计算机科学
COMPUTER SCIENCE

一种面向最佳收益的服务功能链在线编排方法

黄骅, 江俊, 杨永康, 曹斌

引用本文

黄骅, 江俊, 杨永康, 曹斌.一种面向最佳收益的服务功能链在线编排方法[J]. 计算机科学, 2023, 50(6): 66-73.

HUANG Hua, JIANG Jun, YANG Yongkang, CAO Bin. Online Service Function Chain Orchestration Method for Profit Maximization [J]. Computer Science, 2023, 50(6): 66-73.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

融合抗噪和双重蒸馏的文本分类方法

Text Classification Method Based on Anti-noise and Double Distillation Technology

计算机科学, 2023, 50(6): 251-260. https://doi.org/10.11896/jsjkx.220500100

基于双重指针网络的车货匹配双重序列决策研究

Study on Dual Sequence Decision-making for Trucks and Cargo Matching Based on Dual Pointer Network

计算机科学, 2022, 49(11A): 210800257-9. https://doi.org/10.11896/jsjkx.210800257

基于混合软件定义网络的单节点故障保护方法

Single Node Failure Routing Protection Algorithm Based on Hybrid Software Defined Networks

计算机科学, 2022, 49(2): 329-335. https://doi.org/10.11896/jsjkx.210100051

多域SFC部署中基于强化学习的多目标优化方法

Multi-objective Optimization Method Based on Reinforcement Learning in Multi-domain SFC Deployment

计算机科学, 2021, 48(12): 324-330. https://doi.org/10.11896/jsjkx.201100159

基于改进鲸鱼算法的无人机三维路径规划

Three-dimensional Path Planning of UAV Based on Improved Whale Optimization Algorithm

计算机科学, 2021, 48(12): 304-311. https://doi.org/10.11896/jsjkx.201000021

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220400156

一种面向最佳收益的服务功能链在线编排方法

黄 骅[1,3] 江 俊[2] 杨永康[3] 曹 斌[1]
1 浙江工业大学计算机科学与技术、软件学院 杭州 310014
2 浙江树人大学信息科技学院 杭州 310015
3 东方通信股份有限公司博士后科研工作站 杭州 310053
(huanghua@eastcom.com)

摘 要 随着网络功能虚拟化技术的发展,如何对服务功能链进行灵活编排以实现收益最大化已成为服务提供商关注的核心问题。文中以最大化收益为目标,将多数据中心场景下的服务功能链在线编排问题建模为0-1整数规划,并在此基础上提出了一种两阶段启发式算法。在第一阶段,根据负载情况及部署开销计算节点和链路的权重值,将服务功能链部署在优先级最高的节点上,然后根据链路的负载情况选取满足带宽约束且优先级最高的链路。在第二阶段,类比最长有效功能序列方法,提出了一种虚拟服务迁移策略,以降低部署资源消耗。基于NSFNET和USNET网络拓扑设计了仿真实验,实验结果表明,相比现有算法,所提方法在部署收益和部署成功率两个方面均有一定提升,能够实现服务资源的优化配置,有效提升部署收益。

关键词:服务功能链;虚拟网络功能;功能复用;启发式算法;组合优化
中图法分类号 TP393.01

Online Service Function Chain Orchestration Method for Profit Maximization

HUANG Hua[1,3], JIANG Jun[2], YANG Yongkang[3] and CAO Bin[1]
1 College of Computer Science and Software, Zhejiang University of Technology, Hangzhou 310014, China
2 College of Information Science and Electrical Engineering, Zhejiang Shuren University, Hangzhou 310015, China
3 Central Research Institute of Eastern Communications Co., Ltd., Hangzhou 310053, China

Abstract With the development of network function virtualization technology, how to deploy service function chain flexibly to maximize the profit has become one of the major challenging issues for network service providers. In this paper, we formulate the service function orchestration problem for multi-data center as 0-1 integer programming with the aim to maximize the profit, and propose a two-stage heuristic algorithm to solve this problem. In the first stage, the weights of nodes and links are calculated according to the load condition and deployment cost, the service function chain is deployed on the node with the highest priority, then the link with the highest priority that meets the bandwidth constraint is selected according to the load condition. In the second stage, by analogy with the longest effective function sequence method, a virtualized network function migration strategy is proposed to reduce the consumption of deployment resources. Simulation experiment is designed based on NSFNET and USNET network topology. Experimental results show that, compared with existing algorithms, the proposed method has a certain improvement in both total profit and deployment success rate.

