

基于深度Q网络的在线服务功能链部署方法

邱 航 汤红波* 游 伟

(中国人民解放军战略支援部队信息工程大学 郑州 450002)

摘 要: 针对5G网络资源状态动态变化和网络模型高维度下服务功能链部署的复杂性问题, 该文提出一种基于深度Q网络的在线服务功能链部署方法(DeePSCD)。首先, 为描述网络资源动态变化的特征, 将服务功能链部署建模成马尔可夫决策过程, 然后, 针对系统资源模型的高维度问题采用深度Q网络的方法进行在线服务功能链部署策略求解。该方法可以有效描述网络资源状态的动态变化, 特别是深度Q网络能有效克服求解复杂度, 优化服务功能链的部署开销。仿真结果表明, 所提方法在满足服务时延约束条件下降低了服务功能链的部署开销, 提高了运营商网络的服务请求接受率。

关键词: 网络功能虚拟化; 服务功能链; 马尔可夫决策过程; 深度Q网络

中图分类号: TN915.07

文献标识码: A

文章编号: 1009-5896(2021)11-3122-09

DOI: 10.11999/JEIT201009

Online Service Function Chain Deployment Method Based on Deep Q Network

QIU Hang TANG Hongbo YOU Wei

(Information Engineering University People's Liberation Army Strategic Support Force,
Zhengzhou 450002, China)

Abstract: To deal with the dynamic nature of 5G network resource state and the difficulty of service function chain deployment under the high-dimensional network state model, an online Service function Chain Deployment method based on Deep Q network (DeePSCD) is proposed. First, to describe the dynamic nature of network resource state, the service function chain deployment is modeled as a Markov decision process. Then, the deep Q network is used to solve the online service function chain deployment problem in the high-dimensional system resource model. This method can effectively describe the dynamic changes of network resource state. Specifically, deep Q network handles the complexity of problem and determines the optimal deployment solution of service function chain. Simulation results show that the proposed method can reduce the deployment cost of the service function chain and increase the acceptance rate while meeting the service delay constraint.

Key words: Network function virtualization; Service function chain; Markov decision process; Deep Q network

1 引言

随着移动通信的发展, 快速部署各种新出现、多样化和差异化的网络服务已经成为运营商们面临的重要挑战之一^[1]。为了满足这些需求, 第5代移动通信(5G)从僵化的硬件解决方案转变为更加灵活和可扩展的软件解决方案, 降低了运营商们在采

购、管理和运行期间的资本开销(CPEX)和运营支出(OPEX)。这一愿景的实现主要受益于网络功能虚拟化和软件定义网络两项关键技术, 使得按需和近乎实时地部署新服务成为可能^[2]。网络功能虚拟化(Network Function Virtualization, NFV)^[3]实现了网络设备中软件与硬件解耦, 提供了一种新的方式设计、编排、部署和管理多样化的网络服务; 同时, 软件定义网络(Software-Defined Network, SDN)解耦了控制平面与数据平面, 以SDN控制器的形式实现了网络的集中化管理^[4]。在NFV/SDN使能网络中, 运营商能够便捷地监测网络设备和流量, 实现网络服务的高效管理。

收稿日期: 2020-12-02; 改回日期: 2021-06-30; 网络出版: 2021-08-10

*通信作者: 汤红波 hangsoon@foxmail.com

基金项目: 国家自然科学基金(61801515, 61941114, 61521003)

Foundation Items: The National Natural Science Foundation of China (61801515, 61941114, 61521003)

在NFV/SDN使能网络中, 服务功能链(Service Function Chain, SFC, 简称服务链)已经成为一种典型的网络服务表示方式。根据IETF (Internet Engineering Task Force)制定的服务链标准, SFC定义为有序或部分有序的VNFs集合, 特别是一条SFC的流量需按照预定义的顺序经过一系列指定的VNFs^[5]。NFV允许将软件化的VNF部署在通用服务器里任何资源充足的VM或容器中, 因此, 在NFV/SDN使能网络中, 通过确定如何在多个候选服务器中部署服务请求的服务链为改善系统性能和服务质量提供了一个机遇。同时, 网络服务的高效和自动部署无疑是迈向全自动网络(即零接触网络)的最重要的技术构件之一。然而, 由于服务的各种限制条件, 网络服务在基础设施上的部署仍然非常复杂。

目前, 面向服务请求的虚拟网络功能放置或者服务链部署问题已经成为学术界研究的一个热点问题, 被归类为NFV系统中的一个资源管理问题^[6], 并被证明是一个NP难问题^[7-9]。当前文献主要将其建模为不同优化目标的数学规划模型, 包括整数线性规划(Integer Linear Programming, ILP)^[10,11]和混合整数线性规划(Mixed Integer Linear Programming, MILP)^[12,13]等。由于数学规划方法在大规模网络中求出最优解是非常困难的, 因此许多工作通过设计启发式算法以获得近似最优解。文献^[11]研究了数据中心中服务链的部署问题, 通过考虑VNFs的资源开销设计了一种启发式算法完成VNF实例在大规模网络中的放置。文献^[14]提出了一种改进多阶段图的启发式服务链部署算法, 完成VNF的高效放置。在考虑流量预测的条件下, Tang等人^[13]提出一种滑动窗口线性回归的流量预测方法, 然后通过松弛整数变量设计了两两种启发式算法。然而, 由于启发式算法缺乏严格的理论证明, 难以保证其总是求得近似最优解。强化学习(Reinforcement Learning, RL)用于描述和解决智能体(agent)在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题, 目前已证明其在求解组合优化问题方面的优势^[15]。袁泉等人^[16]提出了一种改进的基于Q学习的服务链部署方法, 同时考虑了空间维度下的服务链映射和时间维度下的VNF生命周期管理, 实现了VNF部署收益和时延的多目标优化。在现实网络环境中, 一个网络中包含大量的硬件设备, 由于网络服务和用户请求的多样性, 网络资源状态(例如带宽、内存和CPU等)的变化是复杂的, 将会对VNF的部署产生重要影响; 而且, 在网络功能虚拟化环境中, VNF能够灵活地放置在网络

