基于多智能体柔性演员-评论家学习的服务功能链部署算法

唐 伦 李师锐* 杜雨聪 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：针对网络功能虚拟化(NFV)架构下业务请求动态变化引起的服务功能链(SFC)部署优化问题，该文提出一种基于多智能体柔性演员-评论家(MASAC)学习的SFC部署优化算法。首先，建立资源负载惩罚、SFC部署成本和时延成本最小化的模型，同时受限于SFC端到端时延和网络资源预留阈值约束。其次，将随机优化问题转化为马尔可夫决策过程(MDP)，实现SFC动态部署和资源的均衡调度，还进一步提出基于业务分工的多决策者编排方案。最后，在分布式多智能体系统中采用柔性演员-评论家(SAC)算法以增强探索能力，并引入了中央注意力机制和优势函数，能够动态和有选择性地关注获取更大部署回报的信息。仿真结果表明，所提算法可以实现负载惩罚、时延和部署成本的优化，并随业务请求量的增加能更好地扩展。

关键词：网络功能虚拟化；服务功能链；柔性演员-评论家学习；多智能体强化学习
中图分类号：TN929.5 **文献标识码**：A **文章编号**：1009-5896(2023)08-2893-09
DOI: 10.11999/JEIT220803

Deployment Algorithm of Service Function Chain Based on Multi-Agent Soft Actor-Critic Learning

TANG Lun LI Shirui DU Yucong CHEN Qianbin
(*School of Communication and Information Engineering*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)
(*Key Laboratory of Mobile Communication Technology*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

Abstract: Considering the problem of Service Function Chain (SFC) deployment optimization caused by the dynamic change of service requests under the Network Function Virtualization (NFV) architecture, an SFC deployment optimization algorithm based on Multi-Agent Soft Actor-Critic (MASAC) learning is proposed. Firstly, the model of minimizing resource load penalty, SFC deployment cost and delay cost is established, which is constrained by SFC end-to-end delay and reservation threshold of network resource. Secondly, the stochastic optimization is transformed into a Markov Decision Process (MDP) to realize the dynamic deployment of SFC and the balanced scheduling of resources. The arrangement scheme according to services division for multiple decision makers is further proposed. At last, the Soft Actor-Critic (SAC) algorithm is adopted in distributed multi-agent system to enhance exploration, then the central attention mechanism and advantage function are further introduced, which can dynamically and selectively focus on the information to obtain greater deployment return. Simulation results show that the proposed algorithm can optimize the load penalty, delay and deployment cost, and scale better with the increase of service requests.

Key words: Network Function Virtualization (NFV); Service Function Chain (SFC); Soft Actor-Critic (SAC) learning; Multi-agent reinforcement learning

收稿日期：2022-06-17；改回日期：2022-10-13；网络出版：2022-12-23
*通信作者： 李师锐 2819717062@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)，四川省科技计划项目(2021YFQ0053)
Foundation Items: The National Natural Science Foundation of China (62071078), The Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-M201800601), Sichuan Science and Technology Program (2021YFQ0053)

1 引言

未来网络业务种类繁多且用户需求激增，网络“一刀切”方法不再有效可行[1]，因此网络切片技术受到业界格外关注。在网络功能虚拟化(Network Function Virtualization, NFV)架构中，虚拟化网络功能(Virtual Network Function, VNF)表示网络功能的软件实例化，这些功能与所使用的硬件资源分离[2]。服务功能链(Service Function Chain, SFC)是若干个有序的VNF串联形成的服务请求[3]，SFC部署阶段要求在底层网络上进行VNF放置和实例化，同时伴随着资源分配和路由等问题，从而满足特定的网络服务需求。

如何设计高效的部署方案是SFC编排的关键挑战，文献[4]将SFC部署问题转化为混合整数非线性规划，以在满足时延要求下最小化部署成本，但是没有考虑资源负载的问题。文献[5]采用离散粒子群优化算法求解部署策略，使得在最大化可靠性的同时尽可能地降低网络负载，但是没有对端到端时延进行联合优化。文献[6]联合考虑了SFC部署和调度问题，如果违反最大允许调度时间，则重新将VNF放置到合适的位置，但是没有进一步考虑链路通信所导致的延迟。

传统启发式方法依赖于人工嵌入规则，不能很好地适应动态网络结构和环境。文献[7]通过强化学习来探索NFV基础设施以学习布局决策，但在实现总体功耗成本最小化时没有进行时延度量。文献[8]采用集中式的自适应在线编排方法，目标是在满足服务质量约束的同时最大化体验质量，然而该编排方案不适应未来网络的分布式需求。文献[9]采用了分布式的联邦学习方法，增强对SFC部署决策的隐私保护，但需要交换大量模型参数从而导致昂贵的通信开销。

针对上述问题，本文提出了一种基于多智能体强化学习的SFC部署优化方案，主要贡献包括：(1)设计基于节点容量比例的超载惩罚机制，对占用资源超过阈值的部分施加适当的惩罚，实现网络资源的均衡分配，建立了网络超载惩罚、端到端时延和部署成本最小化的模型；(2)将随机优化问题转化为马尔可夫决策过程(Markov Decision Process, MDP)模型，基于VNF映射和节点计算资源与链路带宽资源的联合分配，实现资源动态分配下的SFC部署，进一步地，提出了多决策者按业务划分的编排方案以扩展到多智能体系统中；(3)为了增强智能体探索的能力和鲁棒性，采用支持随机策略的最大熵学习目标，还在智能体协作时引入中央的注意力机制以关注有效信息，并结合优势函数来实现智能体之间的信用分配，有效解决了智能体数量增加时面临的扩展性问题。