Keywords Service function chain, Virtualized network function, Function reuse, Heuristic algorithm, Combinatorial optimization

1 引言

网络功能虚拟化[1-2](Network Function Virtualization, NFV)是实现5G服务化架构的关键技术之一。NFV将现有网络中的各类网元软件化为虚拟网络功能(Virtualized Network Function, VNF),使网络摆脱对传统硬件设备的依赖,便于网络设备服务的更新升级,降低了运营成本,提升了业务部署的弹性。与此同时,随着NFV应用研究的不断深入,一系列关键问题被提出并成为学术界和工业界关注的热点。

在基于NFV的网络中,其服务能力主要由虚拟网络功能[3](Virtualized Network Function, VNF)承载。VNF是软件化的网元,能够提供特定的服务(如防火墙、入侵检测、

到稿日期:2022-04-14 返修日期:2022-09-30
基金项目:浙江省重点研发计划(2022C01145);浙江省教育厅科研项目(Y202146607)
This work was supported by the Key Research and Development Program of Zhejiang Province(2022C01145) and Scientific Research Foundation of the Education Department of Zhejiang Province(Y202146607).
通信作者:江俊(heifetz@zju.edu.cn)

定位、计费等），通常以软件形式部署于商用服务器上。服务提供商根据业务需求，将 VNF 进行有序组合，以此形成的包含起点和终点的业务链也被称为服务功能链[4-5]（Service Function Chain，SFC）。图 1 给出了由 5 个 VNF 组成的服务 SFC，数据流从起始点开始依次经过防火墙、包过滤器、加密、网络监视和解密，最后到达终点。与之相对的，在网络中部署 VNF，并引导业务流依次被 VNF 处理的过程，被称为服务功能链编排[6]（Service Function Chain Orchestration，SFCO）。服务功能链编排主要由服务编排器完成，即针对多样化的业务场景和需求实现面向服务化的高效 VNF 组合，生成满足经济效益和性能指标要求的 SFC。

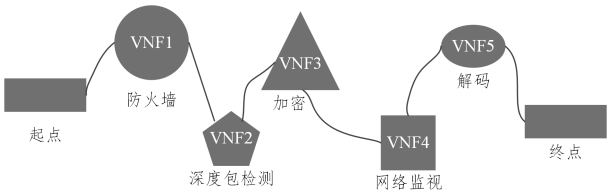

图 1　服务功能链示例

Fig. 1　Example of service function chain

为了向客户提供多样化的服务，网络服务提供商需要部署大量功能不同的 SFC。另一方面，网络中的服务器资源和链路资源有限，因此在编排过程中需要兼顾部署收益与资源消耗。如何在资源有限的网络中为 SFC 选择合适的功能部署方式和数据传输路径，实现收益最大化，是网络服务提供商关注的首要问题。

近年来，许多研究致力于解决多数据中心环境下的 SFC 部署问题，目标包括最优化时延、部署资源耗费、可靠性、负载均衡度及部署收益等。文献[7]提出了一种基于广度优先搜索的启发式方法，在邻近节点列表中优先选取距离起始点跳数较小且可用带宽较大的节点作为目标点，以实现端到端时延最小化。文献[8]以最小化部署资源耗费及链路平衡为目标，建立了多目标优化模型，采用基于博弈论的方法优化资源耗费及传输时延。文献[9]将部署耗费分为信道占用和实例化 VNF 两部分，并建立了深度学习模型，用于预测未来请求量的变化；在此基础上，设计了两阶段启发式算法，以实现收益最大化。文献[10] 通过选择可聚合物理节点集合来提高 SFC 的可靠性并降低端到端传输时延。文献[11] 设计了启发式算法，用于解决 SFC 部署中局部资源负载不均衡的问题。文献[12]利用强化学习算法解决多域网络中的 SFC 编排问题，对传输时延、网络负载均衡性及接受率进行了同步优化。

在优化部署收益方面，文献[13]将端到端时延约束下的分布式云数据中心 SFC 编排建模为 ILP 优化问题，并提出了 3 种在线启发式算法，目标是在最小化部署开销的前提下尽可能提升求解速度。文献[14]考虑了更一般的 SFC 拓扑，而不是单纯的链式结构，目标是实现更高的经济效益。文献[15]研究了时延感知条件下的 SFC 编排问题，包括端到端时延、DC 节点的处理时延及排队时延，以服务提供商收益最大化为目标设计了在线启发式算法。文献[16]提出了一种基于最长公共子序列的启发式算法，通过 VNF 实例的复用来节省部署资源。文献[17]提出了节点部署功能种类受限时最佳路径长度和最佳功能复用度的估计算法，通过启发式算法逐步搜索候选节点逼近最佳路径。文献[18]进一步提出了最长有效功能序列的概念，以功能复用和带宽优化为目标提出了启发式算法。文献[19]针对多数据中心环境下的 SFC 在线部署问题，以最大化收益为目标设计了两阶段启发式算法 ONSO_PKP，在第一阶段预先计算所有 DC 节点之间的 k 最短路径，在第二阶段考虑用户请求接受率和部署耗费，设计了路径和 DC 节点的权重公式，在部署时优先选取权重较高的路径和节点，将 SFC 整体部署在该节点，以实现总收益最大化。ONSO_PKP 算法的局限性在于，其将 SFC 整体部署于单个 DC 节点可能会导致复用度过低，从而增大了部署开销。