的多个位置, 其可能产生大规模VNF放置和优化的动作空间。传统的强化学习方法采用Q表的形式评价当前状态下每个动作的性能, 然而Q表的维度是有限的, 仅能求解有限离散状态和动作空间的低维度问题。因此, 传统的强化学习算法在大规模底层网络环境中, 难以准确地描述复杂的网络资源状态变化, 且求解扩展性不足。深度强化学习(Deep Reinforcement Learning, DRL)采用深度神经网络代替Q表, 深度神经网络能够建立高维状态、动作和Q值之间的关系。因此, DRL拥有更加强大的学习能力, 被广泛地应用于解复杂问题。文献^[17]设计了一种基于策略梯度的DRL算法求解服务链部署问题, 目标是实现运营开销和服务请求总吞吐量的联合优化。文献^[18]将深度确定性策略梯度算法用于求解VNF的放置问题, 以提高服务请求的接受率。策略梯度适用于连续变量求解, 而对于离散空间在拟合误差传递过程中可能出现训练稳定性不足的问题。

上述工作都在努力解决服务链部署问题, 但由于服务请求的随机到达, 网络资源状态通常表现出很大的变化, 同时网络环境的复杂性使得传统算法求解效率明显降低, 因此需要一个合适的模型来捕捉动态网络的状态变化和一种高效的求解策略。本文提出了一种基于深度Q网络的在线服务功能链部署方法。首先, 我们引入马尔可夫决策过程来描述动态的网络资源状态转换过程; 然后, 提出了一种基于DQN的在线服务链部署算法来智能高效地解决服务链部署的复杂性问题, 该算法旨在满足服务请求的时延约束的同时最小化总资源占用开销。仿真结果表明, 本文方法能够克服数学规划的扩展性问题, 而且满足时延要求的条件下最小化服务链部署开销, 可有效提高运营商网络的需求接受率和运营收益。

2 系统模型和问题规划

本节详细描述了服务链部署问题的数学规划, 然后说明了如何使用马尔可夫决策过程建模网络资源状态转变和服务链部署问题, 最后简要介绍了深度Q网络。

2.1 系统模型和问题描述

在一个公共NFV/SDN使能网络环境中, 如图1所示, 首先将底层基础设施网络表示为连接图 $G = (N, L)$, 其中参数 N 表示服务器节点集合, L 表示每两个服务器节点之间连接链路的集合, $\forall mn \in L$ 表示连接服务器节点 $m, n \in N$ 的物理链路。服务器有一定的资源容量, 包含计算资源(CPU)、内存和存储资源(RAM和硬盘)等。 $C_n = (C_n^{\text{cpu}}, C_n^{\text{mem}})^{\text{mem}}$ 表

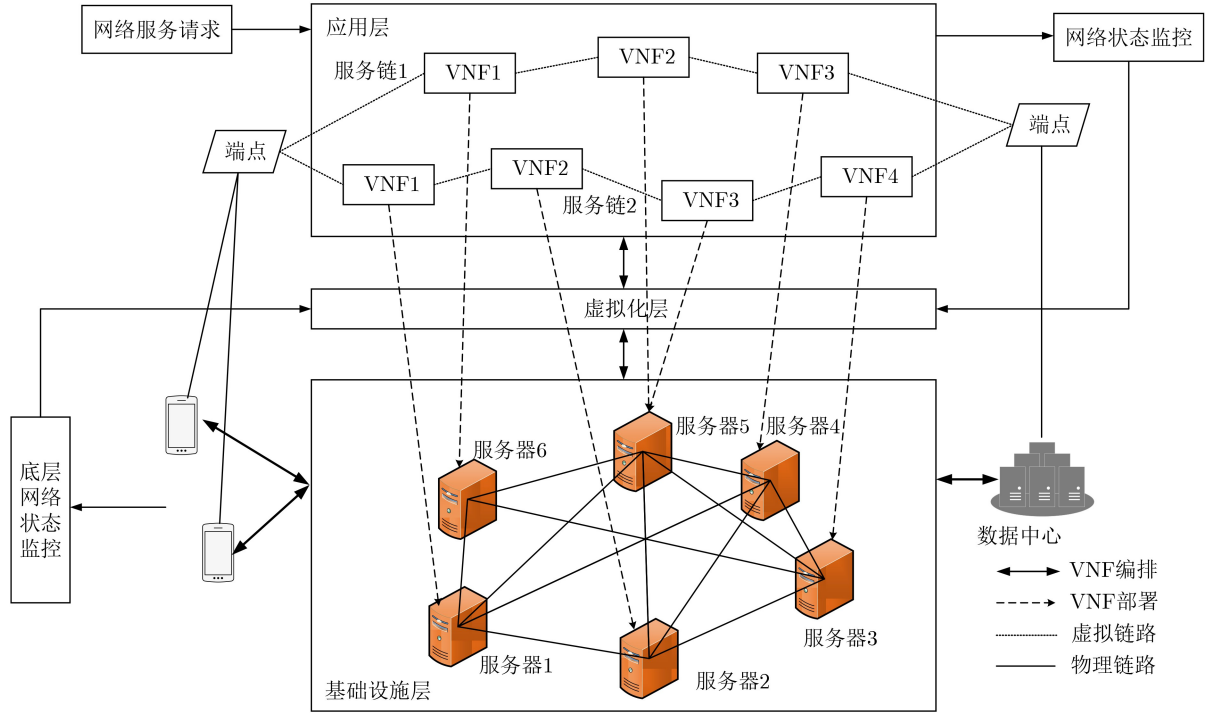


图1 服务功能链部署示意图

示每个服务器的资源容量，分别表示可利用的CPU计算资源和内存资源。由于其他类型的资源(例如硬盘)在服务器中是相当充足的，如果需要也可以添加到 C_n 中。 B_l 表示物理路径 $l \in L$ 可用带宽，同时 T_l 表示物理路径 $l \in L$ 的传输时延，一条底层路径可能包含一条或多条物理链路。

