

基于深度确定性策略梯度的虚拟网络功能迁移优化算法

唐 伦 贺兰钦* 谭 颀 陈前斌

(重庆邮电大学通信与信息工程学院 重庆 400065)

(重庆邮电大学移动通信技术重点实验室 重庆 400065)

摘 要: 针对NFV/SDN架构下, 服务功能链(SFC)的资源需求动态变化引起的虚拟网络功能(VNF)迁移优化问题, 该文提出一种基于深度强化学习的VNF迁移优化算法。首先, 在底层CPU、带宽资源和SFC端到端时延约束下, 建立基于马尔可夫决策过程(MDP)的随机优化模型, 该模型通过迁移VNF来联合优化网络能耗和SFC端到端时延。其次, 由于状态空间和动作空间是连续值集合, 提出一种基于深度确定性策略梯度(DDPG)的VNF智能迁移算法, 从而得到近似最优的VNF迁移策略。仿真结果表明, 该算法可以实现网络能耗和SFC端到端时延的折中, 并提高物理网络的资源利用率。

关键词: 虚拟网络功能; 深度强化学习; SFC端到端时延; 网络能耗

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2021)02-0404-08

DOI: 10.11999/JEIT190921

Virtual Network Function Migration Optimization Algorithm Based on Deep Deterministic Policy Gradient

TANG Lun HE Lanqin TAN Qi CHEN Qianbin

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

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

Abstract: To solve the problem of Virtual Network Function (VNF) migration optimization, which is caused by the dynamic change of resource requirements of Service Function Chain (SFC) under Network Function Virtualization/ Software Defined Network (NFV/SDN) architecture, a VNF migration optimization algorithm is proposed based on deep reinforcement learning. Firstly, based on the underlying CPU, bandwidth resources and SFC end-to-end delay constraints, a Markov Decision Process (MDP) based stochastic optimization model is established. This model is used to optimize jointly network energy consumption and SFC end-to-end delay by migrating VNF. Secondly, since the state space and action space of this paper are continuous value sets, a VNF intelligent migration algorithm based on Deep Deterministic Policy Gradient (DDPG) is proposed to obtain an approximate optimal VNF migration strategy. The simulation results show that the algorithm can achieve the compromise between network energy consumption and SFC end-to-end delay, and improve the resource utilization of the physical network.

Key words: Virtual Network Function (VNF); Deep reinforcement learning; SFC end-to-end delay; Network energy consumption

收稿日期: 2019-11-15; 改回日期: 2020-11-02; 网络出版: 2020-12-09

*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-M20180601), 重庆市重大主题专项项目 (cstc2019jscx-zdztzxX0006)

Foundation Items: The National Natural Science Foundation of China (62071078), The Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-M20180601), The Major Theme Special Projects of Chongqing (cstc2019jscx-zdztzxX0006)

1 引言

在软件定义网络(Software Defined Network, SDN)环境下, 网络功能虚拟化(Network Function Virtualization, NFV)技术根据当前的用户服务请求创建服务功能链(Service Function Chain, SFC), SFC由多个虚拟网络功能(Virtual Network Function, VNF)按照特定的顺序编排构成^[1], 将VNF部署在不同服务器上, 进而为用户提供服务, 增强了网络的灵活性以及拓展性, 使得基础设施资源高效利

用^[2]。由于SFC的资源需求是动态变化的,带来了资源利用率低等问题,因此研究VNF迁移机制十分有必要。

目前,已有大量工作对VNF迁移机制展开了研究,主要考虑触发VNF迁移的时机,待迁移VNF的选择和迁移的目的节点选择,文献^[3]考虑在某台服务器的资源利用率过低时,触发VNF聚合,将该服务器上的所有VNF迁移至其他服务器,进入休眠状态,降低系统的整体能量消耗,但是没有考虑用户服务质量(Quality of Service, QoS)降低,而且当服务器的资源利用率达到阈值时立即触发迁移,会造成频繁迁移,降低系统的稳定性。文献^[4]主要考虑低时延网络,当系统监测到VNF迁移后能够降低SFC端到端时延,则触发迁移,但是忽略了VNF迁移后网络能耗的增加。文献^[5]建立成本模型来评估迁移开销,同时提出一种贪婪算法最小化VNF迁移开销,但是它采用的是静态迁移策略,由于网络环境是动态变化的,需要考虑长时间的优化。

综上所述,现有的VNF迁移机制均没有对网络能耗最小化及SFC端到端时延最小化进行联合考虑,而且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,另外,针对有生命周期的SFC,在长时间尺度下研究VNF迁移的工作并不多。因此本文提出了一种基于深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)的VNF迁移方案,该方案首先考虑SFC资源需求动态变化,在保证底层物理资源和用户QoS需求的前提下,通过VNF迁移,并确定每个通用物理服务器的工作状态,实现网络能耗与SFC端到端时延的联合优化。其次,由于本文所提