基于最长有效功能序列的方法能够实现 VNF 实例的复用，降低资源消耗。现有方法[16-18]均假设 VNF 实例复用不需要消耗额外资源。然而，在实际业务中，VNF 实例多采用容器化部署，实例资源消耗[20-21]包括基础资源消耗和实例资源消耗两部分，其中实例资源消耗会随着接入用户的增多而不断增加。因此，基于最长有效功能序列的方法无法直接应用于该场景。为解决这一问题，类比最长有效功能序列方法，以最大化服务提供商的收益为目标，提出了一种多数据中心服务功能链在线编排方法，该方法能够有效提升部署收益并支持动态在线部署。本文的创新点包括以下几个部分：

(1)将基础资源消耗引入多数据中心 SFC 编排优化场景中，以最大化收益为目标构建了 SFC 编排优化模型。

(2)提出了一种两阶段启发式方法。首先根据链路和节点的负载情况计算权重，优先选取优先级较高的链路和节点进行部署。在此基础上，设计了 VNF 迁移策略对方案进行调优，提高了部署成功率及收益。

(3)以 NSFNET 和 USNET 为场景，设计了仿真实验并与现有方法进行了对比，实验结果验证了本文算法能够有效提升部署收益，具有一定的实用价值。

本文第 1 节介绍了问题背景及相关工作；第 2 节具体描述了多数据中心服务功能链编排问题的优化模型；第 3 节给出了所提方案的具体思想及算法步骤；第 4 节通过仿真实验来评估该方案的具体性能，并与其他类似方案进行比较；最后总结全文并展望未来向。

2　问题描述及系统模型

2.1　问题描述

在 NFV 架构中，服务功能链包括一系列 VNF 以及 VNF 之间的虚拟链路集合，用户流量从起点开始被依次引导经过这些 VNF 的处理后至终点结束。图 2 为服务功能链编排示意图。该服务功能链由 VNF1，VNF2，VNF3 按序组成，Start 和 End 分别为 SFC 的起点和终点。在网络服务请求（Network Service Request，NSR）到达后，编排模块（VNF and Link Orchestration Module）首先根据 NSR 的内容构建端到端的业务逻辑拓扑，然后依次将 VNF 及链路部署至底层网络（Physical Network）的数据中心（DC）节点上。

服务提供商收益最大化是在尽可能提升部署成功率的

前提下减少资源耗费。资源耗费包括计算资源耗费和链路资源耗费。前者由占用 DC 节点的计算资源引起，后者由消耗带宽资源引起。本文的目标是针对多数据中心场景下的服务功能链在线编排问题，设计算法以实现收益最大化。

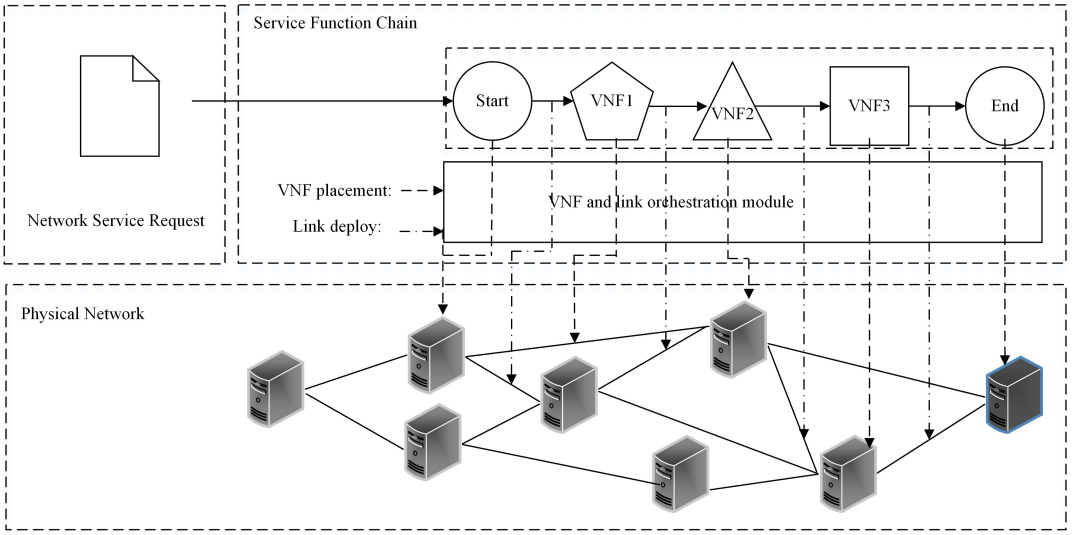

图 2 服务功能链编排示意图

Fig. 2 Sketch map of SFC orchestration

2.2 系统模型

(1)网络模型(Physical Network)。考虑无向加权网络 $G=(N,L)$，其中 N 和 L 分别代表数据中心节点集合与链路集合。令 $c(n)$ 代表编号为 n 的 DC 节点容量，$b(l)$ 代表编号为 l 的链路的容量。每个 DC 节点可以提供资源(如 CPU、内存、存储等)用于部署 VNF，为简化问题描述，将上述资源合并为计算资源。