使用 $F = \{f_1, f_2, \dots, f_{|F|}\}$ 表示服务请求的VNFs，包含常用的防火墙、NAT、深度包检测、负载均衡和定制化VNFs(例如视频处理功能)等，其中 $|F|$ 表示VNFs集合的数量，通过不同的下标分别区分不同的VNF类型。每个VNF $f \in F$ 的实例在CPU和内存方面有一定的资源需求，表示为 $D_f = (D_f^{\text{cpu}}, D_f^{\text{mem}})$ 。然后，使用 R 表示随机到达的服务请求，每个服务请求 $r \in R$ 需要经过一组有序的VNFs，即服务链，如图1应用层所示。服务链 $r \in R$ 可描述为有向图 $G_r = (N'_r, L'_r)$ ， N'_r 和 L'_r 分别表示VNFs集合和有向虚拟链路(Virtual Links, VLs) $l' \in L'$ 集合， $|r|$ 表示服务链 $r \in R$ 包含VNFs的数量，因此服务链 $r \in R$ 的相关VNFs集合可表示为： $N_r = [f_1, f_2, \dots, f_{|r|}]$ ， $f_i \in F, i = 1, 2, \dots, |r|$ 。

每条服务链 $r \in R$ 具有不同的流量特征，包括带宽需求 B_r 和响应时延限制 T_r 。对于服务请求 $r \in R$ ，使用二值变量 $x_n^{r,i}$ 表示服务链 $r \in R$ 中每个VNF f_i 的部署决策，如果VNF f_i 成功部署在服务器 $n \in N$ 上， $x_n^{r,i} = 1$ ；否则 $x_n^{r,i} = 0$ 。同样，二值变量 $y_l^{r,l'} = 1$ 表示服务链中VL l' 成功部署在底层路径 $l \in L$ 上。

首先，服务器有充足的资源可以放置多个VNFs，因此表述服务器的资源约束为

$$\sum_{f_i \in r} x_n^{r,i} D_{f_i} \leq C_n, \forall f, n \quad (1)$$

同时每个VNFs只能部署在一个服务器节点 n 上，因此

$$\sum_{f_i} x_n^{r,i} \leq 1, \forall f \quad (2)$$

然后，本文不仅考虑带宽资源约束，也考虑QoS中的时延要求，因为时延约束同样影响服务请求部署是否成功。采用不可分割的多商品流问题建模VLs的底层网络路径，一条VL部署成功当且仅当其连接的VNFs部署成功，同时满足它的时延要求。对于带宽需求

$$\sum_{l'} y_l^{r,l'} B_r \leq B_l, \forall l' \quad (3)$$

服务器节点 n 的流入链路和流出链路集合分别表示为 $I(n)$ 和 $O(n)$ ，以下约束式(4)保证连接VNFs f_i 和 f_j 的VL l' 在底层路径的连续性，而当VL连接的VNFs f_i 和 f_j 映射在同一服务器时 $x_n^{r,i} - x_n^{r,j} = 0$ ，需要 $\sum y_{l_a}^{l'} = \sum y_{l_b}^{l'}$ ，而 $\sum y_{l_a}^{l'} = \sum y_{l_b}^{l'} = 1$ 时会产生回环，因此采用约束(5)避免存在这种情况

$$\sum_{l_a \in O(n)} y_{l_a}^{l'} - \sum_{l_b \in I(n)} y_{l_b}^{l'} = x_n^{r,i} - x_n^{r,j}, \forall n \in N, \forall l' \in L' \quad (4)$$

$$\sum_{l_a \in O(n)} y_{l_a}' + \sum_{l_b \in I(n)} y_{l_b}' \leq 1, \forall l' \in L', \forall n \in N \quad (5)$$

本文使用 t_r 表示服务链 $r \in R$ 的总响应时延，其是链路传输时延和VNF实例处理时延的总和。 t_l 是底层路径的传输时延，由网络实时测量所得； t_f 表示VNF $f \in F$ 实例的处理时延，服务链 $r \in R$ 的端到端时延约束表示为

$$t_r = \sum_{l' \in L'} y_{l'}^{r,l'} t_l + \sum_{f_i \in r} x_n^{r,i} t_f \leq T_r \quad (6)$$

VNF实例的处理时延定义为VNF实例负载占用服务器总负载的比值与服务器处理时延的乘积，如式(7)所示，其中 load_f 表示VNF $f \in F$ 实例占用负载， $\text{load}_{\text{total}}$ 和 T_{total} 分别表示服务器的总负载和总时延。

$$t_f = \frac{\text{load}_f}{\text{load}_{\text{total}}} \times T_{\text{total}} \quad (7)$$

