基于改进深度强化学习的虚拟网络功能部署优化算法

唐 伦[①②] 贺兰钦*[①②] 连沁怡[③] 谭 颀[①②]

[①](重庆邮电大学通信与信息工程学院 重庆 400065)

[②](重庆邮电大学移动通信技术重点实验室 重庆 400065)

[③](三峡大学国际交流学院 宜昌 443002)

摘 要：针对网络功能虚拟化/软件定义网络 (NFV/SDN)架构下，网络服务请求动态到达引起的服务功能链(SFC)部署优化问题，该文提出一种基于改进深度强化学习的虚拟网络功能(VNF)部署优化算法。首先，建立了马尔科夫决策过程 (MDP)的随机优化模型，完成SFC的在线部署以及资源的动态分配，该模型联合优化SFC部署成本和时延成本，同时受限于SFC的时延以及物理资源约束。其次，在VNF部署和资源分配的过程中，存在状态和动作空间过大，以及状态转移概率未知等问题，该文提出了一种基于深度强化学习的VNF智能部署算法，从而得到近似最优的VNF部署策略和资源分配策略。最后，针对深度强化学习代理通过ε贪婪策略进行动作探索和利用，造成算法收敛速度慢等问题，提出了一种基于值函数差异的动作探索和利用方法，并进一步采用双重经验回放池，解决经验样本利用率低的问题。仿真结果表示，该算法能够加快神经网络收敛速度，并且可以同时优化SFC部署成本和SFC端到端时延。

关键词：虚拟网络功能；深度强化学习；服务功能链端到端时延；服务功能链部署成本

中图分类号：TN929.5 **文献标识码**：A **文章编号**：1009-5896(2021)06-1724-09

DOI: 10.11999/JEIT200297

Virtual Network Function Placement Optimization Algorithm Based on Improve Deep Reinforcement Learning

TANG Lun[①②] HE Lanqin[①②] LIAN Qinyi[③] TAN Qi[①②]

[①](*School of Communication and Information Engineering*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

[②](*Key Laboratory of Mobile Communications Technology*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

[③](*College of International Communications*, *China Three Gorges University*, *Yichang* 443002, *China*)

Abstract: Considering the problem of Service Function Chain (SFC) placement optimization caused by the dynamic arrival of network service requests under the Network Function Virtualization/Software Defined Network (NFV/SDN) architecture, a Virtual Network Function (VNF) placement optimization algorithm based on improved deep reinforcement learning is proposed. Firstly, a stochastic optimization model of Markov Decision Process (MDP) is established to jointly optimizes SFC placement cost and delay cost, and is constrained by the delay of SFC, as well as the resources of common server Central Processing Unit (CPU) and physical link bandwidth. Secondly, in the process of VNF placement and resource allocation, there are problems such as too large state space, high dimension of action space, and unknown state transition probability. A VNF intelligent placement algorithm based on deep reinforcement learning is proposed to obtain an approximately optimal VNF placement strategy and resource allocation strategy. Finally, considering the problems of deep

收稿日期：2020-04-21；改回日期：2021-01-22；网络出版：2021-01-29

*通信作者： 贺兰钦 719097886@qq.com

基金项目：国家自然科学基金(62071078)，重庆市教委科学技术研究项目(KJZD-M201800601)，重庆市重大主题专项 (cstc2019jscx-zdztzxX0006)

Foundation Items: The National Natural Science Foundation of China (62071078), The Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-M201800601), The Major Theme Special Projects of Chongqing (cstc2019jscx-zdztzxX0006)

reinforcement learning agent's action exploration and utilization through ε greedy strategy, resulting in low learning efficiency and slow convergence speed, a method of action exploration and utilization based on the difference of value function is proposed, and further adopts dual experience playback pool to solve the problem of low utilization of empirical samples. Simulation results show that the algorithm can converge quickly, and it can optimize SFC placement cost and SFC end-to-end delay.

Key words: Virtual Network Function(VNF); Deep reinforcement learning; Service Function Chain (SFC) end-to-end delay; Service Function Chain (SFC) placement cost

1　引言

近年来，网络功能虚拟化技术作为网络服务提供的一个重要范式转变，受到了业界和学术界广泛的关注，在网络功能虚拟化 (Network Function Virtualization, NFV)架构下，一系列虚拟网络功能 (Virtual Network Function, VNF)按照特定的顺序构成的服务功能链(Service Function Chain, SFC)为用户提供服务[1]，相同类型的VNF可以部署或者重新实例化在不同通用服务器上，不需要重新购买硬件。

在网络功能虚拟化/软件定义网络 (Network Function Virtualization/Software Defined Network, NFV/SDN)架构下，VNF部署需要解决问题的关键是如何在有限的物理资源中选择满足服务需求的物理通用服务器和物理链路进行部署，在保证网络性能的同时，最大化资源利用率[2]。

目前，已有大量工作对VNF部署机制展开了研究，其中，文献[3]在保证用户服务质量(Quality of Service, QoS)的同时，减少运营商的成本，但是它采用的是静态VNF部署策略，由于网络环境是动态变化的，需要考虑长时间的优化，文献[4]在SFC部署的时候，以最小化SFC端到端时延为目标，通过减少SFC传输时延和处理时延达到减少SFC端到端时延的目的，但是没有关注到物理网络资源利用率的情况，在文献[5]中，在考虑服务器资源容量和流速率的同时，平衡VNF的操作成本、维护VNF实例成本以及VNF部署成本，但是没有考虑SFC时延，进而忽略了用户的QoS。