VNF 实例通常部署在虚拟化容器(如 VM，docker)内，而容器初始化需要消耗一定量的计算资源。这些资源称为基础资源消耗[21](Basic Resource Consumption，BRC)。基础资源消耗为固定值，与容器中部署的 VNF 类型及接入用户数量无关。

(2)网络服务请求(Network Service Request，NSR)。任意 NSR r 可表示为如下四元组 $\langle r_{\text{flow}}, r_{\text{init}}, r_{\text{end}}, S_r \rangle$，其中 r_{flow} 为 r 的带宽需求，r_{init} 和 r_{end} 分别代表 r 的源节点和目标节点，S_r 代表构成该服务功能链的有序 VNF 序列。表 1 列出了 NSR 的一个具体实例，其中起点和终点分别为 DC1 和 DC4，带宽需求为 10Mbps，VNF 序列为 VNF1->VNF2->VNF3。

表 1 网络服务请求示例

Table 1 Example of network service request

Keys	Value
r_{flow}	10 Mbps
r_{init}	DC1
r_{end}	DC4
S_r	VNF1->VNF2->VNF3

(3)资源耗费模型。本文将计算资源消耗分为基础资源消耗和实例资源消耗两部分，其中 BRC 为运行容器占用的资源，与部署在容器中的 VNF 数量与种类无关。实例资源消耗[18]为实例化 VNF 占用的资源，计算方式为单个 VNF 占用的资源乘以接入的用户数量。图 3 给出了某 DC 机房的资源消耗的具体示例。由图 3 可知，该 DC 机房共启用 4 个 VNF 实例。以 Instance1 为例，容器占用的计算资源为 30 单位，VNF1 实例占用的资源为单个实例占用的资源乘以接入用户数量，合计为 30 单位。因此，Instance1 占用的计算资源总量为 60 单位。

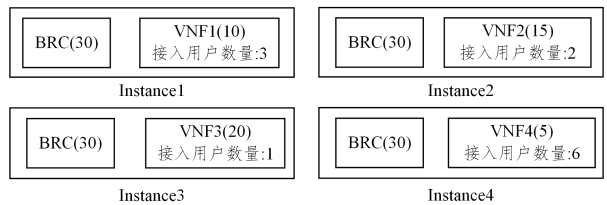

图 3 服务功能链编排资源消耗

Fig. 3 Resource consumption of SFC orchestration

2.3 约束条件

本节进一步分析了系统约束条件，所用符号如表 2 所列。首先，VNF 和虚拟链路分别只能部署至唯一 DC 节点和物理链路，因此以下约束成立。

$$\sum_{n=1}^{|N|} x_n^{r,j} = 1, \forall r \in R, \forall j \in |r| \tag{1}$$

当 VNF 部署时，基础资源消耗与实例资源消耗之和不能超过该节点所能提供的计算资源总量，因此有：

$$\sum_{r=1}^{R} \sum_{j=1}^{|r|} x_n^{r,j} \mu(s_j^r) + brc \cdot \sum_{f=1}^{|F|} z_n^f \leqslant c(n), \forall n \in N \tag{2}$$

$$z_n^f = \begin{cases} 1, & \text{if } \sum_{r=1}^{R} \sum_{j=1}^{|r|} x_n^{r,j} \delta_f^{r,j} \geqslant 1, \forall f \in F \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

其中，$\delta_f^{r,j}$ 代表 s_j^r 是否为类型 f 的 VNF，若是则为 1，否则为 0。对于给定的 r，s^r 为已知，因此 $\delta_f^{r,j}$ 为常量。为表述简明，令 $init$ 为 0，end 为 $j+1$，则 $l_{\text{init},1}^r$，$l_{|r|,\text{end}}^r$，$y_{m,n}^{r,\text{init},1}$ 和 $y_{m,n}^{r,|r|,\text{end}}$ 可以分别表示为 $l_{0,1}^r$，$l_{|r|,|r|+1}^r$，$y_{m,n}^{r,0,1}$ 及 $y_{m,n}^{r,|r|,|r|+1}$，由此得到链路带宽约束(4)。

$$r_{\text{flow}} \cdot \sum_{r=1}^{R} \sum_{j=0}^{|r|} y_{m,n}^{r,j,j+1} \leqslant b(l), \forall l_{m,n} \in L \tag{4}$$

此外，注意到若 s_j^r 部署于 DC 机房 n，则 r 的数据流必经过 n，因此存在如下约束：

$$\sum_{m=1}^{|N|} y_{m,n}^{r,j,j+1} = \begin{cases} 1, & \text{if } x_n^{r,j}=1 \\ 0, & \text{therwise} \end{cases},$$

$$\forall r \in R, j \in [0,|r|+1], l_{m,n} \in L \tag{5}$$

$$\sum_{m=1}^{|N|} y_{n,m}^{r,j,j+1} = \begin{cases} 1, & \text{if } x_n^{r,j}=1 \\ 0, & \text{otherwise} \end{cases},$$

$$\forall r \in R, j \in [0,|r|+1], l_{m,n} \in L \tag{6}$$

假设 SFC 不能拆分为多条链路同时传输，约束(7)成立 。

$$\sum_{m=1}^{|N|} y_{m,n}^{r,j,j+1} \in \{0,1\}, \forall r \in R, j \in [0,|r|+1], l_{m,n} \in L \tag{7}$$