优化目标是 minimized 服务请求占用的资源开销，包括服务器节点资源和链路带宽资源两个部分，如式(8)所示，其中 ξ_D 和 ξ_B 表示加权参数，用来均衡两部分占比，权重系数 ξ_D, ξ_B 满足 $\xi_D, \xi_B \in (0, 1)$ 且 $\xi_D + \xi_B = 1$ ，具体取值见第4节仿真结果与分析。

$$\min \xi_D \sum_n x_n^{r,i} D_{f_i} + \xi_B \sum_l y_l^{r,l'} B_r \quad (8)$$

2.2 马尔可夫决策过程

在通信网络环境中，服务请求的随机到达和离开服从泊松过程，泊松过程满足两个条件：(1)不同服务请求到达或离开是相互独立的事件；(2)在足够小的单位时间内，有且仅有不超过一个服务请求到达或者离开^[19]。定义 $R_r \subset R$ 表示 $[0, \tau]$ 内到达和离开的服务请求，当每个服务请求到达时，NFV/SDN使能网络的管理编排器将会获取底层网络状态信息，包含服务器节点剩余资源、物理链路可用带宽和传输时延等，为服务请求的服务链部署做好准备。马尔可夫决策过程(Markov Decision Process, MDP)是序贯决策的数学模型，用于在系统状态具有马尔可夫性质的环境中模拟智能体可实现的随机性策略与回报^[20]。考虑到服务请求到达和离开以及服务链中VNF的序惯性，本文采用MDP描述服务链部署过程。MDP通常定义为5元组 $\langle S, A, P, R, \gamma \rangle$ ，其中 S 表示状态集合， A 表示离散动作集合， $P: S \times A \times S$ 表示状态转移概率分布， $R: S \times A$ 是奖励函数， $\gamma \in [0, 1]$ 是对未来奖励的折扣因子。

状态定义：对于每一个状态 $s_t \in S$ ，本文将其表示为一个矢量 $\mathbf{s}_t = [C_t, B_t, t_r]$ 。 $C_t = [C_t^1, C_t^2, \dots, C_t^{|N|}]$

表示服务器节点剩余可用资源， $B_t = [B_t^1, B_t^2, \dots, B_t^{|L|}]$ 表示物理链路剩余可用带宽资源， t_r 表示服务链从源节点到当前状态的总时延，包含已经部署在服务器中VNF实例处理时延和物理链路的传输时延，参考服务请求的时延约束作为后序部署的指导。

动作定义：首先为网络中每个服务器节点指定整数索引 $k = 1, 2, \dots, |N|$ ，然后定义动作 $a \in A$ 为整数，其中 $A = \{0, 1, 2, \dots, |N|\}$ 。 $a = 0$ 表示VNF f_i 部署失败的情况；否则， a 表示VNF f_i 部署成功时服务器节点的具体索引。

奖励函数：我们的目标函数最小化服务请求占用底层资源开销，因此定义奖励函数为服务链部署总占用开销的负值，如式(9)所示

$$\text{reward} = -\xi_D \sum_n x_n^{r,i} D_{f_i} - \xi_B \sum_l y_l^{r,l'} B_r \quad (9)$$

状态转移：MDP的转态转移定义为 (s_t, a_t, r_t, s_{t+1}) ，其中 s_t 表示当前网络状态， a_t 是当前网络状态下选择的最优化动作， r_t 和 s_{t+1} 分别表示当前网络状态下执行动作 a_t 获得的奖励和新到达的网络状态。MDP的状态转移会在第3节中具体介绍。

2.3 深度Q网络

传统的Q-learning用表格的方式来记录状态和动作对应的Q值的方法在处理一些大规模的问题上会占用极大的内存，而且重复地搜索大规模表格也是一件很耗时的事情。深度Q网络(Deep Q Network, DQN)是一种将神经网络和Q-learning结合的方法，直接将状态作为神经网络的输入，用神经网络计算出所有的动作价值，并从中选出一个最大值作为输出，或者将状态和动作都作为神经网络的输入，直接输出对应的Q值。

DQN采用行为和观察值的序列作为学习的样本，由于这样的序列彼此之间是完全不同的，所以用这样的序列作为RL中状态时，所有的状态都是完全不同的值，可以将问题转化为MDP，也就方便使用RL来解决问题。同时DQN拥有一个经验复用池来学习之前的学习经历，其中存储的“学习经历”就是之前提到的行为和观察值序列，便于在DQN每次更新时抽取之前的学习经历进行学习。随机抽取的方式打乱了学习经历之间的相关性，也使得神经网络的更新更有效率。

DQN的目标是最大化未来的奖励，假设未来奖励每一时间的步长折扣因子为 γ ，总时长为 T ，则在时间 t 时的未来折扣奖励为 $R_t = \sum_{t'}^T \gamma^{t'-t} r_{t'}$ ，定义动作价值函数 $Q^*(s, a)$ 为所有动作价值中的最大值，则 $Q^*(s, a) = \max_{\pi} E[R_t | s_t = s, a_t = a, \pi]$ ，其中 π 是将状态序列映射到动作的策略， E 是数学期

望。如果对于下一个状态 s' 所有可能存在的动作 a' ，对应的最大值 $Q^*(s', a')$ 都已知，那么当前最优策略就是选择一个使目标函数 $r + \gamma Q^*(s', a')$ 最大的动作 a' ，即Bellman方程式(10)。通过Bellman方程作为迭代更新来估计动作值函数，并收敛于最优动作值函数。

$$Q^*(s, a) = E[r + \gamma \max_{a'} Q^*(s', a') | s, a] \quad (10)$$

在DQN中使用函数逼近器来估计动作值函数 $Q(s, a; \theta) \approx Q^*(s, a)$ ，将具有权重 θ 的神经网络函数逼近器称为Q网络，具体Q网络的训练我们将会在3.3节详细介绍。

3 基于DQN的服务链部署算法

本节首先引进基于DQN的服务链部署架构，然后介绍基于DQN的在线服务链部署算法，最后详细说明深度Q网络的训练过程。

3.1 网络服务部署架构

马尔可夫决策过程能够连续自动地描述网络环境的变化和网络资源状态转移。基于上述条件，我们需要找到一种合适、高效的服务链部署算法，其能够在每个状态下自动采取合适的动作以获得较好的收益。因此，我们推出一种基于DQN的在线服务链部署算法，在满足时延要求的条件下最小化服务请求占用资源开销。