问题为随机优化问题,因此将该优化问题转换成马尔可夫决策过程(Markov Decision Processes, MDP)。最后,由于本文的状态转移概率是未知的,状态和动作空间都是连续值集合,故提出一种基于DDPG的VNF迁移算法求解MDP,得出近似最优的VNF迁移策略。

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。应用层主要为网络业务请求创建SFC,通过SFC为用户提供服务。虚拟化层主要负责网络状态监控和底层网络负载分析,物理层为SFC提供其实例化的物理资源,包括计算资源和链路带宽资源,物理网络主要由通用服务器组成。

2.2 网络模型

2.2.1 物理网络

底层物理网络由全连接无向图 $G^P = (N^P, E^P)$ 组成,其中 $N^P = N_{\text{off}}^P + N_{\text{on}}^P$, N_{off}^P 表示底层物理服务器处于关闭状态的集合, N_{on}^P 表示底层物理服务器处于工作状态的集合, E^P 表示底层链路集合。每个底层物理服务器 $n \in N^P$ 的CPU容量为 C_n ,连接相邻底层物理服务器 n 和 m 的链路 nm 的带宽为 B_{nm} 。 φ_n 为二进制变量, $\varphi_n = 1$ 表示服务器 n 处于工作状态, $\varphi_n = 0$ 表示服务器 n 处于关闭状态。

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 $G_i^V = (N_i^V, L_i^V)$, N_i^V 表示第 i 条SFC上的不同类型VNF的集合, L_i^V 表示第 i 条SFC上的虚拟链路集合,第 i 条SFC上的第 j 个VNF表示为 $N_{i,j}^V$,且它

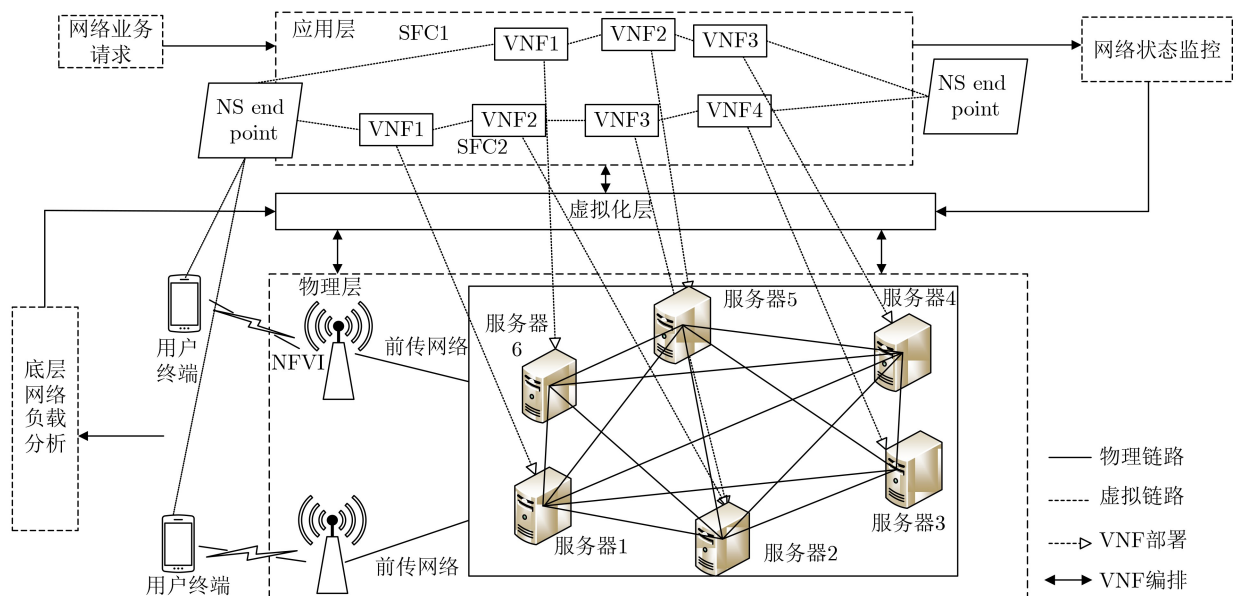


图1 系统模型

的CPU资源需求为 $C_{i,j}^V$, 连接相邻VNF的虚拟链路带宽需求为 $B_{i,jk}^V$, 第 i 条SFC的最长时延限制为 D_i 。接下来定义两个布尔变量 $\theta_{i,j}^m$, $\lambda_{i,jk}^{mn}$, 当第 i 条SFC上的第 j 个VNF映射到服务器 m 时, $\theta_{i,j}^m = 1$, 否则 $\theta_{i,j}^m = 0$ 。 $\lambda_{i,jk}^{mn} = 1$ 表示第 i 条SFC上连接VNF j 和 k 的虚拟链路映射到物理链路 mn 上, 反之 $\lambda_{i,jk}^{mn} = 0$ 。