2.4　优化指标

总收益由服务功能链部署收入和部署开销两部分组成。记总收益为 $Prof$，部署收入为 R，部署开销为 C，则 $Prof = R - C$。其中 $R = \sum_{r}^{|R|} \kappa_r p_r$，$\kappa_r = \begin{cases} 1, & r \text{ is accpeted} \\ 0, & \text{otherwise} \end{cases}$，$p_r$ 为 r 部署成功后获得的收入。部署开销包括节点映射开销和链路映射开销，记节点映射开销为 C_{node}，链路映射开销为 C_{link}，计算式如下：

$$C_{\text{node}} = \sigma \sum_{l=1}^{|L|} \left(\sum_{r=1}^{|R|} \sum_{j=1}^{|r|} x_n^{r,j} \mu(s_j^r) + brc \cdot \sum_{f=1}^{|N|} z_n^f \right) \tag{8}$$

$$C_{\text{link}} = \mu r_{\text{flow}} \sum_{m=1}^{|N|} \sum_{n=1}^{|N|} \sum_{r=1}^{|R|} \sum_{j=0}^{|r|+1} y_{m,n}^{r,j,j+1}, l_{m,n} \in L \tag{9}$$

其中，σ 和 μ 为权重系数，代表 C_{node} 和 C_{link} 所占的比重。综上，面向收益最大化的服务功能链编排问题可以表述为：

$$\begin{aligned} &\max R - (C_{\text{node}} + C_{\text{link}}) \\ &\text{s.t } (1),(2),(3),(4),(5),(6),(7) \end{aligned} \tag{10}$$

表 2　本文中使用的符号

Table 2　Notations used in this paper

Class	Notation	Definition
Physical Network	G	底层网络
	N	DC 机房集合
	L	物理链路集合
	n	编号为 n 的 DC 机房
	$l_{m,n}$	机房 m 和 n 之间的链路
	$b(l)$	链路 l 的可用带宽
	$d(l)$	链路 l 的传输延时
	$c(n)$	DC 机房 n 的资源容量
	BRC	基础资源消耗量
NSR	R	网络服务请求集合
	r	编号为 r 的网络服务请求
	F	VNF 集合
	f	编号为 f 的 VNF 种类
	r_{flow}	r 的带宽需求
	r_{init}	r 的起点
	r_{end}	r 的终点
	s^r	r 的有序 VNF 序列
	p_r	r 的部署收益
	$\|r\|$	r 包含的 VNF 数量
	s_j^r	r 中编号为 j 的 VNF
	$\mu(s_j^r)$	部署 s_j^r 需要消耗的计算资源
	$l_{j,j+1}^r$	s_j^r 和 s_{j+1}^r 之间的虚拟链路
	$l_{\text{init},1}^r$	r_{init} 和 s_0^r 之间的虚拟链路
	$l_{\|r\|,\text{end}}^r$	r_{end} 和 $s_{\|r\|}^r$ 之间的虚拟链路
Variables	$x_n^{r,j}$	若将 s_j^r 部署于 n，则为 1，否则为 0
	z_n^f	若部署于 n 的 VNF 中包含 f，则为 1，否则为 0
	$y_{m,n}^{r,j,j+1}$	若 $l_{j,j+1}^r$ 经过 $l_{m,n}$，则为 1，否则为 0
	$y_{m,n}^{r,\text{init},1}$	若 $l_{\text{init},1}^r$ 经过 $l_{m,n}$，则为 1，否则为 0
	$y_{m,n}^{r,\|r\|,\text{end}}$	若 $l_{\|r\|,\text{end}}^r$ 经过 $l_{m,n}$，则为 1，否则为 0

3　算法设计

相关文献[8]已证明 SFC 编排为 NP-hard 问题。随着问题规模的扩大，如网络拓扑规模扩大或请求数量增多时，求解代价会急剧上升。为快速求解问题(10)，本文提出了一种两阶段启发式方法。首先根据权重对路径和节点进行排序，优先选取优先级较高的路径和节点部署 SFC。并在此基础上，设计了迁移策略，以减小基础资源消耗，提升部署成功率，增加收益。

3.1　节点及路径权重计算

预先计算所有 DC 节点之间的 k 最短路径[22]，为平衡部署成功率与部署成本，参考文献[19]，定义 DC 节点 n 的节点介数如下：

$$bet_n = \sum_{s,t,s \neq t}^{|N|} \sigma_{st}(n) / \sigma_{st} \tag{11}$$

其中，σ_{st} 表示节点 s 和 t 之间的最短路径数量，$\sigma_{st}(n)$ 代表 σ_{st} 中经过了节点 n 的数量。根据定义可知，bet_n 越大，说明有越多最短路径经过 n，即 n 越有可能成为该网络上的一个热点。