网络服务部署架构如图2所示，基础设施网络环境是NFV/SDN使能网络，包含服务器、SDN交换机和物理链路。基于DQN的管理编排智能体能够从底层网络环境中获取状态信息，并自动选择一个动作作为返回值；执行动作后，NFV/SDN使能网络给予一个奖励反馈；最后，智能体根据奖励反馈更新相关策略，并转移到下一个状态；重复上述过程直至奖励收敛。特别是，基于多层全连接深度神经网络的DQN能高效处理复杂的网络状态输入，并通过反向传播训练不断优化输出动作选择，生成最优化的服务链部署策略 $\pi_Q^* = \pi | S \times A$ 。

3.2 基于DQN的服务链部署算法

为有效应对网络的动态变化，本文采用泊松过

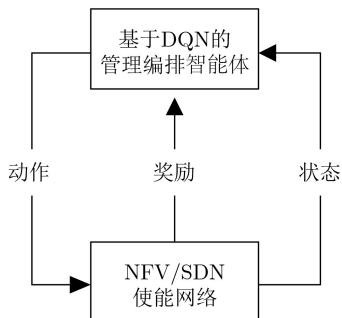


图2 网络服务部署架构

程表示服务请求的到达和离开。当服务请求到达时，NFV/SDN网络管理编排器(MANO)决策是否接受服务请求，然后更新网络状态。MANO按照到达时间依次部署服务请求的服务链，如果服务链部署失败则拒绝服务请求，并返回初始网络状态。服务链部署失败的原因可能包含：(1)服务器资源短缺造成VNFs部署失败；(2)无法满足服务请求的带宽或时延约束。为减小服务链部署过程中VNF放置动作的取值空间，本文采用序列化的方式进行服务链部署，即在每个MDP状态转移内仅部署一个VNF。基于DQN的在线服务链部署算法(Deep Q network based Service Chain Deployment, DeepSCD)如表1所示。首先初始化网络状态，根据输入服务链的长度确定部署步长，检查底层网络资源状态，生成可用服务器节点集合作为动作空间，根据设置条件选择一个动作 a_t 作为返回值；NFV/SDN使能网络给予当前网络状态下执行的动作 a_t 的奖励反馈，智能体根据奖励更新策略并转移至下一个状态。如果部署过程中出现可用服务器节点集合为空，MANO返回信息通知资源无法满足条件，拒绝服务请求，并返回初始化状态。其中经验复用池用于神经网络的训练，本文设置复用池大小

表1 算法1 基于DQN的在线服务链部署算法

输入：服务链信息 $G_r = (N_r', L_r')$ 和当前网络状态 s_t

输出：服务链部署策略 π_Q^*

- (1) 初始化经验复用池 D 的容量 M
- (2) 初始化动作对应的 Q 值为随机值
- (3) 初始化选择策略 π
- (4) for episode in range(EPISODES):
- (5) 初始化状态 s
- (6) for step in range(STEPS):
- (7) 检查底层网络资源状态，生成满足条件服务器节点集合 Φ
- (8) 以 ε 的概率随机选择一个动作 a_t
- (9) 否则选择 $a_t = \max_a Q^*(s_t, a; \theta)$
- (10) 在服务链部署模拟器中执行动作 a_t 并观察对应的奖励 r_t 和下一个状态 s_{t+1}
- (11) 在经验复用池中存储样本 $e_t = (s_t, a_t, r_t, s_{t+1})$
- (12) 从经验复用池中随机抽取小批量的样本 (s_j, a_j, r_j, s_{j+1})
- (13) 令 $y_j = \begin{cases} r_j, & r_j = \text{end} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta), & r_j \neq \text{end} \end{cases}$
- (14) 在 $(y_j - Q(s_{j+1}, a; \theta))^2$ 上执行梯度下降
- (15) 每 L 步更新目标网络参数 θ^-
- (16) end
- (17) end
- (18) 根据 $Q^*(s_t, a; \theta)$ 计算服务链部署方案 π^* 及其开销 R

$M = 500$ ，当经验池存储样本收集满时开始进行训练((11)~(15)行)。如果网络资源满足整个服务链的部署要求，同时检查输入状态中的服务链部署策略的端到端时延值。如果该值满足服务请求的时延约束则输出该策略提供服务；否则MANO返回无法满足QoS条件消息拒绝服务请求。

3.3 训练过程

DQN采用两个神经网络，分别是在线网络和目标网络，其中在线网络不停地更新参数，用来进行神经网络的训练，计算出 Q 估计值；而目标网络则冻结参数，隔一段时间更新一次，用来计算 Q 现实值。DQN的训练流程如图3所示，目标网络 $Q(s', a'|\theta^-)$ 与在线网络 $Q(s, a|\theta)$ 结构相同，只是在每 L 步后对目标网络进行参数更新，使得 $\theta^- = \theta$ 。在一段时间内保持 Q 值是稳定不变的，一定程度上降低了当前 Q 值与目标 Q 值的相关性，提升了算法的稳定性。在Q网络训练过程中，一般会通过随机梯度下降来优化损失函数，损失函数为

$$L(\theta) = (r + \gamma \max_a Q(s', a'|\theta^-) - Q(s, a|\theta))^2 \quad (11)$$

4 仿真结果与分析

4.1 实验环境和参数设置

仿真采用BtEurope^[21]网络拓扑，包含24个数据中心节点和37条全双工链路。数据中心可用资源容量在 $[5, 50]$ (unit)区间内随机选取，链路带宽容量随机选取以下数值：100 Mbps, 150 Mbps, 600 Mbps和1 Gbps，传输时延在 $[500, 1000]$ (μ s)内随机生成。一条服务链包含5~9个VNFs，VNF请求的资源数量服从 $[0.5, 2]$ 上的均匀分布，VL请求带宽在1~40 Mbps内随机选取，服务请求最大容忍时延设置在 $[5, 20]$ (ms)。