2.2.3 SFC时延模型

本文在计算SFC端到端时延的时候, 因为传播时延的值可以忽略不计, 所以主要考虑处理时延和传输时延。

在 t 时隙, SFC的处理时延受物理服务器的负载 $C_m^o(t)$ 影响, 负载 $C_m^o(t)$ 用式(1)表示。

$$C_m^o(t) = \frac{\sum_{i \in F} \sum_{j \in N_i^V} \theta_{i,j}^m(t) C_{i,j}^V(t)}{C_m} \quad (1)$$

当映射到服务器上的VNF资源需求增加时, 将导致服务器的CPU负载增大, 进而SFC的处理时延也将增大。参考文献[7], 假设处理时延是处理负载的凸函数, 使用分段线性化来近似凸时延曲线。所以底层物理服务器的处理时延可以表示为

$$d_m^p(t) = \chi_i + \varepsilon_i C_m^o(t) \quad (2)$$

其中, ε_i 和 χ_i 表示近似凸时延函数曲线的线性函数的系数。

SFC的传输时延与SFC上的虚拟链路映射到物理链路的位置有关, 假设每跳虚拟链路映射到物理链路的时延为 d_{jk} , 第 i 条SFC传输时延为

$$d_i^t(t) = \sum_{j,k \in N_i^V} \lambda_{i,jk}^{mn}(t) d_{jk}(t) \quad (3)$$

因此, 第 i 条SFC的端到端时延可以表示为

$$T_i(t) = \sum_{j \in N_i^V} \theta_{i,j}^m(t) d_m^p(t) + d_i^t(t) \quad (4)$$

2.2.4 网络能耗模型

本文的网络能耗主要考虑底层物理服务器的能耗, 网络能耗包括服务器运行状态时能耗和服务器状态切换时能耗 $P_m^c(t)$, 基于文献[8], 服务器的运行状态时能耗分为服务器待机时的能耗 $P_m^n(t)$ 和服务器CPU资源负载产生的能耗 $P_m^v(t)$, 因此服务器 m 在 t 时隙的待机能耗和CPU负载能耗, 分别用式(5)和式(6)表示。

$$P_m^n(t) = \varphi_m(t) P_{\text{idle}} \quad (5)$$

$$P_m^v(t) = C_m^o(t) \cdot P_m^{\text{cpu}} \quad (6)$$

其中, P_{idle} 表示服务器待机时产生的能耗, P_m^{cpu} 为服务器 m 中CPU资源全部占用时所产生的能耗。

另外, 参考文献[9], 基站切换工作状态时会有

能量消耗。本文假设服务器工作状态发生变化时会产生能耗 P_m^c , 服务器 m 在 t 时隙产生的切换能耗为

$$P_m^c(t) = \eta_m(t) \cdot P_c \quad (7)$$

其中, $\eta_m(t)$ 表示服务器 m 在 t 时隙状态是否发生了改变, 用式(8)表示。

$$\eta_m(t) = \begin{cases} 0, & \varphi_m(t-1) = \varphi_m(t) \\ 1, & \varphi_m(t-1) \neq \varphi_m(t) \end{cases} \quad (8)$$

因此网络能耗可以通过每台服务器的能耗之和表示, 即

$$P^{\text{tot}}(t) = \sum_{m \in N^P} (P_m^n(t) + P_m^v(t) + P_m^c(t)) \quad (9)$$

2.3 优化目标

本文要解决的问题是: 在最小化网络能耗的前提下最小化SFC端到端时延, 同时把底层物理资源等作为限制条件。故该优化问题属于多目标优化问题, 具体表示为

$$\begin{aligned} \min_{\theta_{i,j}^n(t)} \quad & a_1 \cdot \frac{P^{\text{tot}}(t)}{P_{\text{max}}} + a_2 \cdot \frac{\sum_{i \in F} T_i(t)}{\sum_{i \in F} D_i} \\ \text{s.t.} \quad & \text{C1: } \sum_{n \in N^P} \sum_{i \in F} \theta_{i,j}^n(t) = 1, \forall j \in N_i^V \\ & \text{C2: } \sum_{i \in F} \sum_{j \in N_i^V} \theta_{i,j}^n(t) \cdot C_{i,j}^V(t) \leq C_n, \forall n \in N^P \\ & \text{C3: } \sum_{i \in F} \sum_{j \in N_i^V} \theta_{i,j}^n(t) \cdot C_{i,j}^V(t) \geq C_n \cdot r_c, \forall n \in N^P \\ & \text{C4: } \sum_{i \in F} \sum_{j \in N_i^V} (\lambda_{i,jk}^{mn}(t) + \lambda_{i,jk}^{nm}(t)) B_{i,jk}^V(t) \leq B_{nm}, \\ & \quad \forall mn \in E^P \\ & \text{C5: } \sum_{i \in F} \sum_{j \in N_i^V} (\lambda_{i,jk}^{mn}(t) + \lambda_{i,jk}^{nm}(t)) B_{i,jk}^V(t) \geq B_{nm} \cdot r_b, \\ & \quad \forall mn \in E^P \\ & \text{C6: } \sum_{i \in F} \sum_{j \in N_i^V} \lambda_{i,jk}^{mn}(t) - \sum_{i \in F} \sum_{j \in N_i^V} \lambda_{i,jk}^{nm}(t) = \theta_{i,j}^n(t) \\ & \quad - \theta_{i,k}^n(t), \forall n \in N^P, \forall nm \in E^P \\ & \text{C7: } T_i(t) \leq D_i, \forall i \in F \\ & \text{C8: } \sum_{i \in F} \sum_{j \in N_i^V} \theta_{i,j}^n(t) \geq 1, \text{ 则 } \varphi_n(t) = 1, \text{ 否则 } \varphi_n(t) = 0, \\ & \quad \forall n \in N^P \\ & \text{C9: } \theta_{i,j}^n(t) = 0, 1, \forall i \in F, \forall j \in N_i^V, \forall n \in N^P \\ & \text{C10: } \lambda_{i,jk}^{nm}(t) = 0, 1, \forall i \in F, \forall j, k \in N_i^V, \forall n, m \in N^P \\ & \text{C11: } \varphi_n(t) = 0, 1, \forall n \in N^P \end{aligned} \quad (10)$$