节点 n 的权重 $weight_n$ 如下：

$$weight_n = \alpha\left(\left(1 - \frac{c_n}{c_{\max}}\right) + (1 - \mathrm{e}^{-bet_n})\right) + \beta\left(\frac{w_n}{w_{\max}}\right) \tag{12}$$

其中，c_n 代表 n 的剩余可用资源，w_n 代表 n 的单位资源部署开销，$c_{\max}$ 和 $w_{\max}$ 分别代表 c_n 和 w_n 的最大值，参数 α 和 β 为权重。由式(12)可得，在实际部署过程中，算法更倾向于挑选单位部署耗费较小以及剩余资源容量更大的节点进行部署，即节点的 $weight_n$ 值越小，部署优先级越高。

当请求 r 到达时，首先选取 r_{init} 和 r_{end} 之间的 k-最短路径集合作为候选路径集合，随着请求的到达，网络中计算资源与链路资源的使用状况也处于动态变化中，使用路径权重对路径 p 的资源使用情况进行评估，计算式如下：

$$weight_p = \gamma w_p^{\text{util}} + \delta w_p^{\text{cost}} \tag{13}$$

其中，w_p^{util} 用于评价路径 p 的资源利用情况，w_p^{cost} 用于评价路径 p 的部署开销，系数 γ 和 δ 用于衡量两者的权重。w_p^{util} 和 w_p^{cost} 的计算式如下：

$$w_p^{\text{util}} = \left(1 - \frac{\varepsilon_p}{\max\{\varepsilon_p, p \in path_k(s,t)\}}\right) + \max\{u_n, n \in p\} \tag{14}$$

$$w_p^{\text{cost}} = \frac{h_p}{\max\{h_p, p \in path_k(s,t)\}} + \frac{\min\{w_n, n \in p\}}{w_{\max}} \tag{15}$$

其中，ε_p 代表 p 的剩余可用带宽，u_n 代表节点 n 的负载率，h_p 表示路径 p 的跳数。式(14)通过路径上节点的最大负载率和剩余可用带宽来反映路径的资源利用情况，式(15)通过该路径的总跳数和路径上节点的部署开销来反映路径的部署代价，路径的 $weight_p$ 越小，则部署优先级越高。

当请求 r 到达时，首先获取 r_{init} 和 r_{end} 对应的 k 最短路径集合 $path_k(r_{\text{init}}, r_{\text{end}})$，然后使用式(13)－式(15)所示的权重计算式计算 $path_k(r_{\text{init}}, r_{\text{end}})$ 中所有路径的 $weight_p$ 值，并从小到大对其进行排序。对于排序后的每条路径，根据式(11)和式(12)计算路径上所有节点的 $weight_n$ 值，优先选取最小的节点，若该节点满足约束条件(1)－约束条件(7)，则 r 被接受，并在对应的节点处实例化所需资源。上述算法的伪代码

如算法 1 所示。

算法 1 基于权重的网络服务请求部署算法

输入：G，r，F

输出：r 对应的部署方案 solution(r)

1. 初始化：令 solution(r)＝∅，success_flag＝0；
2. 使用权重公式式(13)－式(15)计算 $path_k(r_{init}, r_{end})$ 中所有路径的 $weight_p$ 值，并从小到大对其进行排序
3. 将排序后的路径存入 $path_{k\text{-}sorted}(r_{init}, r_{end})$
4. for all $p \in path_{k\text{-}sorted}(r_{init}, r_{end})$
5. 根据式(11)和式(12)计算 p 上所有节点的 $weight_n$ 值，并从小到大对其进行排序，存入 $Node_{sorted}(p)$
6. for all $n \in Node_{sorted}(p)$
7. if n 满足约束条件(1)—约束条件(7)
8. solution(r)＝n；
9. success_flag＝1；
10. break；
11. end if
12. end for
13. end for
14. if success_flag＝1
15. 接受用户请求 r，部署 SFC
16. 返回部署方案 solution(r)
17. else
18. 部署失败
19. end if

3.2 VNF 迁移策略

算法 1 在分析路径和节点权重的基础上，优先将 SFC 中的所有 VNF 部署在优先级最高的节点，实现整体收益最大化。该算法的局限性在于，当服务请求数量不断增加时，将 SFC 整体部署在单个 DC 节点会导致 BRC 消耗过大，从而降低部署成功率。VNF 复用是解决上述问题的一种可行方法，即将同一类型的 VNF 尽可能部署在一个节点上，以降低 BRC 消耗。受现有工作的启发，本节在算法 1 的基础上提出了一种基于最大复用度的 VNF 迁移策略，伪代码如算法 2 所示。

算法 2 基于最大复用度的 VNF 迁移策略(migrateVNF)

输入：候选部署方案 solution(r)

输出：迁移后的部署方案 solution′(r)