2 系统模型

2.1 网络场景

网络场景基于NFV编排与控制的架构[10]，如图1所示，主要分为物理层、虚拟化层和应用层。物理层是包含通用服务器的底层承载网络，为VNF提供其实例化的物理资源。虚拟化层主要完成网络状态的实时监控、物理网络的负载分析和资源分配策略的执行。应用层主要负责根据业务需求创建SFC，以SFC为载体来为用户提供各种服务。

2.2 网络模型

2.2.1 物理网络

物理网络包括大量的节点和链路，被建模为一个无向图$G^{\mathrm{p}}=(N,L)$。N表示物理节点即服务器的集合，L表示连接各节点的链路集合。服务器为VNF提供其实例化的CPU资源，且每台底层服务器可以实例化多个VNF。每个服务器包含多个CPU，C_v^{p}表示第v个服务器所拥有的CPU资源容量。假设参数$u\in N$和$v\in N$表示两个相邻物理服务器，uv表示连接u和v的物理链路，其有限带宽资源表示为B_{uv}^{p}。考虑到能耗问题以及管理服务器的方便性，需对服务器设置使用门槛，a_v^{p}表示第v个服务器资源分配量阈值，若要将该服务器参与网络资源分配，则它分配的CPU资源量不得低于a_v^p。

2.2.2 SFC映射

在NFV基础架构中，将虚拟网络建模为一个有向图$G^v=(V,P)$。网络中SFC的集合表示为F，第i条SFC表示为有向图$G_i^v=(V_i,P_i)$，V_i表示第i条SFC上不同VNF的集合，P_i表示第i条SFC上虚拟链路的集合。对于第i条SFC上的第j个VNF，$C_{i,j}^v$表示服务器v分配给它的CPU资源量。jk表示连接第i条SFC上的相邻第j个和第k个VNF的链路，$B_{i,jk}^{uv}$表示物理链路uv分配给它的带宽资源量。定义布尔变量$\delta_{i,j}^v$=0，1，当第i条SFC上的第j个VNF映射到物理服务器v上时，$\delta_{i,j}^v=1$，否则$\delta_{i,j}^v=0$。定义布尔变量$\theta_{i,jk}^{uv}$=0，1，当第i条SFC的虚拟链路jk映射到物理链路uv上时，$\theta_{i,jk}^{uv}=1$，否则$\theta_{i,jk}^{uv}=0$。

2.2.3 SFC时延模型

端到端时延表示服务请求映射到底层网络处理时，数据流按服务路径从源节点到目的节点所耗费的时间。对于第i条SFC的数据包到达过程，假设$\omega_i(t)$表示第i条SFC实际到达数据包的个数，服从参数为λ_i的泊松分布。设第i条SFC的端到端最大容忍时延为τ_i。

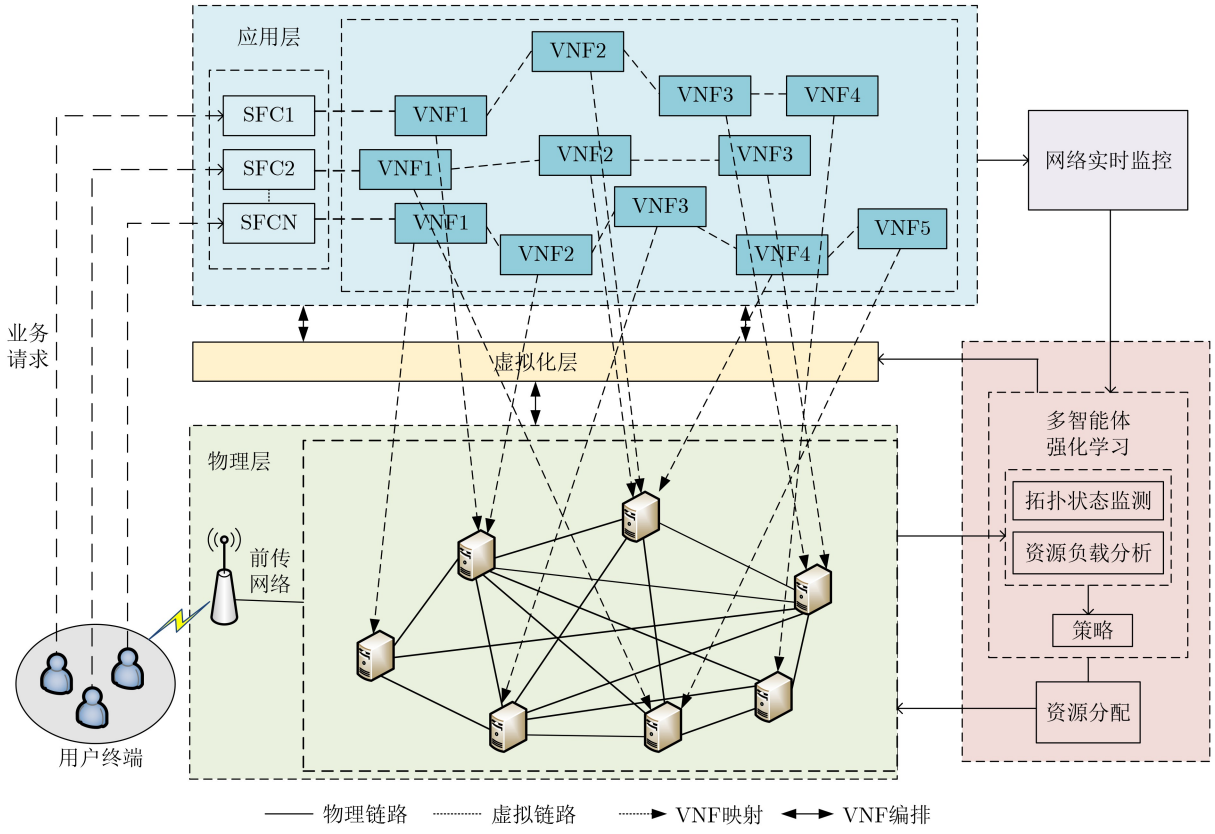

图 1 系统架构

单节点处理时延除了与分配到的计算资源量大小有关，还与各业务不同的待处理数据量有关[11]，用P_i表示第i条SFC的VNF处理总时延，应结合VNF部署情况来分析，在t时隙下表示为

$$P_i(t)=\sum_{j\in V_i}\sum_{v\in N}\delta_{i,j}^{v}(t)\frac{\omega_i(t)m_i}{C_{i,j}^{v}(t)\beta},\forall i\in F \tag{1}$$