其中, a_1, a_2 是权重系数, 且 $a_1 + a_2 = 1$, P_{max} 表示网络能耗的最大值。

该优化目标受到C1~C11限制条件约束，保证优化目标的有效性。C1用于保证一条SFC上的VNF只能选择一个服务器进行映射。C2~C5表示物理网络CPU资源和带宽资源限制。C6表示SFC中相邻两个VNF映射到底层服务器 n 和 m 时， nm 之间必须存在一条连续路径，满足流守恒原则。C7表示每条SFC在任意时隙都要满足时延要求。C8表示如果某一台服务器上有VNF映射，那么这台服务器一定处于开启状态。C9~C11分别表示VNF映射，虚拟链路映射和底层服务器状态的二进制变量约束。

3 本文算法

3.1 MDP模型

本文先将优化问题建模成MDP模型，得到深度强化学习的基础框架，用一个四元组 (S, A, P, R) 对其进行描述。

(1) 状态空间为 S ：定义状态空间为 $S(t) = \{(\varphi(t), c(t), b(t)) | \forall \varphi \in \psi, \forall c \in C, \forall b \in B\}$ ，其中， φ 表示底层物理网络拓扑， ψ 表示网络拓扑状态空间， $c(t), b(t)$ 分别表示在 t 时隙VNF的CPU资源和链路带宽资源需求，而 C, B 则分别表示VNF的CPU资源和虚拟链路带宽资源需求状态空间：

(2) 动作空间为 A ：在 t 时隙定义动作空间为 $A(t) = \{\theta(t) | \theta \in \Theta\}$ ，其中 θ 表示VNF的映射动作， Θ 则表示VNF的映射动作空间。

(3) 状态转移概率为 P ：在 t 时隙状态 $s(t)$ 采取动作 $a(t)$ ，将转移到下一时隙的状态 $s(t+1)$ ，状态转移概率为 $P(s(t+1)|s(t), a(t))$ 。

(4) 奖励函数 R ：本文定义一个效用函数 $r(t)$ 作为即时奖励，通过式(10)转化得到

$$r(t) = -a_1 \cdot \frac{P_{\text{total}}(t)}{P_{\text{max}}} - a_2 \cdot \frac{\sum_{i \in F} \text{delay}_i(t)}{\sum_{i \in F} D_i} \quad (11)$$

环境在 t 时隙处于状态 $s(t)$ ，会根据策略 π 采取一个动作 $a(t) = \pi(s(t))$ 。通过该映射策略 π ，环境在 t 时隙会通过值函数 $Q^\pi(s(t), a(t))$ 来评判采取映射动作的好坏，可以将其表示为

$$\begin{aligned} Q^\pi(s(t), a(t)) \\ = E[r(t) + \mu \cdot Q^\pi(s(t+1), a(t+1)) | s(t), a(t)] \end{aligned} \quad (12)$$

VNF最优迁移策略就可以表示为

$$\pi^* = \arg \max_a Q^\pi(s, a) \quad (13)$$

3.2 基于DDPG的VNF迁移算法

MDP模型通常由动态规划(Dynamic Programming,

DP)或强化学习(Reinforcement Learning, RL)方法来解决^[10]，因此本文采用深度强化学习算法进行求解，但是常见的深度强化学习算法如深度Q学习(Deep Q Network, DQN)^[11]等不适合解决连续动作问题，因此本文提出一种基于DDPG的虚拟网络功能迁移算法，DDPG算法以确定性策略梯度(Deterministic Policy Gradient, DPG)算法为基础，首先采用行动者-评判家(Actor-Critic, AC)框架，其次在评判家网络和行动者网络里面采用双神经网络架构，分别构成了估计行动者网络(θ^π)，目标行动者网络($\theta^{\pi'}$)，估计评判家网络(θ^Q)与目标评判家网络($\theta^{Q'}$)，其中 $\theta^\pi, \theta^{\pi'}, \theta^Q$ 和 $\theta^{Q'}$ 分别代表各个深度神经网络的权重值，并且它们具有相同的神经网络架构。最后采用经验回放池打破数据的关联性和非平稳分布问题。

