 以此方式来体现 DSRA 算法的收敛性. 图 3 表明, 随系统中资源种类增加, 状态空间和动作空间变得越来越大, 寻找最佳的资源分配动作需要更多的探索过程和更长的学习时间, 因此需要更多的训练轮次来达到收敛. 但所有的曲线都能在 500

个 episodes 以内完全收敛. 因此该算法能够适应 IoT 资源种类与设备数量繁多的特点, 具备较强的扩展性. 此外, 在资源充足场景下, 由于设备的 QoS 非常容易得到满足, DSRA 算法只需要重点完成降低系统成本的任务, 因此在该场景下, 算法的收敛速度要快于资源受限的场景.

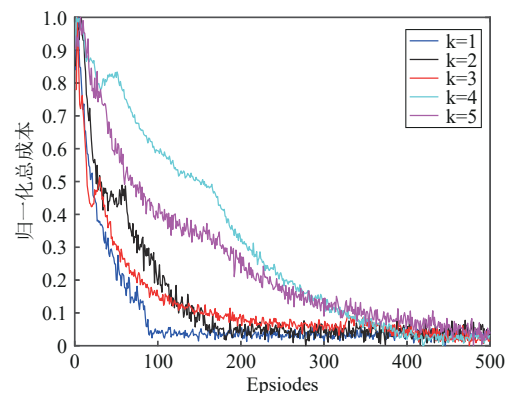


图3 DSRA 在不同资源种类数量下的收敛性

Fig. 3 The convergence of DSRA under the different numbers of resource types

在资源受限场景下, 不同算法的系统总成本对比和 QoS 满意率对比如图 4 所示. 图 4(a) 中, 当 $k=1$ 时, 除 Random 外都能其他算法都几乎能满足设备的资源需求. 随资源种类增加, AFA 算法只有不到 82% 的 QoS 满意率, 而 Optimal、RFA 和 DSRA 算法的 QoS 满意率仍维持在 90% 以上. 图 4(b) 中, 随资源种类增加, 各种算法产生的系统总成本不断升高, DSRA 算法的成本控制要明显优于 AFA 与 RFA 算法, 当 $k=5$ 时, DSRA 算法相较于 RFA 算法能够降低 21% 的系统总成本. 综合上述算法的表现, RFA 算法是以更高的系统总成本换取比 DSRA 算法更高的 QoS 满意率, 因此 DSRA 算法是在资源受限场景中

更优的动态资源分配算法.

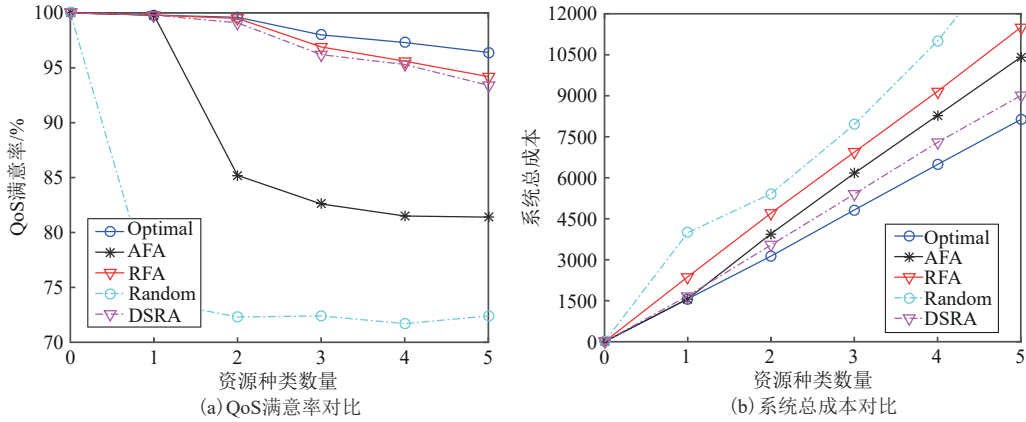


图 4 资源受限场景下不同算法的性能对比

Fig. 4 Performance comparison of different algorithms in confined resource scenarios