综上所述，在目前研究VNF部署的相关文献中，大多数文献研究都是基于环境状态已知的情况下，没有考虑到环境随时间的动态变化，而且没有考虑到大量的网络业务请求达到，引起业务请求积压，进而影响网络稳定性问题，而且在考虑SFC部署成本的同时，没有保证用户的QoS。本文针对网络服务请求动态到达引起的SFC部署优化问题，提出了一种基于改进深度强化学习的虚拟网络功能部署算法，本文主要贡献有：(1) 将随机优化问题建立为马尔科夫决策过程(Markov Decision Process, MDP)模型，该模型联合优化SFC端到端时延和SFC部署成本，同时受限于每条SFC的端到端时延以及通用服务器CPU资源、物理链路带宽资源约束；(2) 在VNF部署和资源分配的过程中，存在状态空间过大、动作空间维度高、状态转移概率未知等问题，提出了一种基于深度强化学习的VNF智能部署算法，本算法通过神经网络近似最优动作值函数，从而得到近似最优的VNF部署策略和资源分配策略；(3) 针对深度强化学习代理通过ε贪婪算法进行动作探索和利用，造成神经网络学习效率低、收敛速度慢等问题，提出了一种基于值函数差异的动作探索和利用方法，并进一步采用双重经验回放池，解决经验样本利用率低的问题，加速训练神经网络。

2　系统模型

2.1　网络场景

本文采用NFV编排和控制架构[6]，如图1所示，主要分为3层。

2.2　网络模型

2.2.1　物理网络

将物理网络抽象为一个无向图$G^{\mathrm{P}}=(V^{\mathrm{P}},E^{\mathrm{P}})$，$V^{\mathrm{P}}$表示物理节点，即通用物理服务器，为VNF提供其实例化的CPU资源，且每台底层通用物理服务器可以实例化多个VNF，E^{P}表示物理链路集合。每台底层通用服务器$v\in V^{\mathrm{P}}$的CPU容量为C_v^{P}，连接相邻通用服务器v和u的物理链路uv的带宽容量为B_{uv}^{P}。值得注意的是，由于某些通用服务器CPU资源利用率低，故本文为通用服务器设置一个CPU资源阈值ℓ_v^{P}，即每个时隙通用服务器的CPU剩余资源要小于ℓ_v^{P}，否则通用服务器不会被使用，以此保证通用服务器的资源利用率，并达到节能的目的。

2.2.2　SFC请求

网络中SFC的集合为F，将第i条SFC形式化为有向图$G_i^{\mathrm{V}}=(N_i^{\mathrm{V}},L_i^{\mathrm{V}})$，$N_i^{\mathrm{V}}$表示第$i$条SFC上的不同类型VNF的集合，$L_i^{\mathrm{V}}$表示$i$条SFC上的虚拟链路集合，第$i$条SFC上的第$j$个VNF表示为$N_{i,j}^{\mathrm{V}}$，且通用物理服务器分配给第$i$条SFC上的第$j$个VNF的

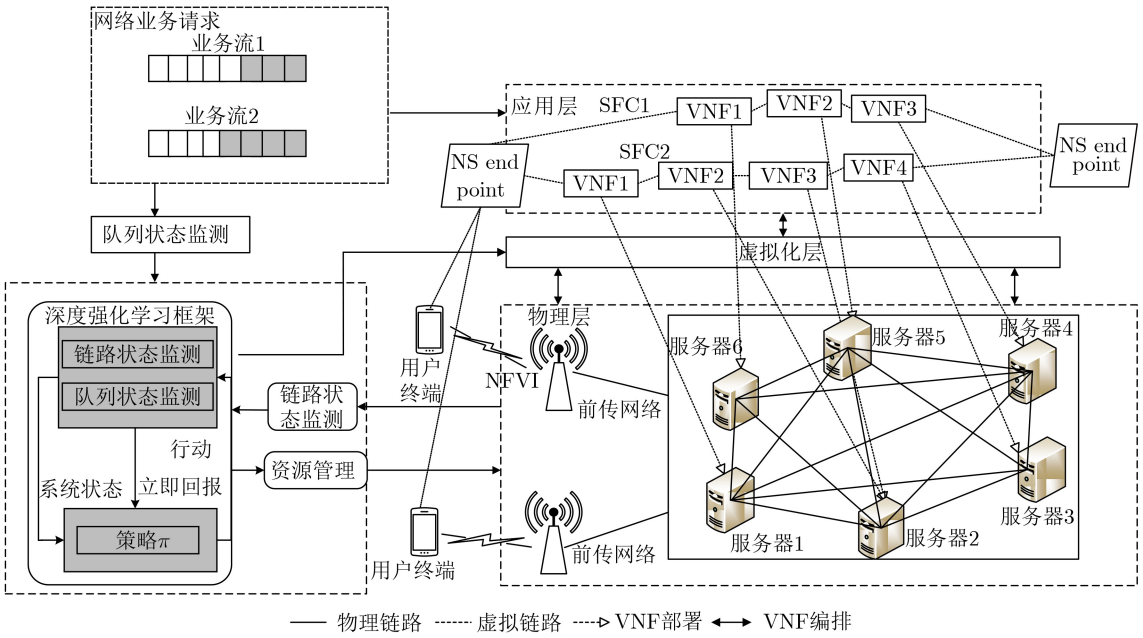

图 1 系统模型

CPU资源表示为$C_{i,j}^{v}$，物理链路分配给连接相邻VNFjk的虚拟链路带宽资源为$B_{i,jk}^{uv}$，第i条SFC的最长时延限制为D_i。接下来定义一个布尔变量$\theta_{i,j}^{v}$，当第i条SFC上的第j个VNF映射到服务器v时，$\theta_{i,j}^{v}=1$，否则$\theta_{i,j}^{v}=0$。