在评判家网络中，估计critic网络负责近似式(13)中的 Q 值函数，即

$$Q^\pi(s(t), a(t)) \approx Q_{\theta^Q}(s(t), a(t)) \quad (14)$$

近似过程通过深度神经网络(Deep Neural Network, DNN)完成。首先对输入的状态 $s(t)$ 和VNF映射动作 $a(t)$ 进行预处理，文献[12]通过状态标准化来预处理训练样本状态，加快梯度下降求最优解的速度，进而加快神经网络模型的训练，因此本文对状态和动作做归一化处理。

最后，DNN的输出为当前状态 $s(t)$ 采取VNF映射动作 $a(t)$ 对应的 $Q(s(t), a(t))$ 值，而在对估计评判家网络进行参数更新时，主要通过最小化损失函数，即

$$L(\theta^Q) = E[y(t) - Q(s(t), a(t) | \theta^Q)]^2 \quad (15)$$

其中， $y(t) = r(t) + \beta \max_{a(t+1)} Q_{\theta^{Q'}}(s(t+1), a(t+1))$ 为目标值，通过目标评判家网络得到。另外，VNF映射动作 $a(t)$ 通过估计行动者网络得到。

当损失函数对参数 θ^Q 连续可微时，估计评判家网络参数可以用损失函数的梯度进行更新，即

$$\Delta \theta^Q = l_c [y(t) - Q_{\theta^Q}(s(t), a(t))] \nabla_{\theta^Q} Q_{\theta^Q}(s(t), a(t)) \quad (16)$$

其中， l_c 表示估计评判家网络的学习率。

在行动者网络中，估计行动者网络使用策略梯度算法评估和改进策略，该算法会迭代评估策略，然后按照梯度进行改进以优化策略。策略梯度更新思想为：代理根据采取某种动作得到结果的好坏而决定是否增加或者减少采取该动作的可能性。

估计行动者网络同样采取DNN架构，将预处理后的状态 $\hat{s}(t)$ 作为输入，输出为当前环境状态 $s(t)$ 采取的策略 $\pi(s(t) | \theta^\pi)$ 。对估计行动者网络参数更新时，先从经验回放池随机抽取一部分样本训练

DNN参数,接着以最大化长期平均回报为目标,迭代改进策略,策略目标函数可以表示为

$$J(\pi_{\theta^{\pi}}) = E\{Q^{\pi}(s(t), a(t))\} \\ = \sum_S d(s(t)) \sum_A \pi_{\theta^{\pi}}(a(t)|s(t)) Q^{\pi}(s(t), a(t)) \quad (17)$$

其中, $d(s(t))$ 表示状态分布,然后对策略目标函数偏微分,得到策略目标函数梯度

$$\nabla_{\theta^{\pi}} J(\pi_{\theta^{\pi}}) = \sum_S d(s(t)) \sum_A \nabla_{\theta^{\pi}} \pi_{\theta^{\pi}}(a(t)|s(t)) Q^{\pi}(s(t), a(t)) \quad (18)$$

其中, $Q^{\pi}(s(t), a(t))$ 通过估计critic网络近似得到。如果式(14)中, $Q_{\theta^Q}(s(t), a(t))$ 很接近 $Q^{\pi}(s(t), a(t))$, 就可以用 $Q_{\theta^Q}(s(t), a(t))$ 代替 $Q^{\pi}(s(t), a(t))$, 将式(18)转化为

$$\nabla_{\theta^{\pi}} J(\pi_{\theta^{\pi}}) = \sum_S d(s(t)) \sum_A \nabla_{\theta^{\pi}} \pi_{\theta^{\pi}}(a(t)|s(t)) Q_{\theta^Q}(s(t), a(t)) \quad (19)$$

最后沿着目标策略函数梯度上升方向更新DNN参数

$$\Delta \theta^{\pi} = l_a \nabla_{\theta^{\pi}} J(\pi_{\theta^{\pi}}) \quad (20)$$

其中, l_a 表示估计actor网络的学习率。

为了探索VNF迁移最优动作,筛选出更好的策略问题,DDPG算法引入随机噪声 n , 从而获得动作 $a(t)$, 即

$$a(t) = \pi(s(t)|\theta^{\pi}) + n \quad (21)$$

其中, n 表示高斯随机噪声,均值为 n_0 。

目标评判家网络和目标行动者网络采用软更新参数,可以将其表示为

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}, \theta^{\pi'} \leftarrow \tau \theta^{\pi} + (1 - \tau) \theta^{\pi'} \quad (22)$$