1. 初始化：假设 r 部署在路径 p 中编号为 n 的节点上，令 solution(r)＝$\{r_1(n), r_2(n), \cdots, r_{|r|}(n)\}$，其中括号内为 VNF r_i 对应的部署位置，$\{m_1, m_2, \cdots, m_{|p|}\}$ 为路径 p 上按顺序对应的 DC 节点序列，令 tag＝1，solution′(r)＝solution(r)；
2. for all $i \in 1, \cdots, |r|$
3. if 当 VNF r_i 部署于 n 时接入用户数量为 1
4. for j＝tag，…，|p|
5. if m_j 满足以下条件
6. (i)已部署类型与 r_i 相同的 VNF 实例；
7. (ii)m_j 已占用的资源加上接入 r_i 消耗的资源之和小于 n_j 的最大可用资源总量；
8. (iii)将 r_i 部署于 m_j 后，满足 r 中所有 VNF 的先后顺序；
9. then 将 solution′(r)中对应的 $r_i(n)$ 改为 $r_i(m_j)$，令 tag＝j
10. end if
11. end for
12. end if
13. end for
14. return solution′(r)

migrateVNF 策略中的迁移规则为：按 VNF 的先后顺序，依次将接入数量为 1 的 VNF 实例向路径 p 的其他节点迁移。假设路径 p 上的节点顺序为 $\{m_1, m_2, \cdots, m_{|p|}\}$，则待迁移 VNF 为 r_i，按节点的顺序依次判断 m_j 是否满足步骤 6—步骤 8 中对应的条件(i)、条件(ii)和条件(iii)。条件(i)意味着 m_j 已部署与 r_i 类型相同的 VNF 实例，因此将 r_i 迁移至 m_j 能够减少资源消耗。条件(ii)意味着将 r_i 迁移至 m_j 后不会破坏 2.3 节中的约束条件(2)－约束条件(3)。条件(iii)意味着迁移后满足 VNF 之间的先后顺序。若出现首个满足条件的 m_j，则将 VNF 迁移至该节点并更新 $solution(r)$。然后，令该节点为新的循环起点 tag，如此往复直至完成全部迭代。

结合算法 1 与算法 2，基于最大复用度的网络服务请求部署算法的伪代码如算法 3 所示。

算法 3 基于最大复用度的网络服务请求部署算法

输入：G，r，F

输出：r 对应的部署方案

1. 初始化：令 solution(r)＝∅，solution′(r)＝∅，success_flag＝0，Prof(r)＝∞；
2. 使用权重公式式(13)－式(15)计算 $path_k(r_{init}, r_{end})$ 中所有路径的 $weight_p$ 值并按升序进行排序
3. 将排序后的路径存入 $path_{k\text{-}sorted}(r_{init}, r_{end})$
4. for all $p \in path_{k\text{-}sorted}(r_{init}, r_{end})$
5. if p 满足带宽约束条件(4)
6. 3 根据式(11)和式(12)计算 p 上所有节点的 $weight_n$ 值，并按升序对其进行排序，将结果存入 $Node_{sorted}(p)$
7. for all $n \in Node_{sorted}(p)$
8. 令 solution(r)＝$\{r_1(n), r_2(n), \cdots, r_{|r|}(n)\}$，其中括号内为 VNF r_i 对应的部署位置，$\{m_1, m_2, \cdots, m_{|p|}\}$ 为路径 p 上按顺序对应的 DC 节点序列，令 tag＝1；
9. if solution(r) 满足资源约束条件(2)、约束条件(3)
10. 令 Prof(r)为 solution(r) 对应的部署收益；
11. 调用 migrateVNF 得到迁移后的部署方案 solution′(r)；
12. if solution′(r)对应的收益小于 Prof(r)
13. return solution(r)
14. else
15. return solution′(r)
16. end if
17. else
18. 调用 migrateVNF 得到迁移后的部署方案 solution′(r)；
19. 判断 solution′(r)是否满足资源约束条件(2)和条件(3)，若满足
20. return solution′(r)
21. end if
22. end for
23. end if
24. end for
25. 若没有找到满足条件的部署方案，则部署失败并返回错误

算法 3 对最短路径集合 $path_{k\text{-}sorted}(r_{init}, r_{end})$ 以及最短

路径上的节点集合$Node_{sorted}(p)$进行遍历。若 $solution(r)$满足资源约束条件(2)和(3)，则进一步调用 migrateVNF 得到迁移后的部署方案 $solution'(r)$，并判断两者对应的收益值，选取收益较高的部署方案(即 $solution(r)$与 $solution'(r)$中较大者)。若 $solution(r)$不满足部署条件资源约束条件(2)和(3)，则直接调用 migrateVNF 得到 $solution'(r)$，并判断 $solution'(r)$是否满足约束条件(2)和(3)，若是，则返回 $solution'(r)$，否则进入下一轮迭代。

算法 3 首先基于算法 1 得到候选部署方案 $solution(r)$，在此基础上针对是否满足约束条件分别调用迁移策略 migrateVNF 对候选方案进行优化，减少基础资源损耗，提升部署成功率。

3.3　算法整体框架

考虑 SFC 在线部署场景，假设一系列服务请求按顺序到达，使用队列 arrivedSFC 存储到达的服务请求。根据先入先出规则依次从队列中选择请求，并尝试部署，若部署成功，则分配资源；否则放弃此次部署。综上，本文提出的面向最佳收益的服务功能链在线编排算法如算法 4 所示。