在资源充足的场景下,除 Random 和未对资源进行切片的 AFA 算法外,其余算法均能够将 QoS 满意率维持在接近 100% 的水平,所以在该场景下重点关注资源过度分配带来的系统成本上升问题.资源充足场景下的系统总成本如图 5(a)所示.相较于资源受限场景中不同 k 的条件下,AFA、RFA 和

DSRA 算法的总成本有所增加,而 Optimal 算法用极少的资源分配成本增加了满足设备资源需求的数量,获得了较大的系统收益,因此降低了系统的总成本.DSRA 算法虽然有更高的成本提升,但在系统总成本上仍然优于 AFA 和 RFA 算法,最接近于 Optimal 算法.此外,图 5(b)给出了不同算法在资源

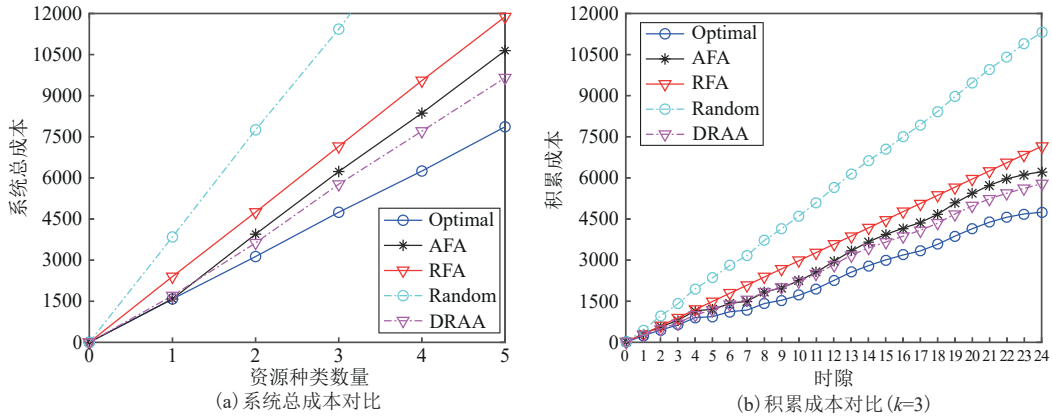


图 5 资源充足场景下不同算法的性能对比

Fig. 5 Performance comparison of different algorithms in adequate resource scenarios

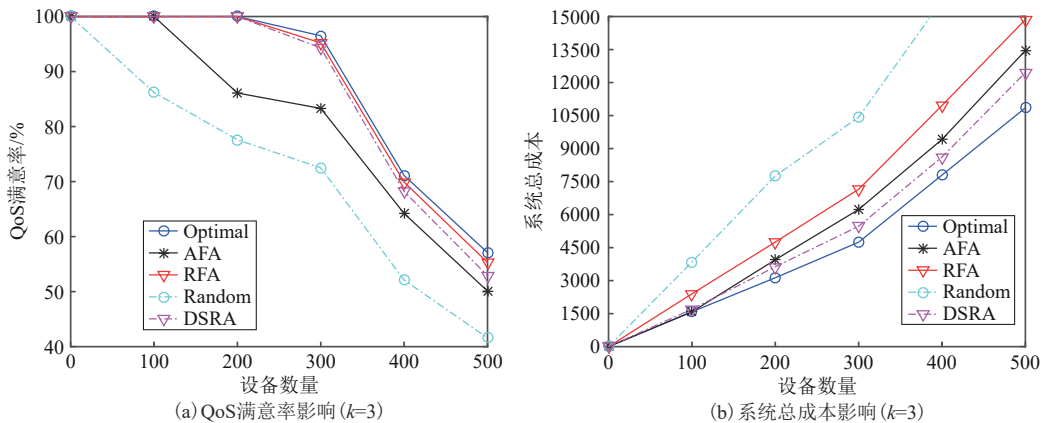


图 6 设备数量对系统性能的影响

Fig. 6 The impact of equipment quantity on system performance

充足场景的累积成本对比,从中可以看出 DSRA 算法在每个时隙都能做出比 AFA 算法与 RFA 算法更好的资源分配决策,进而实现更低的系统总成本.综上所述,DSRA 算法在资源充足场景中的综合表现更优.