具体的VNF迁移策略训练如表1所示。

本文算法的时间复杂度主要与状态的维度 $\aleph(s)$, 神经网络的层数 i 及每层的神经元个数 v_i 有关, 假设行动者网络有 A 个全连接层, 评判家网络有 Z 个全连接层, 在考虑每层有偏置单元的情况下, 时

表 1 基于DDPG的迁移策略训练算法

输入: DDPG参数: 回合数 M , 训练次数 T , 训练样本长度 L , 评判家网络学习率 l_c , 行动者学习率 l_a , 折扣因子 μ , 软更新因子 τ , 经验回放池大小 B , 最小样本长度 N , 高斯噪声 n
输出: 策略 π

1. 初始化经验回放池 H
2. 随机初始化行动者网络参数 $(\theta^{\pi}, \theta^{\pi'})$ 和评判家网络参数 $(\theta^Q, \theta^{Q'})$
3. **for** episode = 1, 2, ..., M **do**
4. 初始化环境 $s(0)$
5. **for** $t = 1, 2, \dots, T$ **do**
6. 根据当前策略得到动作, 增添随机噪声进行探索: $a(t) = \pi(s(t)|\theta^{\pi}) + n$
7. **if** C1~C10约束满足 **then**
8. 采取动作 $a(t)$, 得到状态 $s(t+1)$, 根据式(11)得到 $r(t)$
9. **if** 经验回放池 H 没有溢出 **then**
10. 将 $(s(t), a(t), r(t), s(t+1))$ 存储到经验池中
11. **else**
12. 用 $(s(t), a(t), r(t), s(t+1))$ 随机替代存入经验池的集合
13. 随机选择 N 个集合构成样本: $(s(i), a(i), r(i), s(i+1)), \forall i = 1, 2, \dots, N$
14. 通过目标评判家网络得到 $Q(s(i+1), a(i+1)|\theta^{Q'})$, 然后根据式(15)得到损失函数 $L(\theta^Q)$
15. 通过式(16)来更新估计评判家网络参数
16. 从估计评判家网络得到 $Q_{\theta^Q}(s(i), a(i))$, 通过式(19)得到策略梯度 $\nabla_{\theta^{\pi}} J(\pi_{\theta^{\pi}})$
17. 通过式(20)更新估计行动者网络的参数
18. 通过式(22)软更新目标评判家网络和目标行动者网络的参数
19. **end if**
20. **end if**
21. **end for**
22. **end for**

间复杂度可以表示为 $O\left(\sum_{a=0}^{A-1} v_{\text{act},a} v_{\text{act},a+1} + \sum_{z=0}^{Z-1} v_{\text{crit},z} v_{\text{crit},z+1}\right) + O(N(s))$ 。

4 性能仿真与分析

本文通过调整评判家网络的学习率 lc ，保持行动者网络的学习率不变，验证DDPG算法的收敛性能。由图2可以看出，当评判家网络学习率 lc 为0.0020时，损失函数的收敛速度最快，抖动最明显，但是梯度可能会在最小值附近来回波动，容易得到局部极小值，随着学习率的降低，收敛速度较缓慢，但是损失函数曲线较平稳，会使得在收敛时，损失函数值更加趋近于最优值。由图3可以看出，当评判家网络学习率 lc 为0.0010时，得到的累积奖励值最多，更加利于本文求得最优解。因此本

文选取评判家学习率 lc 为0.0010。

本文将权重 a_1 分别取值为0.7, 0.5和0.3，验证算法的有效性，考虑应用层有10条SFC的情况下，经过15000次训练迭代后，得到系统总时延和网络能耗情况如图4、图5所示。随着 a_1 的减小，系统总时延占的优化比重减小，故收敛时的时延值逐渐增大。随着 a_2 的增大，网络能耗占的优化比重增大，故收敛时的能耗值逐渐增大。基于以上分析，本文算法能够在优化网络能耗的同时，降低迁移带来的时延增量，保证用户的QoS。

本文首先从DDPG算法可以实现系统总时延与网络能耗联合优化的角度，与迁移仅实现SFC端到端时延最小化的实时VNF迁移(Virtual Network Functions Real-time Migration, VNF-RM)算法^[4]，及仅实现网络能耗最小化的VNF映射最小化能耗

仿真参数

仿真参数	值	仿真参数	值
通用服务器数量	12	服务器CPU容量(MIPS)	Uniform[250, 300]
VNF集合长度(个)	Uniform[2, 5]	链路带宽容量(Mbps)	Uniform[100, 200]
r_b	0.2	服务器CPU总能耗(W)	Uniform[170, 230]
r_c	0.2	服务器待机时的能耗(W)	Uniform[90, 120]
SFC最长时延限制(ms)	30	服务器状态切换能耗(W)	Uniform[25, 35]
软更新因子	0.01	虚拟链路带宽资源(Mbps)	Uniform[5, 10]
折扣因子	0.99	VNF CPU资源需求(MIPS)	Uniform[10, 20]