另外，通用服务器v在t时隙的剩余CPU容量$\kappa_v(t)$可以表示为

$$\kappa_v(t)=C_v^{\mathrm{P}}-\sum_{i\in F}\sum_{j\in N_{i,j}^{\mathrm{V}}}\theta_{i,j}^{v}(t)C_{i,j}^{v}(t),\forall v\in V^{\mathrm{P}} \tag{1}$$

在t时隙的剩余带宽资源量$\kappa_{vu}(t)$如式(2)所示

$$\kappa_{vu}(t)=B_{uv}^{\mathrm{P}}-\sum_{i\in F}\sum_{j,k\in N_{i,j}^{\mathrm{V}}}\theta_{i,j}^{v}(t)\theta_{i,k}^{u}(t)B_{i,jk}^{uv}(t),\forall v,u\in V^{\mathrm{P}} \tag{2}$$

2.2.3 SFC时延模型

在计算SFC时延的时候，本文主要考虑排队时延、处理时延以及链路传输时延，传播时延的值可以忽略不计，对第i条SFC，令$Q_i(t)$表示其在t时隙的队列长度，$A_i(t)$表示第i条SFC的数据包达到过程，本文在这里假设数据包的到达$A_i(t)$服从参数为λ_i的泊松分布，数据包大小$P_i(t)$服从参数为$\overline{P}$的指数分布[7]，则队列的更新过程可以表示为

$$Q_i(t+1)=\max[Q_i(t)-\mu_{i,1}(t)+A_i(t),0] \tag{3}$$

其中，$\mu_{i,1}(t)$表示时隙第i条SFC的第1个VNF服务速率，且第i条SFC的第j个VNF的服务速率$\mu_{i,j}(t)$由物理通用服务器分配给其的CPU资源量决定，即$\mu_{i,j}(t)=C_{i,j}^{v}\cdot\beta$，其中$\beta$为服务速率系数，表示CPU资源与服务速率之间的比例[8]。则第i条SFC的时延表示为

$$T_i(t)=\frac{Q_i(t)}{\lambda_i(t)}+\sum_{j,k\in N_i^{\mathrm{V}}}\left(\frac{R_{i,j}(t)}{\mu_{i,j}(t)}+\frac{R_{i,k}(t)}{B_{i,jk}^{uv}(t)}\right) \tag{4}$$

其中，$R_{i,j}(t)$表示t时隙在VNF $N_{i,j}^{\mathrm{V}}$处的数据包到达量，$R_{i,k}(t)$表示从VNF $N_{i,j}^{\mathrm{V}}$到VNF $N_{i,k}^{\mathrm{V}}$的数据量，即t时隙在VNF $N_{i,k}^{\mathrm{V}}$处的数据包达到量。

2.2.4 SFC部署成本模型

SFC部署成本主要分为两个方面，一方面为VNF部署完成之后，VNF占用CPU资源的成本$S_{i,j}^{v}$，另一方面为虚拟链路部署完成之后，占用物理链路带宽资源成本$S_{i,jk}^{vu}$，参考文献[9]，在t时隙$S_{i,j}^{v}(t)$可以表示为与通用服务器v剩余CPU资源$\kappa_v(t)$成反比，即$S_{i,j}^{v}(t)=\partial/\kappa_v(t)$，其中，$\partial$表示正数，$S_{i,jk}^{vu}(t)$可以表示为与物理链路$vu$剩余带宽资源$\kappa_{vu}(t)$成反比，即$S_{i,jk}^{vu}(t)=\varsigma/\kappa_{vu}(t)$，其中，$\varsigma$为正数。

则在t时隙第i条SFC的部署成本可以表示为

$$S_i(t)=\sum_{j\in N_i^{\mathrm{V}}}\sum_{v\in V^{\mathrm{P}}}\theta_{i,j}^{v}(t)\cdot\frac{\partial}{\kappa_v(t)}+\sum_{j,k\in N_i^{\mathrm{V}}}\sum_{u,v\in V^{\mathrm{P}}}\theta_{i,j}^{v}(t)\theta_{i,k}^{u}(t)\cdot\frac{\varsigma}{\kappa_{vu}(t)} \tag{5}$$

2.3 优化目标

本文主要考虑的问题是：在保证用户时延的同时最小化SFC的部署成本，并将物理通用服务器的CPU资源、物理链路带宽资源以及SFC端到端时延作为约束条件，设计了一个效用函数$U(t)$，将其定义为

$$U(t)=-\mathrm{e1}\frac{\sum\limits_{i\in F}S_i(t)}{S_{\max}}-\mathrm{e2}\frac{\sum\limits_{i\in F}T_i(t)}{\sum\limits_{i\in F}D_i(t)} \tag{6}$$