基于上述设置，本实验在Xeon E5-2630 v4 2.2 GB和64 GB内存的Windows 10系统内使用

Pycharm IDE进行仿真。采用基于Python 3.8的Pytorch 1.6机器学习库执行深度学习，使用NetworkX^[22]仿真数据中心基础设施的底层网络。管理编排智能体的设置使用以下参数，折扣因子 $\gamma = 0.85$ ，神经网络的参数学习采用Adma，目标网络的更新周期 $L = 50$ ，神经网络隐藏层采用2层全连接结构，神经元个数为100，线性整流函数(Rectified Linear Unit, ReLU)作为激活函数。

4.2 性能分析

为验证本文算法DeepSCD的性能和有效性，我们比较DeepSCD与启发式算法First-Fit-Dijkstra(FFD)和贝叶斯方法(Bayes)的性能。FFD算法采用First-Fit算法为VNFs分配底层网络服务器资源和Dijkstra算法定义VLs的底层传输路径，在文献[23–25]中作为评估性能的基准线算法。贝叶斯方法采用贝叶斯学习方法来解决NFV组件的预测、分割和部署问题^[26]。在DeepSCD中，VNFs的放置位置是由神经网络决定的，因此DQN agent生成动作的质量将会对算法性能产生明显的影响。

图4显示了本文所提服务链部署算法在不同学习率下的奖励，学习率(Learning Rate, LR)分别为0.010, 0.025, 0.050和0.100。从图中可以看出，学习率在算法的训练阶段影响奖励的值，因为不同的学习率表示奖励函数收敛的学习步长。在学习过程中，较大的学习率可能错过全局最优解，而较小的学习率可能导致收敛速度较慢。仿真环境中，LR=0.050表现出较好的性能，它不仅能获得近似最优解，而且收敛速度较快。

图5描述了本文算法在记忆回放阶段抽取样本批量大小对奖励函数的影响，批量大小(Batch Size, BS)分别为16, 32, 64和128。在训练过程中，从经验复用池随机抽取小批量样本，并使用随机梯度下降算法更新网络参数。可以看出，不同抽取样本批

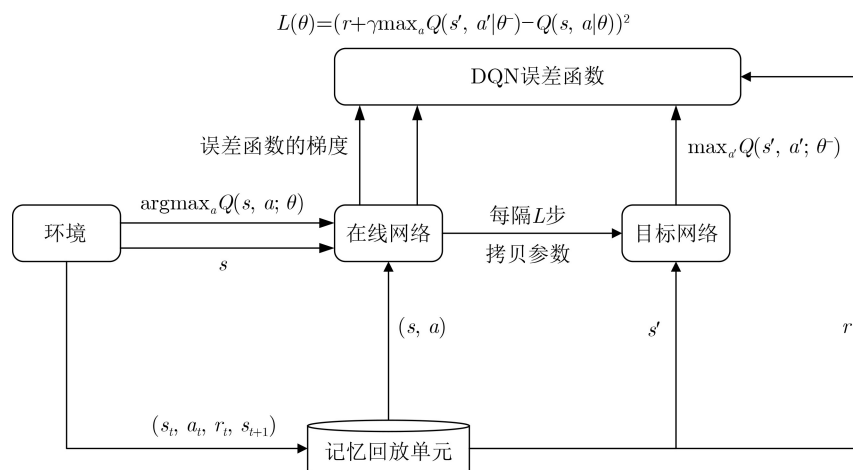


图3 DQN训练流程

量大小影响训练阶段奖励函数的收敛速度。抽取样本批量过小可能带来较大的方差,影响算法收敛速度甚至不收敛。大批量样本的梯度估计更加稳定和准确,其需要较高的计算时间,且可能导致神经网络陷入较差的局部最优解。因此,样本批量设置不能过大或者过小。根据仿真结果,样本批量设置为 $BS=64$ 。

图6表示不同请求到达强度下成功部署的服务链的平均部署开销,部署开销由式(8)定义,对应参数分别取值 $\xi_D = 0.95$ 和 $\xi_B = 0.05$ 。从图中可以看出,本文的DeepSCD算法相较于FFD和Bayes算法的部署开销存在明显的降低。FFD算法采用简单的贪婪式部署,DeepSCD和Bayes算法能够根据网络资源状态变化整体统筹调整SFC部署策略,保证服务请求部署开销总体较低。Bayes学习基于假设模型参数和样本的后验概率估计总体分布,样本较少和网络状态的变化导致其准确性相对降低。而DeepSCD算法采用的马尔可夫决策模型,对网络状态变化的适应性更好,因此在部署策略上更优。

图7表示不同请求强度下服务请求的请求接受率。3种算法均随着请求到达强度的上升而下降,是因为随着服务请求数量的增加网络资源被占用而导致后续服务请求被拒绝。FFD算法不考虑底层服务器节点和链路资源状态对部署策略的影响,部分中心链路和节点被频繁占用以致局部拥塞,故下降速度最快;而DeepSCD和Bayes算法统筹考虑了底

层资源的分配策略,同时DeepSCD算法在服务链部署过程中序列化的方式能更好地捕捉网络资源状态的动态变化,故在请求接受率方面存在优势。

图8表示不同请求强度下部署成功网络服务的平均时延。FFD算法在接受服务请求时采用Dijkstra算法首先占用较低时延链路,使得后续服务链仅能选择迂回链路以致时延迅速上升,而Bayes和DeepSCD算法通过全局规划考虑避免了局部链路过度使用。相较于Bayes算法未考虑链路的时延参数动态变化,DeepSCD算法在部署过程中以序列化的方式实时更新链路时延信息,并根据输入信息更新部署策略,使得网络整体的链路使用更加高效。结果表明,本文算法能有效降低服务链的传输时延。