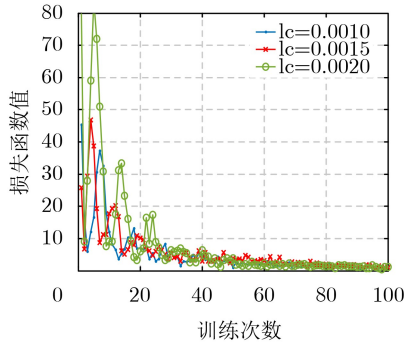


图2 不同评判家学习率下的损失函数值

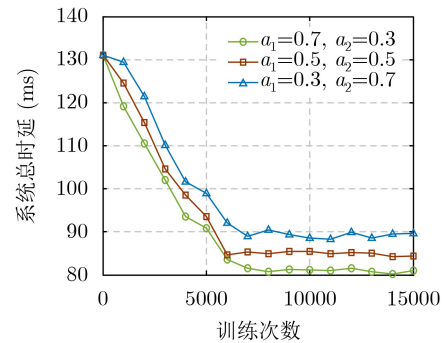


图4 权重对系统总时延的影响

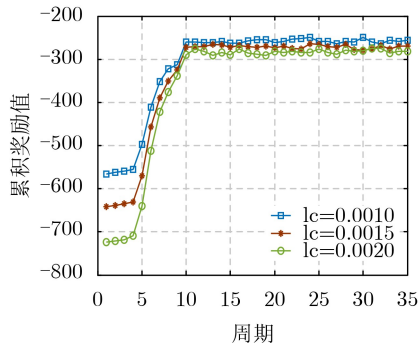


图3 不同评判家学习率下的累积奖励值

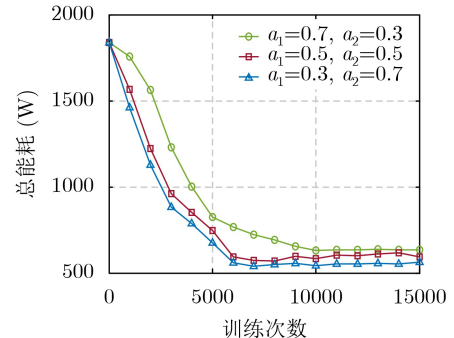


图5 权重对系统总能耗的影响

(VNFI Mapping Minimizing the Power Consumption, VMMPC)算法^[13]进行对比, 验证算法的性能。此时 $a_1=0.5$, $a_2=0.5$ 。

从图6, 图7可以看出, 由于VNF-RM算法只考虑在VNF迁移前后, SFC端到端时延是否减少, 而忽略了对底层低负载服务器的整合, 因此通过VNF-RM算法得到的能耗必然会高于通过DDPG算法得到的能耗。同样地, VMMPC算法尽可能整合底层服务器资源, 最小化网络能耗, 但是没有考虑SFC端到端时延是否增加, 因此通过VMMPC算法得到的SFC端到端时延必然会大于通过DDPG算法得到的SFC端到端时延。基于以上分析, DDPG算法能够实现网络能耗与系统总时延的联合优化。

其次, 本文与同样可以实现网络能耗与SFC端到端时延联合优化的AC(Actor-Critic)算法进行对比, 得到物理网络计算资源利用率情况如图8所示, 由于AC算法主要取决于Critic的价值判断, 但是Critic难收敛, 再加上Actor的更新, 就更难收敛, 故AC算法很容易得到一个局部最优解, DDPG算法通过神经网络近似价值函数, 解决了收敛难问题, 因此通过AC算法得到的资源利用率必然要低于通过DDPG算法得到的资源利用率, 进一步说明在VNF迁移过程中, DDPG算法能够有效提升物理网络计算资源利用率。