其中，e1和e2表示两个权重值，且e1 + e2 = 1，$S_{\max}$表示SFC部署成本的最大值，将SFC部署成本做归一化处理，则两者处在同一量级上，可以无单位相，则优化目标表示为

$$\begin{aligned}&\max_{\theta_{i,j}^v(t),C_{i,j}^v(t),B_{i,jk}^{uv}(t)} U(t)\\ &\text{s.t. C1}:\sum_{v\in V^{\mathrm{P}}}\theta_{i,j}^v(t)=1,\forall v\in V^{\mathrm{P}}\\ &\quad\mathrm{C2}:\sum_{i\in F}\sum_{j\in N_{i,j}^{\mathrm{V}}}\theta_{i,j}^v(t)C_{i,j}^v(t)\le C_v^{\mathrm{P}},\forall v\in V^{\mathrm{P}}\\ &\quad\mathrm{C3}:\kappa_v(t)\le\ell_v^{\mathrm{P}},\forall v\in V^{\mathrm{P}}\\ &\quad\mathrm{C4}:\sum_{i\in F}\sum_{j,k\in N_{i,j}^{\mathrm{V}}}\theta_{i,j}^v(t)\theta_{i,k}^u(t)B_{i,jk}^{uv}(t)\le B_{uv}^{\mathrm{P}},\\ &\qquad\forall v,u\in V^{\mathrm{P}}\\ &\quad\mathrm{C5}:Q_i(t)<Q_i^{\max}(t),\forall i\in F\\ &\quad\mathrm{C6}:T_i(t)<D_i,\forall i\in F\\ &\quad\mathrm{C7}:\theta_{i,j}^v(t)=\{0,1\},\forall i\in F,\forall j\in N_i^{\mathrm{V}},\forall v\in V^{\mathrm{P}}\end{aligned} \tag{7}$$

该优化目标受到C1～C6限制条件约束，保证优化目标的有效性。C1用于保证虚拟网络中每个VNF只能选择一个服务器进行映射。C2确保每台通用服务器分配给映射在其上的VNF CPU资源之和不能超过该通用服务器的CPU容量。C3用于保证通用服务器的资源利用率，即每台通用服务器的剩余CPU资源低于阈值，否则将不会使用，进一步达到节能的效果。C4表示映射到某条物理链路上的所有虚拟链路数据量之和不能超过该物理链路的总带宽资源。C5用于保证每条SFC的队列长度不溢出。C6用于保证每条SFC在任何时隙都要满足时延要求。C7表示VNF映射的二进制变量。

本文将该随机优化问题转化成一个MDP模型，主要包含状态空间、动作空间和回报函数。

设状态空间为X，且定义$x(t)$表示网络在时隙t时的状态，由每条SFC的队列状态，物理链路带宽剩余资源和通用服务器剩余CPU资源构成，其表达式为

$$x(t)=\left(Q_i(t),\kappa_{vu}(t),\kappa_v(t),\quad\forall u,v\in V^{\mathrm{P}},\forall i\in F\right) \tag{8}$$

设动作空间为A，且定义$a(t)$表示网络在时隙$C_{i,j}^v(t)$时采取的动作，主要包括CPU资源分配$C_{i,j}^v(t)$、链路带宽资源分配$B_{i,jk}^{uv}(t)$和VNF部署$\theta_{i,j}^v(t)$，进一步表示为

$$\begin{aligned}a(t)=&(C_{i,j}^v(t),B_{i,jk}^{uv}(t),\theta_{i,j}^v(t)),\forall i\in F,\\ &\forall j\in N_i^{\mathrm{V}},\forall u,v\in V^{\mathrm{P}}\end{aligned} \tag{9}$$

值得注意的是，动作空间需要满足式(7)中的约束C1～C6。

在网络状态$x(t)$采取动作$a(t)$后，网络环境会得到即时奖励$r(x(t),a(t))$，本文的奖励为效用函数，则$r(x(t),a(t))$定义为

$$r(x(t),a(t))=U(t) \tag{10}$$

设网络初始状态$x(0)$的动作策略为π，即$\pi:x\to a$，具体表示为

$$\pi=\{x(0),a(0),\cdots,x(t),a(t)\} \tag{11}$$

通常最优策略π^*是通过动作值函数$Q^\pi(x,a)$得到的，即

$$\pi^*=\arg\max_A Q^\pi(x,a) \tag{12}$$

动作值函数$Q^\pi(x(t),a(t))$是用来评判当前状态为$x(t)$时选择行为$a(t)$的好坏，可以通过贝尔曼方程迭代获得，即

$$\begin{aligned}Q^\pi(x(t),a(t))=&r(x(t),a(t))\\ &+\gamma\sum_{x(t+1)\in X}\Pr(x(t),a(t),x(t+1))\\ &\cdot Q^\pi(x(t+1))\end{aligned} \tag{13}$$

其中，$\gamma\in(0,1)$表示折扣因子。

则最优动作值函数$Q^*(x(t),a(t))$可以表示为

$$\begin{aligned}Q^*(x(t),a(t))=&\max_{a(t)\in A}\{r(x(t),a(t))\\ &+\gamma\sum_{x(t+1)\in X}\Pr(x(t),a(t),x(t+1))\\ &\cdot Q^\pi(x(t+1))\}\end{aligned} \tag{14}$$

最优动作值函数$Q^*(x(t),a(t))$对应着当前状态$x(t)$下采取的最优动作$a^*(t)$，将其表示为

$$\begin{aligned}a^*(t)=&\arg\max_{a(t)\in A}\{r(x(t),a(t))\\ &+\gamma\sum_{x(t+1)\in X}\Pr(x(t),a(t),x(t+1))\\ &\cdot Q^*(x(t+1))\}\end{aligned} \tag{15}$$

3　基于改进深度强化学习的VNF部署算法

由式(15)知，当获取到每时隙状态的最优值函数，便可得到状态对应的最优动作，且每时隙的最优动作就构成了最优策略π^*，由于本文中数据包的到达是随机的，状态转移概率很难获知，因此无法使用值迭代的方式进行求解，同时，传统的基于模型的优化方法通常要作一些假设，存在一定的局限性。根据式(7)，本文的关键问题是确定待VNF部