5 结束语

本文主要研究了NFV/SDN使能网络环境中服

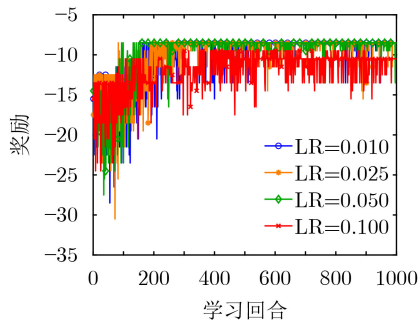


图4 不同学习率对奖励函数的影响

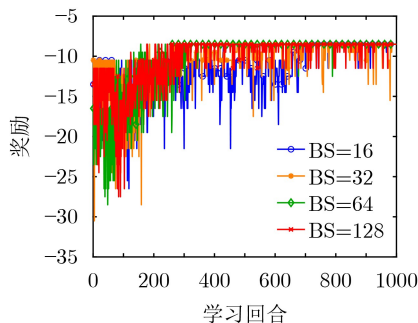


图5 不同抽取样本批量大小对奖励函数的影响

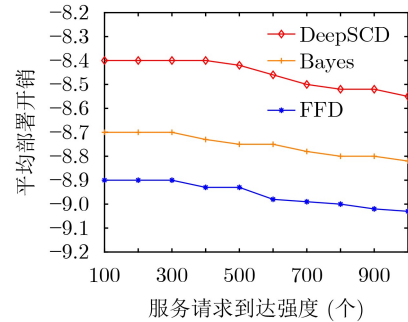


图6 不同请求强度下的平均部署开销

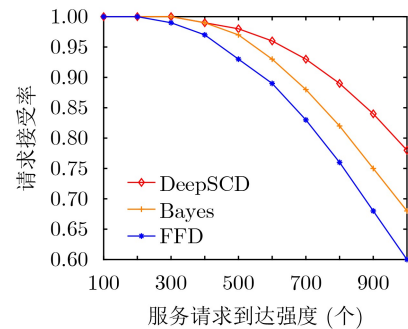


图7 不同请求强度下的请求接受率

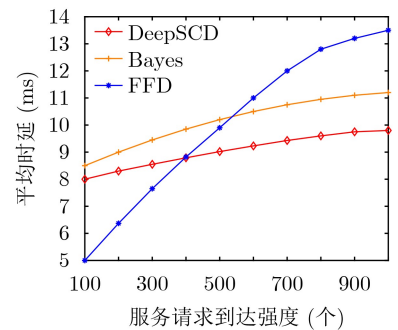


图8 不同请求强度下的平均时延

务链的部署问题, 针对传统服务链部署算法的不足和网络状态动态变化问题, 在保障5G业务时延性约束的条件下, 为降低运营商网络的资源开销, 提出基于DQN的在线服务链部署方法, 并验证了方法的有效性。为了进一步提高云化环境中资源利用的效率, 后续将针对网络中流量动态变化研究虚拟网络功能的自动化扩缩容问题, 以满足下一代移动网络的智能化和自动化特征。