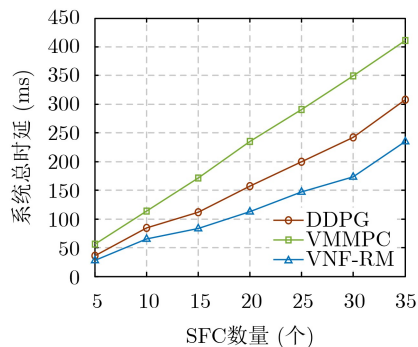


图6 系统总时延对比

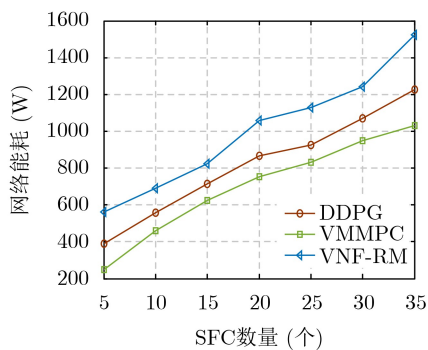


图7 网络能耗对比

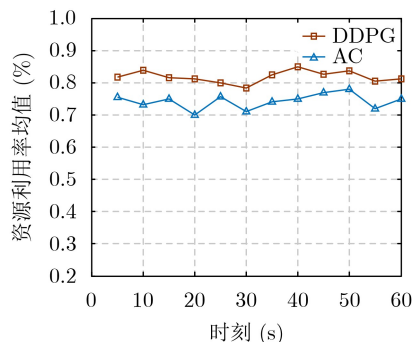


图8 计算资源利用率

5 结束语

本文基于NFV/SDN架构, 针对SFC资源需求动态变化引起的VNF迁移问题, 联合优化SFC时延与系统能耗, 然后将该随机优化问题建立成MDP模型, 由于状态转移概率未知, 以及状态空间过大和动作维度高, 本文提出一个基于DDPG算法的VNF迁移方案, 通过VNF迁移, 实现收益最大化。仿真结果显示, 本文所提算法能够有效地收敛, 并可以实现SFC时延和网络能耗折中, 提高服务器资源利用率。

参考文献

- [1] 唐伦, 周钰, 杨友超, 等. 5G网络切片场景中基于预测的虚拟网络功能动态部署算法[J]. 电子与信息学报, 2019, 41(9): 2071–2078. doi: 10.11999/JEIT180894.
TANG Lun, ZHOU Yu, YANG Youchao, et al. Virtual network function dynamic deployment algorithm based on prediction for 5g network slicing[J]. *Journal of Electronics & Information Technology*, 2019, 41(9): 2071–2078. doi: 10.11999/JEIT180894.
- [2] 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.
- [3] 程国振. 基于元能力的网络功能组合关键技术研究[D]. [博士学位论文], 解放军信息工程大学, 2015.
CHENG Guozhen. Research on the key technologies of function composition based on atomic capability[D]. [Ph. D. dissertation], Information Engineering University, 2015.
- [4] CHO D, TAHERI J, ZOMAYA A Y, et al. Real-time virtual network function (VNF) migration toward low network latency in cloud environments[C]. The 10th IEEE International Conference on Cloud Computing (CLOUD), Honolulu, USA, 2017: 798–801. doi: 10.1109/CLOUD.2017.118.

- [5] XIA Jing, CAI Zhiping, and XU Ming. Optimized virtual network functions migration for NFV[C]. The 22nd IEEE International Conference on Parallel and Distributed Systems (ICPADS), Wuhan, China, 2016: 340–246. doi: [10.1109/ICPADS.2016.0053](https://doi.org/10.1109/ICPADS.2016.0053).
- [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](https://doi.org/10.1109/NFV-SDN.2018.8725645).
- [7] 唐伦, 赵培培, 赵国繁, 等. 基于QoS保障的服务功能链动态部署算法[J]. 北京邮电大学学报, 2018, 41(6): 90–96. doi: [10.13190/j.jbupt.2018-013](https://doi.org/10.13190/j.jbupt.2018-013).
TANG Lun, ZHAO Peipei, ZHAO Guofan, *et al.* Dynamic deployment algorithm for service function chaining with QoS guarantee[J]. *Journal of Beijing University of Posts and Telecommunications*, 2018, 41(6): 90–96. doi: [10.13190/j.jbupt.2018-013](https://doi.org/10.13190/j.jbupt.2018-013).
- [8] ERAMO V, AMMAR M, and LAVACCA F G. Migration energy aware reconfigurations of virtual network function instances in NFV architectures[J]. *IEEE Access*, 2017, 5: 4927–4938. doi: [10.1109/ACCESS.2017.2685437](https://doi.org/10.1109/ACCESS.2017.2685437).
- [9] LI Han, GAO Hui, LÜ Tiejun, *et al.* Deep q-learning based dynamic resource allocation for self-powered ultra-dense networks[C]. 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, USA, 2018: 1–6. doi: [10.1109/ICCW.2018.8403505](https://doi.org/10.1109/ICCW.2018.8403505).
- [10] YE Junhong and ZHANG Y J A. DRAG: Deep reinforcement learning based base station activation in heterogeneous networks[J]. *IEEE Transactions on Mobile Computing*, 2020, 19(9): 2076–2087. doi: [10.1109/TMC.2019.2922602](https://doi.org/10.1109/TMC.2019.2922602).
- [11] CHU Man, LIAO Xuewen, LI Hang, *et al.* Power control in energy harvesting multiple access system with reinforcement learning[J]. *IEEE Internet of Things Journal*, 2019, 6(5): 9175–9186. doi: [10.1109/JIOT.2019.2928837](https://doi.org/10.1109/JIOT.2019.2928837).
- [12] 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, United Arab Emirates, 2018: 1–6. doi: [10.1109/GLOCOMW.2018.8644157](https://doi.org/10.1109/GLOCOMW.2018.8644157).
- [13] 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).
- 唐 伦：男，1973年生，教授，博士，主要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。
- 陈前斌：男，1967年生，教授，博士生导师，主要研究方向为个人通信、多媒体信息处理与传输、异构蜂窝网络等。

责任编辑：陈 倩