署的目标服务器和资源分配策略，文献[10]中的Q学习算法可以用来直接解决上述问题，但是本文的状态空间和动作空间都是连续值集合，Q学习算法面临维度灾、收敛速度慢等问题。本文接下来提出一种基于改进深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)的虚拟网络功能在线部署算法，该算法通过神经网络近似动作值函数，解决维度灾的问题，并且通过基于探索的值函数差异(Value-Difference Based Exploration, VDBE)方法[11]扩展ε贪婪策略，平衡代理在动作选取时的探索和利用问题，进一步针对传统深度强化学习算法从经验回放池中抽取训练样本利用效率低的问题，设置了双重经验回放池，加速神经网络的训练速度。改进的深度强化学习算法框架如图2所示。

在该算法框架中，式(14)中$Q^*(x(t),a(t))$的值是通过critic网络中估计Q网络近似得到，即

$$Q^*(x(t),a(t))\approx Q(x(t),a(t),\theta^Q) \tag{16}$$

其中，θ^Q表示估计Q网络的权重，$a(t)$是通过actor网络输出得到的。

critic网络中的TD-error定义为

$$\begin{aligned}\delta(t)=&r(x(t),a(t))+\gamma Q(x(t+1),a(t+1),\theta^{Q'})\\&-Q(x(t),a(t),\theta^Q)\end{aligned} \tag{17}$$

其中，$Q(x(t+1),a(t+1),\theta^{Q'})$通过目标$Q$网络获到，$a(t+1)$通过目标actor网络得到，$\theta^{Q'}$表示目标$Q$网络的权重。

进一步，估计Q网络的损失函数就可以表示为

$$L(\theta^Q)=\mathrm{E}[\delta(t)^2] \tag{18}$$

当从经验回放池中抽取N个样本时，损失函数就变为

$$L(\theta^Q)=\frac{1}{N}\sum_i(\delta(t)^2) \tag{19}$$

然后采用随机梯度下降算法对估计Q网络的参数进行更新。

actor网络的作用是得到一个确定性策略$\pi(x,\theta^\mu)$，其中θ^μ表示估计actor网络的权重，等到神经网络参数训练完成之后，就可以得到近似最优策略，即

$$\pi^*=\pi(x,\theta^\mu) \tag{20}$$

估计actor网络的参数主要是通过策略梯度$\nabla_{\theta^\mu}J$进行更新的，将其表示为

$$\begin{aligned}\nabla_{\theta^\mu}J=&\mathrm{E}[\nabla_{\theta^\mu}Q(x(t),a(t),\theta^Q)|_{x=x(t),a=\mu(x(t),\theta^u)}]\\=&\mathrm{E}[\nabla_aQ(x(t),a(t),\theta^Q)|_{x=x(t),a=\mu(x(t),\theta^u)}\\&\cdot\nabla_{\theta^\mu}\pi(x(t),\theta^\mu)|_{x=x(t)}]\end{aligned} \tag{21}$$

其中，$Q(x(t),a(t),\theta^Q)$是通过估计Q网络得到的。

在经验回放池抽取N个样本后，策略梯度可以转化为

$$\begin{aligned}\nabla_{\theta^\mu}J=&\frac{1}{N}\sum_i\nabla_aQ(x(i),a(i),\theta^Q)|_{x=x(i),a=\mu(x(i),\theta^u)}\\&\cdot\nabla_{\theta^\mu}\pi(x(i),\theta^\mu)|_{x=x(i)}\end{aligned} \tag{22}$$

最后，为了探索最优动作，筛选出更好的策略问题，引入随机噪声，从而获得动作，即

$$a(t)=\mu(x(t),\theta^\mu)+N \tag{23}$$

其中，N表示随机过程。

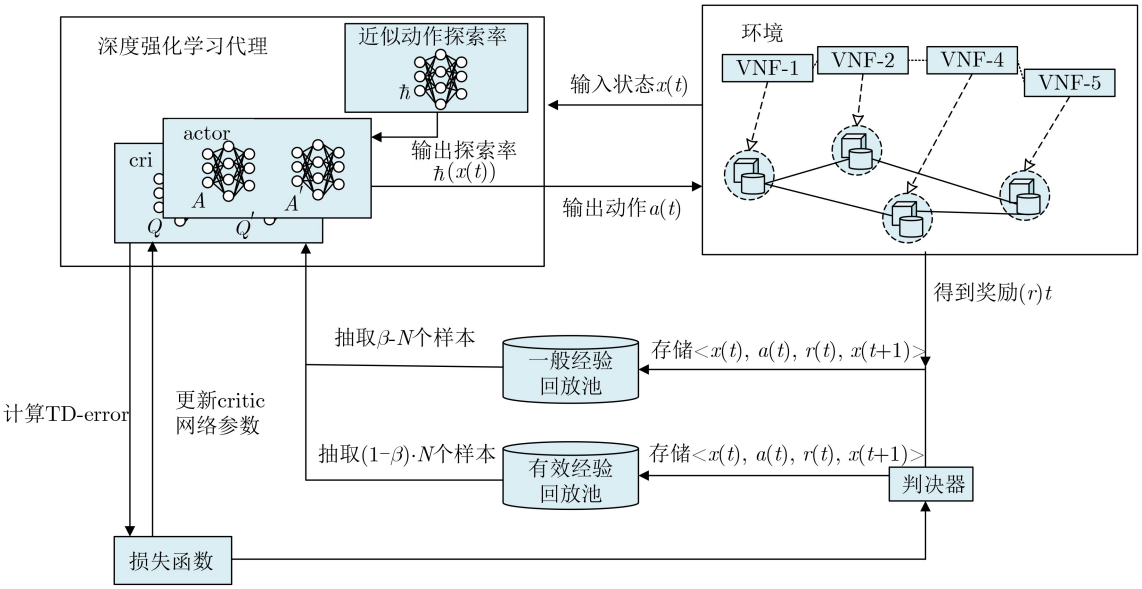

图 2 改进深度强化学习算法框架

目标Q网络的参数$\theta^{Q'}$和目标actor网络的参数$\theta^{\mu'}$通过软更新的方式进行更新，即

$$\theta^{Q'} \leftarrow \tau\theta^{Q}+(1-\tau)\theta^{Q'}, \theta^{\mu'} \leftarrow \tau\theta^{\mu'}+(1-\tau)\theta^{\mu'} \quad (24)$$