参 考 文 献

- [1] ETSI. Network Functions Virtualisation (NFV)[EB/OL]. <https://www.etsi.org/technologies/nfv>, 2020.
- [2] YI Bo, WANG Xingwei, LI Kebin, *et al.* A comprehensive survey of network function virtualization[J]. *Computer Networks*, 2018, 133: 212–262. doi: [10.1016/j.comnet.2018.01.021](https://doi.org/10.1016/j.comnet.2018.01.021).
- [3] ERAMO V, MIUCCI E, AMMAR M, *et al.* An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures[J]. *IEEE/ACM Transactions on Networking*, 2017, 25(4): 2008–2025. doi: [10.1109/TNET.2017.2668470](https://doi.org/10.1109/TNET.2017.2668470).
- [4] KARAKUS M and DURRESI A. A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)[J]. *Computer Networks*, 2017, 112: 279–293. doi: [10.1016/j.comnet.2016.11.017](https://doi.org/10.1016/j.comnet.2016.11.017).
- [5] BHAMARE D, JAIN R, SAMAKA M, *et al.* A survey on service function chaining[J]. *Journal of Network and Computer Applications*, 2016, 75: 138–155. doi: [10.1016/j.jnca.2016.09.001](https://doi.org/10.1016/j.jnca.2016.09.001).
- [6] MIJUMBI R, SERRAT J, GORRICO J L, *et al.* Network function virtualization: State-of-the-art and research challenges[J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(1): 236–262. doi: [10.1109/COMST.2015.2477041](https://doi.org/10.1109/COMST.2015.2477041).
- [7] BARI M F, CHOWDHURY S R, AHMED R, *et al.* Orchestrating virtualized network functions[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(4): 725–739. doi: [10.1109/TNSM.2016.2569020](https://doi.org/10.1109/TNSM.2016.2569020).
- [8] LIU Jiaqiang, LI Yong, ZHANG Ying, *et al.* Improve service chaining performance with optimized middlebox placement[J]. *IEEE Transactions on Services Computing*, 2017, 10(4): 560–573. doi: [10.1109/TSC.2015.2502252](https://doi.org/10.1109/TSC.2015.2502252).
- [9] KUO T W, LIOU B H, LIN K C J, *et al.* Deploying chains of virtual network functions: On the relation between link and server usage[C]. IEEE INFOCOM 2016 - the 35th Annual IEEE International Conference on Computer Communications, San Francisco, USA, 2016: 1–9. doi: [10.1109/INFOCOM.2016.7524565](https://doi.org/10.1109/INFOCOM.2016.7524565).
- [10] SUN Quanying, LU Ping, LU Wei, *et al.* Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement[C]. 2016 IEEE Global Communications Conference (GLOBECOM), Washington, USA, 2016: 1–6. doi: [10.1109/GLOCOM.2016.7841846](https://doi.org/10.1109/GLOCOM.2016.7841846).
- [11] LI Defang, HONG Peilin, XUE Kaiping, *et al.* Virtual network function placement considering resource optimization and SFC requests in cloud datacenter[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 29(7): 1664–1677. doi: [10.1109/TPDS.2018.2802518](https://doi.org/10.1109/TPDS.2018.2802518).
- [12] HAWILO H, JAMMAL M, and SHAMI A. Network function virtualization-aware orchestrator for service function chaining placement in the cloud[J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(3): 643–655. doi: [10.1109/JSAC.2019.2895226](https://doi.org/10.1109/JSAC.2019.2895226).
- [13] TANG Hong, ZHOU D, and CHEN Duan. Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2019, 30(3): 530–543. doi: [10.1109/TPDS.2018.2867587](https://doi.org/10.1109/TPDS.2018.2867587).
- [14] QI Dandan, SHEN Subin, and WANG Guanghui. Towards an efficient VNF placement in network function virtualization[J]. *Computer Communications*, 2019, 138: 81–89. doi: [10.1016/j.comcom.2019.03.005](https://doi.org/10.1016/j.comcom.2019.03.005).
- [15] SINGH S, OKUN A, and JACKSON A. Learning to play Go from scratch[J]. *Nature*, 2017, 550(7676): 336–337. doi: [10.1038/550336a](https://doi.org/10.1038/550336a).
- [16] 袁泉, 汤红波, 黄开枝, 等. 基于Q-learning算法的vEPC虚拟网络功能部署方法[J]. 通信学报, 2017, 38(8): 172–182. doi: [10.11959/j.issn.1000-436x.2017173](https://doi.org/10.11959/j.issn.1000-436x.2017173).
YUAN Quan, TANG Hongbo, HUANG Kaizhi, *et al.* Deployment method for vEPC virtualized network function via Q-learning[J]. *Journal on Communications*, 2017, 38(8): 172–182. doi: [10.11959/j.issn.1000-436x.2017173](https://doi.org/10.11959/j.issn.1000-436x.2017173).
- [17] XIAO Yikai, ZHANG Qixia, LIU Fangming, *et al.* NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning[C]. Proceedings of the International Symposium on Quality of Service, Arizona, USA, 2019: 1–10. doi: [10.1145/3326285.3329056](https://doi.org/10.1145/3326285.3329056).
- [18] QUANG P T A, HADJADJ-AOUL Y, and OUTTAGARTS A. A deep reinforcement learning approach for VNF forwarding graph embedding[J]. *IEEE Transactions on Network and Service Management*, 2019, 16(4): 1318–1331. doi: [10.1109/TNSM.2019.2947905](https://doi.org/10.1109/TNSM.2019.2947905).
- [19] KINGMAN J F C. Poisson Processes[M]. ARMITAGE P and COLTON T. Encyclopedia of Biostatistics. Chichester: John Wiley & Sons, 2005. doi: [10.1002/0470011815.b2a07042](https://doi.org/10.1002/0470011815.b2a07042).
- [20] HOWARD R A. Dynamic programming and Markov processes[J]. *Technometrics*, 1961, 3(1): 120–121. doi: [10.2307/1266484](https://doi.org/10.2307/1266484).

- [21] The University of Adelaide. The internet topology zoo[EB/OL]. <http://www.topology-zoo.org/dataset.html>, 2012.
- [22] Networkx. Network analysis in python: Important structures and bipartite graphs[EB/OL]. <https://codertzcolumn.com/tutorials/data-science/network-analysis-in-python-important-structures-and-bipartite-graphs-networkx>, 2020.
- [23] YALA L, FRANGOUDIS P A, LUCARELLI G, *et al.* Cost and availability aware resource allocation and virtual function placement for CDNaaS provision[J]. *IEEE Transactions on Network and Service Management*, 2018, 15(4): 1334–1348. doi: [10.1109/TNSM.2018.2874524](https://doi.org/10.1109/TNSM.2018.2874524).
- [24] OCHOA-ADAY L, CERVELLÓ-PASTOR C, FERNÁNDEZ-FERNÁNDEZ A, *et al.* An online algorithm for dynamic NFV placement in cloud-based autonomous response networks[J]. *Symmetry*, 2018, 10(5): 163. doi: [10.3390/sym10050163](https://doi.org/10.3390/sym10050163).
- [25] MAROTTA A, ZOLA E, D'ANDREAGIOVANNI F, *et al.* A fast robust optimization-based heuristic for the deployment of green virtual network functions[J]. *Journal of Network and Computer Applications*, 2017, 95: 42–53. doi: [10.1016/j.jnca.2017.07.014](https://doi.org/10.1016/j.jnca.2017.07.014).
- [26] SHI Runyu, ZHANG Jia, CHU Wenjing, *et al.* MDP and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization[C]. 2015 IEEE International Conference on Services Computing, New York, USA, 2015: 65–73. doi: [10.1109/SCC.2015.19](https://doi.org/10.1109/SCC.2015.19).
- 邱 航: 男, 1994年生, 博士生, 主要研究方向为新一代移动通信网络技术、移动通信网络安全.
- 汤红波: 男, 1968年生, 教授、博士生导师, 主要研究方向为移动通信网络、新型网络体系结构.
- 游 伟: 男, 1984年生, 博士, 讲师, 主要研究方向为移动通信网络安全、新一代移动通信网络技术.

责任编辑: 陈 倩