其中，m_i表示第i条SFC上数据包的大小，服务速率系数β表示单个CPU每秒能处理的数据量[12]。

T_i表示第i条SFC的链路通信总时延，与链路映射情况有关，在t时隙下表示为

$$T_i(t)=\sum_{jk\in P_i}\sum_{uv\in L}\theta_{i,jk}^{uv}(t)\left[\frac{\omega_i(t)m_i}{B_{i,jk}^{uv}(t)}+\psi\right],\forall i\in F \tag{2}$$

其中，ψ表示数据包排队调度导致的延迟。

2.2.4 SFC部署成本

SFC部署产生的成本包括两部分，一部分来自底层网络服务器对VNF的处理，另一部分来自链路带宽的使用[13]。处理成本$Z_{i,j}^{v}$包括动态和静态两方面，动态成本描述为CPU运行产生的可变成本，与分配资源量有关，系数为正数ε，而静态成本是任意服务器对VNF进行激活的成本，用正数∂表示，因此$Z_{i,j}^{v}$可表示$Z_{i,j}^{v}(t)=\partial+\varepsilon C_{i,j}^{v}(t)$。物理链路带宽使用成本$Z_{i,jk}^{uv}$与占用的物理链路带宽量成正比，单位带宽开销系数用正数ς表示，则$Z_{i,jk}^{uv}$表示为$Z_{i,jk}^{uv}(t)=\varsigma B_{i,jk}^{uv}(t)$。则第$i$条SFC的部署总成本$Z_i$在$t$时隙下可表示为

$$\begin{aligned}Z_i(t)=&\sum_{j\in V_i}\sum_{v\in N}\delta_{i,j}^{v}(t)Z_{i,j}^{v}(t)\\&+\sum_{jk\in P_i}\sum_{uv\in L}\theta_{i,jk}^{uv}(t)Z_{i,jk}^{uv}(t)\end{aligned} \tag{3}$$

2.2.5 网络超载惩罚

为降低网络负载和提高资源分配的均匀程度，以便进行后续的SFC部署以及应对VNF迁移等突发状况，需要对资源分配不当导致负载过大的节点进行惩罚，考虑到网络中每个节点的资源容量不同，以资源剩余绝对量为基准是不合理的，因此对剩余资源与各节点容量的相对比例进行分析。

假设$\eta_c^v(t)$表示在t时隙第v个节点的CPU资源预留率，其计算公式为

$$\eta_c^v(t)=\frac{C_v^{\mathrm{p}}-a_v(t)}{C_v^{\mathrm{p}}},\forall v\in N \tag{4}$$

其中，$a_v(t)$表示在t时隙第v个服务器已分配的CPU资源，可表示为

$$a_v(t)=\sum_{i\in F}\sum_{j\in V_i}\delta_{i,j}^{v}(t)C_{i,j}^{v}(t),\forall v\in N \tag{5}$$

令α_c表示底层服务器的资源超载警戒值，对资源预留率不足该门限的部分给予惩罚，与警戒值相差越多则应受惩罚越多，网络中服务器v的超载惩罚E_c^v在t时隙下可表示为

$$E_c^v(t)=\begin{cases}\varepsilon_c[\alpha_c-\eta_c^v(t)]^2 & \eta_c^v(t)<\alpha_c\\ 0 & \eta_c^v(t)\geq\alpha_c\end{cases} \tag{6}$$

其中，ε_c表示资源预留率不足部分所受到的单位惩罚。设置服务器CPU资源预留率最小阈值α_{cm}，各底层节点的$\eta_c^v(t)$不得小于α_{cm}，当α_{cm}为0时表示不考虑未来情况而允许对CPU资源进行不保留分配，此时等价于分配的最大资源量受容量限制。

2.3 优化目标

本文系统的SFC部署问题可表述为：如何进行VNF的映射以及CPU与带宽的联合资源分配，使得在最小化部署成本与网络超载惩罚的同时，尽可能地降低SFC端到端时延。为了统一时延、成本和惩罚的单位，现对各部分进行归一化处理，设计效用函数为

$$\begin{aligned}U(t)=&-\sigma_1\frac{\sum\limits_{i\in F}[P_i(t)+T_i(t)]}{\sum\limits_{i\in F}\tau_i}-\sigma_2\frac{\sum\limits_{i\in F}Z_i(t)}{Z_{\max}}\\&-\sigma_3\frac{\sum\limits_{v\in N}E_c^v(t)}{E_{\max}}\end{aligned} \tag{7}$$