但是，传统的深度强化学习代理对环境采取动作的时候，主要采取的是ε贪婪策略，ε贪婪策略的思想是以ε的概率随机选取动作，以$1-\varepsilon$的概率选取由神经网络训练出的最优值函数对应的动作，即式(23)中的动作$a(t)$，其中ε的值是人为设定的，需要不断地调试，才能获得一个比较好的结果，而且该值是一个固定值，存在很大的随机性和误差，如果设定ε的值较大，则深度强化学习代理将过多地探索，导致代理学习效率低，甚至达不到长期最好收益，如果设定ε的值较小，则深度强化学习代理将注重利用，导致算法只能获得较少的收益，不能找到最优的动作，针对该问题，本文采用VDBE方法对ε贪婪策略进行扩展，该方法的主要思想是通过状态值函数的差异来确定ε的值，即在深度强化学习代理刚开始训练的时候，对环境是未知的，所以应该不断地探索，于是根据不同动作产生的状态值函数差异很大，随着深度强化学习代理对环境慢慢熟悉，不同动作得到的状态值函数差异变小，代理也应该降低它的探索率，于是我们就可以根据状态值函数的差距来动态地刻画概率ε的值。

根据动作值函数估计的玻尔兹曼分布[11]，每个时隙环境根据状态采取动作的概率用$\hbar(x(t))$表示，其更新公式表示为

$$f(x(t),a(t),\sigma)=\frac{1-\mathrm{e}^{\frac{-|Q(x(t+1),a(t+1))-Q(x(t),a(t))|}{\sigma}}}{1+\mathrm{e}^{\frac{-|Q(x(t+1),a(t+1))-Q(x(t),a(t))|}{\sigma}}} \quad (25)$$

$$h(x(t+1))=\delta\cdot f(x(t),a(t),\sigma)+(1-\delta)h(x(t)) \quad (26)$$

其中，σ表示一个正数，称为反向灵敏度，σ的值越小，探索概率就越大，$\delta\in[0,1)$表示所选动作对探索概率的影响参数，通常δ的值为当前状态下可以采取动作总数的倒数[11]，即$\delta=1/|A(x)|$。

在深度强化学习代理开始训练的时候，通常将$h(x(0))$设置为1。另外，由于本文的状态空间过大，因此本文采用深度神经网络去近似$h(x(t))$的值，然后将该深度神经网络定义为$\hbar$网络。

该神经网络的输入为状态$(x(t))$，输出为$\hat{h}(x(t))$，即

$$h(x(t))\approx\hat{h}(x(t)) \quad (27)$$

该神经网络的参数通过最小化损失函数$L(w)$进行训练，表示为

$$L(w)=\mathrm{E}[(y(t)-\hat{h}(x(t),w))^2] \quad (28)$$

其中，w表示网络的权重，$y(t)$表示目标值，将其表示为

$$y(t)=\delta\cdot f(x(t-1),a(t-1),\sigma)+(1-\sigma)\cdot\hat{h}(x(t-1)) \quad (29)$$

从经验回放池抽取N个样本后，h网络的损失函数通过式(27)计算得到

$$L(w)=\frac{1}{N}\sum_i\left(y(i)-\hat{h}(x(i),w)\right)^2 \quad (30)$$

最后采用随机梯度下降算法对h网络参数w进行更新。

另外，由于传统的深度强化学习算法对经验回放池的样本是随机采样，有些无用的样本被重复使用，导致样本的利用率低下，于是本文提出双重经验回放池架构，利用TD-error的值来区分样本的好坏，当TD-error的绝对值很大的，$\delta(t)>\Lambda$，其中，Λ表示一个正数，说明当前样本对神经网络的权重改变大，给神经网络带来的信息量多，在后面的训练过程中，优先选择该样本[12]。于是将这种样本存储到有效经验回放池中，另外，考虑到一般性，防止过拟合的现象发生，同时也需要从所有的样本值进行随机采样，于是在每个时隙，假设深度强化学习代理需要抽取的样本集大小为N，则从有效经验回放池中抽取$(1-\gamma)\cdot N$个样本，从一般经验回放池中抽取$\gamma\cdot N$个样本进行训练，其中γ表示权重值。

4　性能仿真与分析

为了评估算法的有效性以及收敛性，本文将SFC端到端时延、SFC部署成本、资源利用率等作为算法评价标准，对所提算法进行仿真验证。为了更好地评估基于改进深度强化学习的VNF在线部署算法的有效性，与文献[13]基于遗传算法(Genetic Algorithm,GA)的VNF部署算法，文献[14]中的DDPG算法以及文献[15]中的深度信念网络-服务功能链 (Deep Q Network-Service Function Chain, DQN-SFC)算法进行了对比，所有仿真实验均在i7 CPU和内存为8 GB的主机上运行，仿真平台主要基于Python 3.6，通过TensorFlow模块对神经网络进行搭建。