图 6 给出了系统中的设备数量对 QoS 满意率和系统总成本的影响,以 $k=3$ 的系统资源总量作为标准.由图 6(a)可以看出,当设备数量接近 300 时,系统资源已经无法满足所有设备的需求,当资源总量固定而设备数量进一步增加时, QoS 满意率呈断崖式下降.由图 6(b)可以看出,当设备数量增加到 300 后,固定的资源总量无法满足越来越多的设备资源需求,系统总成本快速升高.仿真结果表明,本文提出的 DSRA 算法在资源极度匮乏的场景下,仍能保持 QoS 满意率和系统总成本维持在最接近 Optimal 算法的水平,具备较强的环境变化适应能力.

4 结语

本文研究了物联网场景下 RAN 网络切片的动态资源分配问题.首先根据物联网的动态性构建系统模型,并将网络切片的资源分配问题建模为无限空间下的马尔科夫决策过程;其次提出了一种基于深度强化学习的动态切片资源分配算法 DSRA,按需制定资源分配策略;最后设计了一套基于成本收益的性能评估方案,并对所提出的 DSRA 算法进行性能对比,证明了在不同资源规模的场景下,DSRA 算法在提升网络设备 QoS 满意度和降低系统总成本方面均有明显优势,接近于理论最优.进一步将实现多智能体分布式动态切片资源分配方案,这也将是未来研究的重点方向.

参 考 文 献

- [1] WIJETHILAKA S, LIYANAGE M. Survey on network slicing for Internet of Things realization in 5G networks[J]. IEEE Communications Surveys & Tutorials, 2021, 23(2): 957-994.
- [2] PORAMBAGE P, OKWUIBE J, LIYANAGE M, et al. Survey on multi-access edge computing for Internet of Things realization[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2961-2991.
- [3] CAO H, DU J, ZHAO H, et al. Toward tailored resource allocation of slices in 6G networks with softwarization and virtualization[J]. IEEE Internet of Things Journal, 2022, 9(9): 6623-6637.
- [4] 刘彩霞, 胡鑫鑫. 5G 网络切片技术综述[J]. 无线电通信技术, 2019, 45(6): 569-575.
- [5] HARUTYUNYAN D, FEDRIZZI R, SHAHRIAR N, et al. Orchestrating end-to-end slices in 5G networks[C]//2019 15th International Conference on Network and Service Management (CNSM). Halifax:IEEE, 2019: 1-9.
- [6] YIN B, TANG J, WENT M. Maximizing the connectivity of wireless network slicing enabled industrial Internet-of-things[C]//2021 IEEE Global Communications Conference (GLOBECOM). Madrid: IEEE, 2021: 1-6.
- [7] 李悦, 任春莉, 章国安. 车联网中网络切片资源分配方案[J]. 电讯技术, 2023, 63(1): 85-92.
- [8] DEBBABI F, TAKTAK R, JMAL R, et al. Inter-slice B5G bandwidth resource allocation[C]//2022 IEEE 21st International Symposium on Network Computing and Applications (NCA). Boston:IEEE, 2022: 157-163.
- [9] CAO H, DU J, ZHAO H, et al. Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization [J]. IEEE Transactions on Vehicular Technology, 2021, 70(4): 3846-3859.
- [10] JANG I, SUH D, PACK S, et al. Joint optimization of service function placement and flow distribution for service function chaining[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(11): 2532-2541.
- [11] BERA S, MEHTA N B. Network slicing in 5G edge networks with controlled slice redistributions[C]//2021 17th International Conference on Network and Service Management (CNSM). Izmir:IEEE, 2021: 118-124.
- [12] SHEN X, GAO J, WU W, et al. AI-assisted network-slicing based next-generation wireless networks [J]. IEEE Open Journal of Vehicular Technology, 2020, 1: 45-66.
- [13] LI R, WANG C, ZHAO Z, et al. The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility [J]. IEEE Communications Letters, 2020, 24(9): 2005-2009.
- [14] HUA Y, LI R, ZHAO Z, et al. GAN-powered deep distributional reinforcement learning for resource management in network slicing [J]. IEEE Journal on Selected Areas in Communications, 2020, 38(2): 334-349.
- [15] HAO M, YE D, WANG S, et al. URLLC resource slicing and scheduling in 5G vehicular edge computing [C]//2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring). Helsinki:IEEE, 2021: 1-5.
- [16] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv:2015.1509.02971.