其中，$Z_{\max}$表示网络的最大部署成本，$E_{\max}$表示因资源超载受到的最大惩罚，σ_1, σ_2和σ_3表示各项重要程度，它们为3个大于零的权重值，且$\sigma_1+\sigma_2+\sigma_3=1$。本文优化目标的数学模型表示为

$$\left.\begin{aligned}&\max_{\delta_{i,j}^v(t),C_{i,j}^v(t),B_{i,jk}^{uv}(t)}U(t)\\&\text{s.t.C1}:\delta_{i,j}^v(t)=\{0,1\},\forall i\in F,\forall j\in V_i,\forall v\in N\\&\quad\text{C2}:\sum_{v\in N}\delta_{i,j}^v=1,\forall i\in F,\forall j\in V_i\\&\quad\text{C3}:\delta_{i,j}^v(t)C_{i,j}^v(t)=C_{i,j}^v(t),\\&\qquad\forall i\in F,\forall j\in V_i,\forall v\in N\\&\quad\text{C4}:\theta_{i,jk}^{uv}(t)=\{0,1\},\forall i\in F,\forall jk\in P_i,\forall uv\in L\\&\quad\text{C5}:\theta_{i,jk}^{uv}(t)B_{i,jk}^{uv}(t)=B_{i,jk}^{uv}(t),\forall i\in F,\\&\qquad\forall jk\in P_i,\forall uv\in L\\&\quad\text{C6}:\eta_c^v(t)\geq\alpha_{cm}\geq 0,\forall v\in N\\&\quad\text{C7}:\sum_{i\in F}\sum_{jk\in P_i}\theta_{i,jk}^{uv}(t)B_{i,jk}^{uv}\leq B_{uv}^p,\ \forall uv\in L\\&\quad\text{C8}:\text{若}\sum_{i\in F}\sum_{j\in V_i}\delta_{i,j}^v(t)>0,\text{则}a_v(t)\geq a_v^p,\forall v\in N\\&\quad\text{C9}:D_i(t)\leq\tau_i,\forall i\in F\end{aligned}\right\} \tag{8}$$

C1表示VNF的映射需满足二进制约束。C2保证了网络中任意VNF只能选择一个服务器进行映射。C3保证了每台服务器只会对映射到它的VNF分配CPU资源。C4表示虚拟链路的映射需满足二进制约束。C5保证了每条物理链路只会对映射到它的虚拟链路分配带宽资源。C6保证了服务器可分配的CPU资源受容量限制，且对资源的预留应满足前瞻性需求。C7保证了物理链路可分配的带宽资源受容量限制。C8保证了各服务器分配CPU资源时满足一定门槛。C9保证了各条SFC在任何时隙都需满足时延要求。

将本文建立的SFC部署问题，转化为一个MDP模型，用一个4元组$M=<S,A,P,R>$来表示，其中S为状态空间，A为动作空间，P为状态转移概率，R为奖励函数。

定义状态空间包括各SFC映射状态K_i(t)、各节点剩余计算资源率$\eta_c^v(t)$和各物理链路剩余带宽资源比例$\eta_b^{uv}(t)$，$s(t)\in S$表示为$s(t)=\{\boldsymbol{K}(t),\eta_c(t),\eta_b(t)\}$，其中$\boldsymbol{K}(t)=[K_i(t)]$，$\eta_c(t)=[\eta_c^v(t)]$，$\eta_b(t)=[\eta_b^{uv}(t)]$。

定义动作空间包括各链所有VNF的映射、节点CPU资源分配和链路带宽资源分配，因此$a(t)\in A$表示为$a(t)=\{\boldsymbol{\delta}(t),\boldsymbol{C}(t),\boldsymbol{B}(t)\}$，其中$\boldsymbol{\delta}(t)=[\delta_{i,j}^v(t)]$，$\boldsymbol{C}(t)=[C_{i,j}^v(t)]$，$\boldsymbol{B}(t)=[B_{i,jk}^{uv}(t)]$。

网络在t时隙状态$s(t)$下，采取行动$a(t)$后，会转移到下一时隙的状态$s(t+1)$，定义这一过程的状态转移概率为$p(s(t+1)|s(t),a(t))$。

本文目标是最小化SFC端到端平均时延、网络部署成本和超载惩罚，则可以定义奖励函数为$R(t)=-[U(t)]^{-1}$。

3 基于MASAC学习的SFC部署算法

强化学习通过不断试错与给予奖励来指导智能体探寻学习策略，相比启发式算法能更有效解决SFC部署问题，但基于传统强化学习的SFC部署方案仍存在不足，首先，若对整体建模然后采用集中式的单智能体强化学习，存在着动作空间庞大的问题，并且会产生较大的信令开销。其次，若分别采用完全独立式的强化学习，在对某个体进行探讨时，其他智能体的策略随时可能发生调整，因此会面临着环境不平稳的问题。为了解决以上问题，本文采用多智能体强化学习的方法，结合支持随机策略的最大熵来增强探索性，同时为了进一步增强扩展能力，还在各智能体之间交互时引入了中央注意力机制，通过动作边缘化的优势函数来分配信用。

在本文提出的多智能体系统中，将有各种业务需求的用户视为不同的智能体，并对其编号为

$i \in \{1,2,\cdots,N\}$，各智能体基于独立于其他智能体的局部观测$o_i(t)=\{K_i(t),\eta_{ci}(t),\eta_{bi}(t)\}$，各自采取行动$a_i(t)=\{\boldsymbol{\delta}_i(t),\boldsymbol{C}_i(t),\boldsymbol{B}_i(t)\}$，并获得私有奖励$r_i(t)$来与环境不断交互，每个智能体学习策略$\pi_i: O_i \to P(A_i)$。各个智能体互相合作去服务到达的请求，如图2所描述的多决策者场景，网络中各服务器的资源容量不相同，容量大小用长方形总面积表示，各业务请求用不同SFC来表示，为展示网络资源负载与每个智能体占用资源的情况，设计了象征不同智能体的各类形状的线条，其中，黑框矩形表示网络资源未使用的预留部分，此外，各服务器对智能体分配的CPU资源量用其所占空间的面积大小表示，物理链路的带宽资源分配量用各类线条的长度表示。每个智能体都可以访问环境中的所有资源，并选择适当的网络资源来满足各自业务需求，它们的共同目标是获得最大的累积共享奖励。