4.1　参数设置

本文假设物理网络为全连接网络，其中包括6台物理通用服务器，假设每条SFC由2～5个VNF构成，另外参考文献[9]对相关参数进行设置，具体网络参数如表1所示。

4.2　仿真分析

由于本文数据包的到达是随机的、状态空间

表 1 网络场景的仿真参数

仿真参数	值	仿真参数	值
数据包到达过程	泊松过程$\lambda_i=2$	数据包大小	500 kByte/packet
通用服务器总台数H	6台	物理链路带宽资源	640 MB
通用服务器v的CPU资源容量	8核	单个CPU服务速率β	25 MB/s
折扣因子γ	0.99	软更新因子τ	0.01
最大迭代轮数	2000	学习率α	$\{0.00001, 0.0001\}$
SFC的长度	Uniform[2,5]个	SFC的时延最长限制D_i	30 ms
正数∂	30	正数ς	20

大以及VNF部署和资源分配的动作空间维度高，因此采用改进深度强化学习算法去近似状态Q值，得到近似最优的VNF部署和资源分配策略，图3证明了本文所提改进深度强化学习算法收敛性，并且在相同的学习率$\alpha=0.0001$下，对比DDPG算法，在训练50次的时候，改进深度强化学习算法已经收敛，而DDPG算法大约在训练100次的时候收敛，因此改进的深度强化学习算法能够快速收敛，原因是本文所提算法在DDPG算法基础上，改进动作探索和利用，解决经验池样本利用率低的问题。

图4、图5分别表示在相同的SFC数量下，本文所提基于改进深度强化学习的VNF在线部署算法与其他3种算法在系统总时延和部署成本方面的对比。在此仿真过程中，设置权重值e1, e2都为0.5。

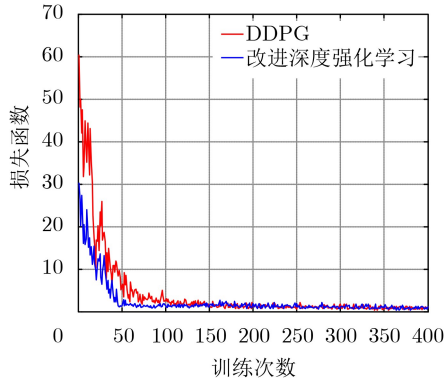

图 3 损失函数对比

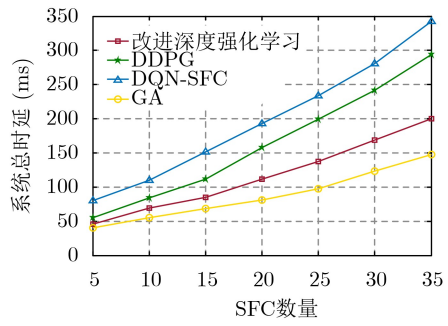

图 4 系统总时延对比

从图4、图5可以看出，随着SFC数量的增加，4种算法下的部署成本和系统总时延都将增大，由于GA算法只对时延进行优化，所以它的系统总时延是最低的，但是其没有考虑VNF的部署成本，没有进行合理的资源分配，导致其产生的部署成本是最大的，DQN-SFC算法由于同时优化时延和部署失败产生的运维开销，故它的总时延比GA算法要高，其次，DQN算法适用于解决离散的动作空间，由于本文的动作空间是连续值，故DQN-SFC算法在优化时延和VNF部署成本方面明显劣于DDPG算法和改进深度强化学习算法，而且改进深度强化学习算法比DDPG算法在部署成本和系统总时延方面更低，这是因为改进深度强化学习算法在DDPG算法基础上通过状态值函数差异去刻画每时隙动作的探索率，使其能够找到更优的动作，进而获得更好的奖励值，因此改进深度强化学习算法的

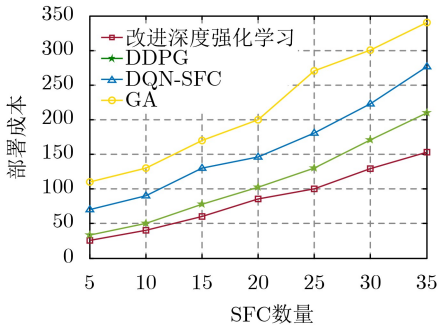

图 5 部署成本对比

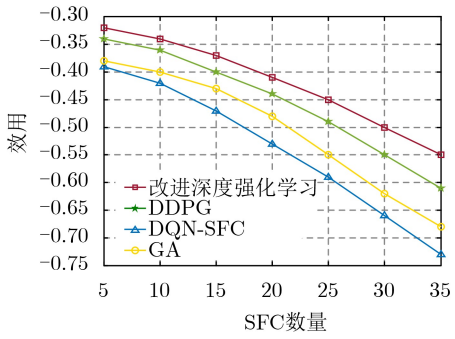

图 6 效用对比

效用是最大的，如图6所示，由于随SFC数量增加时系统总时延和总部署成本增大，导致效用也会随SFC数量增加而减少，其中，DQN-SFC算法的效用下降最快，而改进深度强化学习中的效用下降最慢。

5　结论

针对NFV/SDN架构下网络服务请求动态到达引起的VNF部署优化问题，本文首先将随机优化问题建立为一个MDP模型，该模型以最小化SFC部署成本和时延成本为目标，同时受限于每条SFC时延、通用服务器CPU资源以及物理链路带宽资源约束，其次提出一种基于改进深度强化学习的VNF在线部署算法，通过深度神经网络去近似最优的动作状态值函数，从而获得近似最优的VNF部署策略和资源分配策略，最后提出一种基于探索的值函数差异方法去动态地刻画动作探索率，并采用双重经验回放池，解决经验样本利用率低的问题。仿真结果显示，本文所提算法能够加速收敛，并能够优化SFC端到端时延和部署成本。

参 考 文 献

[1] 唐伦, 杨恒, 马润琳, 等. 基于5G接入网络的多优先级虚拟网络功能迁移开销与网络能耗联合优化算法[J]. 电子与信息学报, 2019, 41(9): 2079–2086. doi: 10.11999/JEIT180906.
TANG Lun, YANG Heng, MA Runlin, *et al*. Multi-priority based joint optimization algorithm of virtual network function migration cost and network energy consumption[J]. *Journal of Electronics & Information Technology*, 2019, 41(9): 2079–2086. doi: 10.11999/JEIT180906.

[2] 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[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(4): 1562–1576. doi: 10.1109/TNET.2018.2842798.

[3] VIZARRETA P, CONDOLUCI M, MACHUCA C M, *et al*. QoS-driven function placement reducing expenditures in NFV deployments[C]. 2017 IEEE International Conference on Communications (ICC), Paris, France, 2017: 1–7. doi: 10.1109/ICC.2017.7996513.

[4] XIONG Gang, HU Yuxiang, TIAN Le, *et al*. A virtual service placement approach based on improved quantum genetic algorithm[J]. *Frontiers of Information Technology & Electronic Engineering*, 2016, 17(7): 661–671. doi: 10.1631/FITEE.1500494.

[5] LUO Ziyue and WU Chuan. An online algorithm for VNF service chain scaling in datacenters[J]. *IEEE/ACM Transactions on Networking*, 2020, 28(3): 1061–1073. doi: 10.1109/TNET.2020.2979263.

[6] GHARBAOUI M, CONTOLI C, DAVOLI G, *et al*. Demonstration of latency-aware and self-adaptive service chaining in 5G/SDN/NFV infrastructures[C]. 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Verona, Italy, 2018: 1–2. doi: 10.1109/NFV-SDN.2018.8725645.

[7] CHENG Aolin, LI Jian, YU Yuling, *et al*. Delay-sensitive user scheduling and power control in heterogeneous networks[J]. *IET Networks*, 2015, 4(3): 175–184. doi: 10.1049/iet-net.2014.0026.

[8] YANG Jian, ZHANG Shuben, WU Xiaomin, *et al*. Online learning-based server provisioning for electricity cost reduction in data center[J]. *IEEE Transactions on Control Systems Technology*, 2017, 25(3): 1044–1051. doi: 10.1109/TCST.2016.2575801.

[9] 唐伦, 杨恒, 赵国繁, 等. 基于时延感知的5G网络切片节点和链路映射算法[J]. 北京邮电大学学报, 2018, 41(6): 71–77. doi: 10.13190/j.jbupt.2018-018.
TANG Lun, YANG Heng, ZHAO Guofan, *et al*. Delay-aware 5G network slicing node and link embedding algorithm[J]. *Journal of Beijing University of Posts and Telecommunications*, 2018, 41(6): 71–77. doi: 10.13190/j.jbupt.2018-018.

[10] WANG Zhuzhu, LIU Yang, MA Zhou, *et al*. LiPSG: lightweight privacy-preserving Q-learning-based energy management for the IoT-Enabled smart grid[J]. *IEEE Internet of Things Journal*, 2020, 7(5): 3935–3947. doi: 10.1109/JIOT.2020.2968631.

[11] TOKIC M. Adaptive ε-greedy exploration in reinforcement learning based on value differences[C]. The 33rd Annual German Conference on KI 2010: Advances in Artificial Intelligence, Karlsruhe, Germany, 2010: 203-210.

[12] CAO Xi, WAN Huaiyu, LIN Youfang, *et al*. High-value prioritized experience replay for off-policy reinforcement learning[C]. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, USA, 2019: 1510–1514. doi: 10.1109/ICTAI.2019.00215.

[13] 陈卓, 冯钢, 刘蓓, 等. 运营商网络中面向时延优化的服务功能链迁移重配置策略[J]. 电子学报, 2018, 46(9): 2229–2237. doi: 10.3969/j.issn.0372-2112.2018.09.026.
CHEN Zhuo, FENG Gang, LIU Bei, *et al*. Delay optimization oriented service function chain migration and re-deployment in operator network[J]. *Acta Electronica Sinica*, 2018, 46(9): 2229–2237. doi: 10.3969/j.issn.0372-2112.2018.09.026.

[14] LI Han, LÜ Tiejun, and ZHANG Xuewei. Deep deterministic policy gradient based dynamic power control for self-powered ultra-dense networks[C]. 2018 IEEE

Globecom Workshops (GC Wkshps), Abu Dhabi, 2018: 1–6. doi: 10.1109/GLOCOMW.2018.8644157.

[15] 金明, 李琳琳, 张文瑾, 等. 基于深度强化学习的服务功能链映射算法[J]. 计算机应用研究, 2020, 37(11): 3456–3460, 3466.
JIN Ming, LI Linlin, ZHANG Wenjin, *et al*. SFC mapping algorithm based on deep reinforcement learning[J]. *Application Research of Computers*, 2020, 37(11): 3456–3460, 3466.

唐 伦：男，1973年生，教授，博士，研究方向为下一代无线通信网络、异构蜂窝网络、软件定义无线网络等.

贺兰钦：男，1995年生，硕士生，研究方向为5G网络切片、机器学习算法.

谭 颀：女，1995年生，硕士生，研究方向为5G网络切片、资源分配、随机优化理论.

责任编辑：余 蓉