为了让智能体采取能获得高回报且随机性高的策略，本文考虑将柔性演员-评论家(Soft Actor-Critic, SAC)作为基准算法，由于其基于最大熵强化学习框架，因此具有较强的探索性和鲁棒性。对于支持随机策略的最大熵学习，其强化目标表示为

$$J(\pi)=\sum_{t=0}^{T} E_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t)+\alpha H(\pi(\cdot|s_t))] \tag{9}$$

其中，温度参数α在奖励大小和熵二者之间起到平衡的作用，当α为0时目标等价为标准的强化学习目标。对于一个固定的策略，为使软Q价值函数能够被迭代计算，需要重复应用修正过的贝尔曼算子Γ^π，即

$$\Gamma^\pi Q(s_t,a_t) \triangleq r(s_t,a_t)+\gamma E_{s_{t+1}\sim p}[V(s_{t+1})] \tag{10}$$

其中，$V(s_t)$表示柔性状态价值函数，即

$$V(s_t)=E_{a_t\sim\pi}[Q(s_t,a_t)-\ln\pi(a_t|s_t)] \tag{11}$$

为充分对环境进行探索，柔性演员评价算法通过修改策略梯度以加入熵项，即

$$\begin{aligned}\nabla_\theta J(\pi_\theta)=&E_{s\sim D,a\sim\pi}[\nabla_\theta \ln(\pi_\theta(a|s))\\&(-\alpha\ln(\pi_\theta(a|s))+Q_\psi(s,a)-b(s))]\end{aligned} \tag{12}$$

MASAC遵循集中式训练评论家与分布式执行策略的范式，为了让特定智能体能选择性地关注来自其他智能体的信息，还需引入带多个注意力头部的中央注意力机制，分布式的actor网络采取动作并获取对应的观测，通过本地信息映射后再进行共享式的训练。为了将关注于不同子空间的信息联合，应将所有头部的贡献连接起来，可将其表示为$\mathrm{Mhead}=\mathrm{Concat}(\mathrm{head}_1,\mathrm{head}_2,\cdots,\mathrm{head}_h)\boldsymbol{W}^o$，每个头部使用独立的一组参数$(\boldsymbol{W}_q,\boldsymbol{W}_k,\boldsymbol{V})$，并产生其他智能体对某特定智能体$i$的聚合贡献$x_i$，其中，$\boldsymbol{V}$为合用矩阵。定义$Q_i^\psi(o,a)$为智能体$i$的观察-动作值函数，除自身的观测和动作以外，还与其他智能体的贡献有关，表示为

$$Q_i^\psi(o,a)=f_i(g_i(o_i,a_i),x_i) \tag{13}$$

其中，g_i和f_i都为多层感知机映射函数，(o_i,a_i)表示某个智能体i所采取的VNF映射、CPU与带宽资源分配的动作，以及个体从SFC部署环境中获取到的观察，x_i是去除智能体i后的智能体贡献值，表示为各智能体贡献的加权求和，即

$$x_i=\sum_{j=\backslash i}\alpha_j v_j=\sum_{j=\backslash i}\alpha_j h(Vg_j(o_j,a_j)) \tag{14}$$

其中，$\backslash i$表示除了i以外的智能体集合，v_j表示智能体j提供的价值函数，需要用映射函数对观测和动

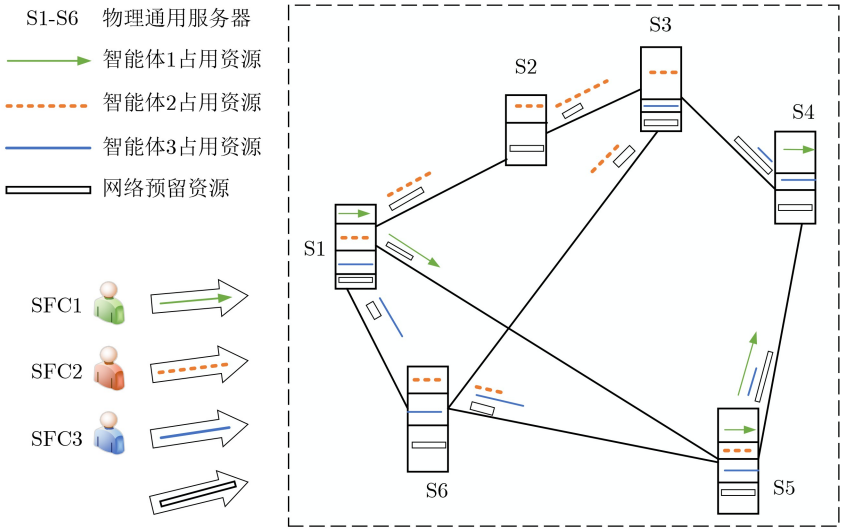

图 2 多智能体业务编排与资源分配示意图

作进行编码，然后通过合用矩阵$\boldsymbol{V}$线性变换得到，h表示非线性的哈达玛积。α_j表示注意力权重值，采用双线性映射(即查询-键值系统)，并将映射值$\boldsymbol{e}_j$和$\boldsymbol{e}_i$之间的相关性值传递到归一化指数函数中，即

$$\alpha_j \propto \exp(\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{W}_q\boldsymbol{e}_i) \tag{15}$$

其中，$\boldsymbol{e}_j$表示为$\boldsymbol{e}_j = g_j(o_j, a_j), j = 1, 2, \cdots, N$，$\boldsymbol{W}_q$和$\boldsymbol{W}_k$是各注意力头部中的参数，分别将映射函数$e_i$和$e_j$转化成查询值和键值。之后将查询值和键值输入到缩放点积模块中，从而对两个矩阵的维度进行尺度变换[14]，最后还通过归一化指数函数模块，以获得各价值的权重。

MASAC中的所有评论家网络一起更新，目标是实现联合回归损失函数的最小化，即

$$L_Q(\psi) = \sum_{i=1}^{N} E_{(o,a,r,o')\sim D}[(Q_i^{\psi}(o,a) - y_i)^2] \tag{16}$$

其中，D是存储以往经验的重放缓存池，Q_i^{ψ}是智能体i动作的价值估计值，需通过注意力机制得到，y_i是目标值，表示为

$$y_i = r_i + \gamma E_{a'\sim\pi_{\overline{\theta}}(o')}[Q_i^{\overline{\psi}}(o', a') - \alpha\ln(\pi_{\overline{\theta}_i}(a_i'|o_i'))] \tag{17}$$

α作为柔性温度参数，能有效衡量SFC部署奖励与最大熵的重要性，$\overline{\psi}$和$\overline{\theta}$分别是目标评论家网络与目标演员网络的参数。目标网络的更新采用软更新的方式，即

$$\overline{\psi} \leftarrow (1-\tau)\overline{\psi} + \tau\psi,\ \overline{\theta} \leftarrow (1-\tau)\overline{\theta} + \tau\theta \tag{18}$$

为了解决智能体间的信用分配问题，考虑在智能体学习中引入优势函数，思想是从观察-动作值函数$Q_i^{\psi}(o,a)$中边缘化所给智能体的动作，然后与原本值大小进行对比，以此了解奖励的增加是否归属于其他智能体。优势函数表示为

$$A_i(o,a) = Q_i^{\psi}(o,a) - b(o, a_{\setminus i}) \tag{19}$$

其中，$b(o, a_{\setminus i})$是多智能体基线，在保持其他智能体动作不变的情况下，将某智能体的特定动作替换为其他可能动作的平均化，即

$$b(o, a_{\setminus i}) = \sum_{a_i'\in A_i} \pi(a_i'|o_i) Q_i(o, (a_i', a_{\setminus i})) \tag{20}$$

各智能体的演员网络策略由梯度上升来更新，梯度计算表达式为

$$\begin{aligned}&\nabla_{\theta_i} J(\pi_\theta) = E_{o\sim D, a\sim\pi}[\nabla_{\theta_i}\ln(\pi_{\theta_i}(a_i|o_i))\\ &(-\alpha\ln(\pi_{\theta_i}(a_i|o_i)) + A_i(o,a))]\end{aligned} \tag{21}$$

本文算法可有效解决SFC的部署优化问题，算法的细节在算法1中。

4 性能仿真与分析

4.1 仿真设置

为了评估模型的有效性和算法的收敛性，本文对所提出算法进行仿真验证，仿真平台基于Python3.7和Pytorch工具实现。本文网络场景为全连接型网络，任意设备的CPU资源和物理链路的带宽容量均随机取值，服务器CPU随机取8, 10, 12核，链路带宽资源为400～800 Mbit/s，每条SFC由3～7个有序VNF构成。本文权重σ_1, σ_2, σ_3分别设置为0.4, 0.3, 0.3。首先探究算法软更新因子与注意力权重的影响，然后在SFC总条数取值为[12,36]的范围下与其他算法进行对比，包括文献[15]中的DDPG方法及其多智能体的版本MADDPG，基于DQN的传统强化学习方法，还将注意力机制的权重固定为1/(N–1)，得到了本文算法的静态注意力简易版本MASAC(U)用于对比。对于训练过程中的每个时间点，将各智能体的数据元组$(o_{1\cdots N}(t), a_{1\cdots N}(t), r_{1\cdots N}(t), o_{1\cdots N}(t+1))$放入重放缓冲区，每次更新对$Q$函数损失目标和策略目标执行梯度下降，都使用Adam作为优化器，学习率为0.001。

4.2 性能分析

首先在包含15个物理节点的网络中处理24条SFC，图3表示了软更新因子τ与收敛性能的关系，更新因子处于[0.001,0.01]内时训练较稳定，并且随着更新因子的降低，收敛速度会变慢，而当更新因子设置为0.1时，奖励曲线带有剧烈的抖动，因为τ过大导致目标网络与先前经验的关联度很低，让训练变得不稳定。

为评估注意力机制动静态对资源分配均匀程度的影响，将本文算法与其注意力权重固定的版本进行比较，以初始值为基准，统计网络中节点资源使用比例的方差百分比，如图4所示，MASAC算法收敛速度比采用MASAC(U)快，且收敛时方差降低了2.8%，即提升了服务器节点分配资源的均衡程度，这是因为智能体之间的合作关系是动态变化的，而本文算法简易版本MASAC(U)固定了注意力的权重值，导致收敛速度和训练效果均不如MASAC，该结果指示了采用动态注意力比静态方案更高效。

为了评估算法随业务需求增加的扩展性能，依次将SFC数量设置为12, 18, 24, 30, 36。首先考察资源分配的均匀程度，以初始值为基准，统计各算法训练后网络节点CPU资源使用率的方差，结果如图5所示，随着SFC数量的增加，各节点资源使用的方差逐渐变小，即资源分配变得更加均匀，原因在于较多业务请求竞争资源时，网络受到节点超载惩罚的可能性增大，从而不倾向于在警戒值内随

算法1 基于多智能体柔性演员-评论家学习的SFC部署算法

输入：多智能体数量N，软更新因子τ，折扣因子γ，温度参数α，注意力头数量h，回放缓存池大小D，回合数M，回合最大长度T

输出：各智能体的策略

(1) 初始化：E个并行的环境，回放缓存池D，$T_{\text{update}} \leftarrow 0$

(2) **for** $i_{ep}= 1, 2, \cdots, M$ episodes **do**

(3)　重置SFC部署的环境，初始化各决策者i的观察o_i^e

(4)　**for**t=$1, 2, \cdots, T$ **do**

(5)　并行环境为决策者i选取动作$a_i^e \sim \pi_i(\cdot|o_i^e)$，进行VNF放置和节点CPU、链路带宽资源分配

(6)　所有决策者获得SFC部署的局部观察o'^e_i，得到VNF放置与资源分配的奖励r_i^e

(7)　**if** C1～C9约束满足，在D中储存各环境的转变

(8)　$T_{\text{update}} = T_{\text{update}} + E$

(9)　**if** T_{update} ≥更新的最小步数，**then**

(10)　**for** $j = 1, 2, \cdots,$ num 评论家网络 更新 **do**

(11)　从缓存池中打包小批次样本B，$(o^B_{1\cdots N}, a^B_{1\cdots N}, r^B_{1\cdots N}, o'^B_{1\cdots N}) \leftarrow B$

(12)　在并行环境中，由式(22)与式(23)计算各决策者的观察-动作值$Q_i^{\psi}(o^B_{1\cdots N}, a^B_{1\cdots N})$，通过目标策略网络计算$a'^B_i \sim \pi_i^{\overline{\theta}}(o'^B_i)$，通过目标评论家网络计算$Q_i^{\overline{\psi}}(o'^B_{1\cdots N}, a'^B_{1\cdots N})$

(13)　由式(25)计算联合损失函数$L_Q(\psi)$，并结合Adam来更新评论家网络

(14)　**end for**

(15)　**for** $j = 1, 2, \cdots,$ num 演员网络 更新 **do**

(16)　采取样本$m \times (o_{1\cdots N}) \sim D$

(17)　计算$a^B_{1\cdots N} \sim \pi_i^{\overline{\theta}}(o'^B_i), i \in 1, 2, \cdots, N$，$Q_i^{\psi}(o^B_{1\cdots N}, a^B_{1\cdots N})$

(18)　由式(28)计算优势函数，再代入式(30)计算$\nabla_{\theta_i} J(\pi_\theta)$，并结合Adam来更新演员网络

(19)　**end for**

(20)　由式(27)，更新目标评论家和演员网络参数：$\overline{\psi} \leftarrow (1-\tau)\overline{\psi} + \tau\psi$，$\overline{\theta} \leftarrow (1-\tau)\overline{\theta} + \tau\theta$

(21)　$T_{\text{update}} \leftarrow 0$

(22)　**end if**

(23)　**end for**

(24) **end for**

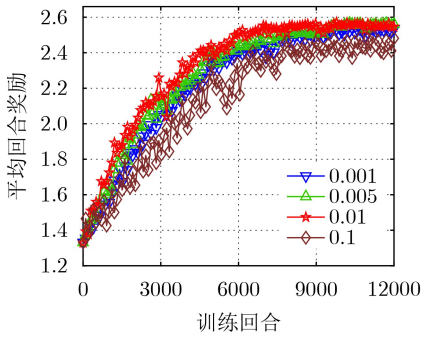

图 3 软更新因子与收敛的关系

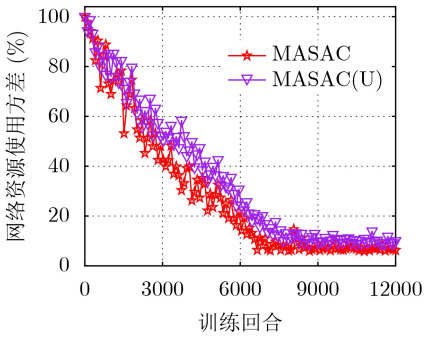

图 4 注意力动静态与资源分配的关系

意地部署，此外，SFC数量为36时本文算法的资源使用方差趋于平稳，同时其他方法的性能出现不同程度的反弹，这是因为底层物理资源有限，当网络中达到一定数目的业务请求后，较大的资源竞争使得资源分配的效果变差。

图6、图7分别为各算法在平均时延和网络惩罚上的对比，随着业务数量的增加，网络中对资源的竞争变大，分配到各业务的资源整体下降，因此端到端的平均时延变高，同时节点能预留的资源减少，导致网络受到的超载惩罚加大，本文算法在优化端到端时延和节点负载上均保持较优水平，尤其当SFC数量为36时，本文算法下的平均时延和网络惩罚相比DQN算法分别降低了13%和24.7%。

为进一步考察资源超载警戒值的大小对网络的

影响，在含15和25节点的两种不同规模网络拓扑下处理36条SFC，结果如图8所示，随着警戒值门限的设定值增大，两种不同规模的网络所受惩罚均加重，这是因为原本资源充足的节点不再容易达到门限要求，而已超载节点的剩余资源率与警戒值的差距越来越大，此外，在同一警戒值的设定下对比两种网络，可以看出小规模网络所受超载惩罚更重，这是因为小规模网络的可分配物理资源少，在处理相同数量的服务请求时，比大规模网络面临的资源竞争更大。

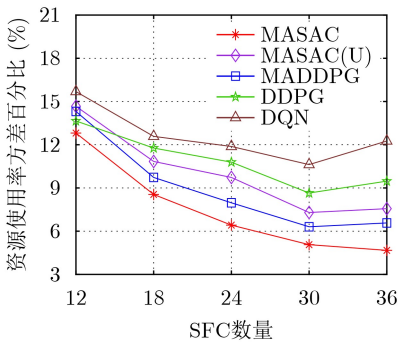

图 5 资源使用方差对比

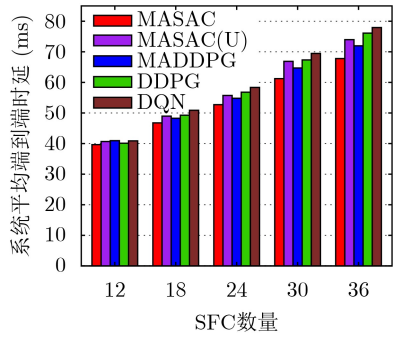

图 6 平均时延对比

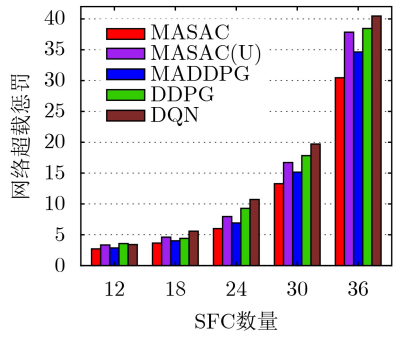

图 7 网络惩罚对比

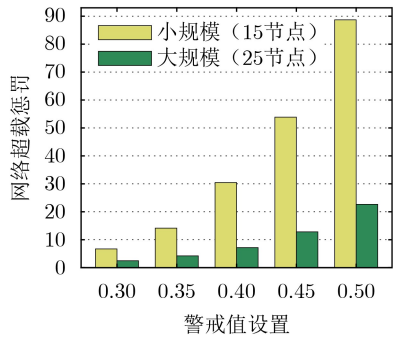

图 8 各警戒值下两种网络超载惩罚

5 结束语

针对网络功能虚拟化下的SFC部署问题，本文建立了带资源预留的优化模型，该模型以最小化时延、部署成本和节点超载惩罚为目标，可以在高效部署SFC的同时平衡网络各节点的负载，为后续部署、迁移等服务做准备。本文以柔性演员-评论家学习为基准，提出了一种基于多智能体强化学习的SFC部署方案，在最大化奖励的强化目标上增加了最大熵项，从而增强探索能力，在智能体学习如何合作时，引入了带多头注意力的中央注意力机制，能够有选择性地关注有利于自己获取更大回报的信息，此外还结合了优势函数来实现智能体间的信用分配。仿真结果表明，本文方案比其他参考策略拥有更高的网络效用和更小的资源使用方差，并且随SFC数量的增加表现出更好的扩展性。

参 考 文 献

[1] CHAHBAR M, DIAZ G, DANDOUSH A, *et al*. A comprehensive survey on the E2E 5G network slicing model[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(1): 49–62. doi: 10.1109/TNSM.2020.3044626.

[2] GONZALEZ A J, NENCIONI G, KAMISINSKI A, *et al*. Dependability of the NFV orchestrator: state of the art and research challenges[J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(4): 3307–3329. doi: 10.1109/COMST.2018.2830648.

[3] SUN Gang, XU Zhu, YU Hongfang, *et al*. Low-latency and resource-efficient service function chaining orchestration in network function virtualization[J]. *IEEE Internet of Things Journal*, 2020, 7(7): 5760–5772. doi: 10.1109/JIOT.2019.2937110.

[4] LI Junling, SHI Weisen, YE Qiang, *et al*. Joint virtual network topology design and embedding for cybertwin-enabled 6G core networks[J]. *IEEE Internet of Things Journal*, 2021, 8(22): 16313–16325. doi: 10.1109/JIOT.2021.3097053.

[5] CHAI Rong, XIE Desheng, LUO Lei, *et al*. Multi-objective optimization-based virtual network embedding algorithm for software-defined networking[J]. *IEEE Transactions on Network and Service Management*, 2020, 17(1): 532–546. doi: 10.1109/TNSM.2019.2953297.

[6] CAO Haotong, DU Jianbo, ZHAO Haitao, *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. doi: 10.1109/TVT.2021.3065967.

[7] SOLOZABAL R, CEBERIO J, SANCHOYERTO A, *et al*. Virtual network function placement optimization with deep reinforcement learning[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(2): 292–303. doi: 10.1109/JSAC.2019.2959183.

[8] CHEN Jing, CHEN Jia, and ZHANG Hongke. DRL-QOR: Deep reinforcement learning-based QoS/QoE-aware adaptive online orchestration in NFV-enabled networks[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(2): 1758–1774. doi: 10.1109/TNSM.2021.3055494.

[9] HUANG Haojun, ZENG Cheng, ZHAO Yangmin, *et al*. Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach[J]. *IEEE Journal on Selected Areas in Communications*, 2021, 39(8): 2558–2571. doi: 10.1109/JSAC.2021.3087227.

[10] 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.

[11] LIU Yu, SHANG Xiaojun, and YANG Yuanyuan. Joint SFC deployment and resource management in heterogeneous edge for latency minimization[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 32(8): 2131–2143. doi: 10.1109/TPDS.2021.3062341.

[12] 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.

[13] PEI Jianing, HONG Peilin, XUE Kaiping, *et al*. Resource aware routing for service function chains in SDN and NFV-enabled network[J]. *IEEE Transactions on Services Computing*, 2021, 14(4): 985–997. doi: 10.1109/TSC.2018.2849712.

[14] VASWANI A, SHAZEER N, PARMAR N, *et al*. Attention is all you need[C]. The 31st International Conference on Neural Information Processing Systems, Long Beach, USA, 2017: 6000–6010.

[15] 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.

唐　伦：男，教授，博士生导师，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.

李师锐：男，硕士生，研究方向为网络功能虚拟化、资源分配和强化学习.

杜雨聪：男，硕士生，研究方向为网络切片、资源分配和机器学习.

陈前斌：男，教授，博士生导师，研究方向为个人通信、多媒体信息处理与传输、下一代移动通信网络、异构蜂窝网络等.

责任编辑：马秀强