算法 4　面向最佳收益的服务功能链在线编排算法

输入：G，F，R

输出：总收益 Profit

1. 初始化：计算所有的 DC 节点之间的 k-最短路径集合 $path_k$，令 arrivedSFC＝∅，finishedSFC＝∅；
2. while arrivedSFC≠ ∅
3.　按照先入先出规则，选取服务请求 r
4.　利用算法 2 计算部署方案 solution(r)
5.　if 部署成功
6.　　接受用户请求 r，部署 SFC，占用资源
7.　else
8.　　部署失败
9. end while
10. 根据部署方案计算并输出总收益 Profit

3.4　算法时间复杂度分析

本节对算法 4 的时间复杂度进行了分析，参考前文中的定义，假设 NSR 总数为$|R|$，NSR 中包含的平均 VNF 数量为r'，K 为最短路径数量，M 为最短路径中包含的节点数的平均值。根据算法 3，首先计算最短路径和每条路径上所有节点的权重值，这一步骤的时间复杂度为 $O(KM)$。然后使用迁移策略对部署方案进行调整，在最坏情况下，迁移步骤的时间复杂度为 $O(KM^2r')$。上述两个步骤的时间复杂度之和为 $O(KM(1+Mr'))$。因此，算法 4 的整体时间复杂度为 $O(|R|KM(1+Mr'))$。

4　仿真实验

4.1　参数设置

本节通过仿真实验对算法 4 进行评估，对比算法取 ONSO_PKP[19] 以及 OLPM_NDelay[23]。其中 ONSO_PKP 与算法 1 类似，计算路径与节点权重值后，将 SFC 部署在满足约束条件且权重值最高的节点上。OLPM_NDelay 在部署 SFC 时先遍历网络中的全部节点，寻找一条经过该点的权重最小的路径，并在该节点实例化 VNF。

选取以下性能指标用于评估算法的性能：1)部署成功率，指部署成功的网络服务请求数量占总网络服务请求数量的百分比；2)总收益，指服务提供商的总利润；3)总部署耗时，指计算部署方案花费的总时间；4)实例总数，指各 DC 节点上部署的 VNF 实例数量的总和。

实验选取两种类型的网络拓扑，即 14 个节点的国家科学基金网络(National Science Foundation Network，NSFNET)和 46 个节点的美国拓扑(US network，USNET)来评估算法 4 的性能。网络拓扑如图 4 所示，每个 DC 节点的容量取 1000 单位，节点的单位资源花费服从均匀分布 U(0.8，1)，链路可用带宽取 1Gbps。共生成 10 种类型的 VNF，每种 VNF 的资源需求服从均匀分布 U(10，20)，基础资源消耗 BRC 取 30 单位。每组 NSR 包含的 VNF 数量服从均匀分布 U(4，6)，收益均取 100 单位，带宽资源消耗服从均匀分布 U(10，50) Mbps，参数 σ 和 μ 分别取 0.5 和 0.15，权重系数 α 和 β 均取 0.5。两种网络的最短路径数量 k 分别取 5 和 10。

实验环境如下：处理器为 Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz，内存为 8 GB，操作系统为 Windows10 专业版。

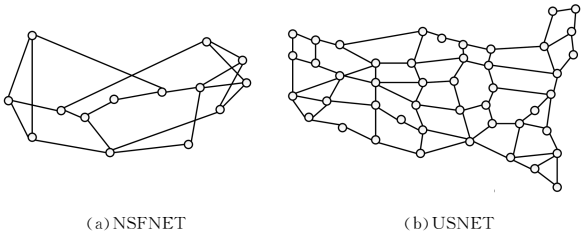

图 4　NSFNET 和 USNET 拓扑图

Fig. 4　Network topologies of NSFNET and USNET

4.2　实验结果

图 5 为 NSFNET 的仿真结果，图 6 为 USNET 的仿真结果，图 5(a)－图 5(d)和图 6(a)－图 6(d)依次为总收益、部署成功率、实例总数以及总部署耗时。如图 5 所示，NSFNET 网络约在 NSR 数量为 200 时达到部署能力上限，此时 OLPM_NDelay、ONSO_PKP 和算法 4 的总收益分别为 5764，6181 和 6818，相比其他两种对比算法，算法 4 分别提升了 18.3%及 10.3%。这是由于算法 4 引入了权重系数，能够在部署收益以及开销之间取得更好的折衷，并且通过迁移策略降低了部署过程中使用的实例总数，从而在 ONSO_PKP 的基础上进一步降低了总部署开销，提升了部署收益。由图 5(b)和图 5(c)可知，算法 4 的成功率使用的实例总数和部署成功率均优于对比算法，例如当 NSR 请求数为 300 时，算法 4 的部署成功率相比 OLPM_NDelay 和 ONSO_PKP 分别提升了 5.4%及 7.1%，运行实例总数均降低了 13.5%。

当网络规模较大时，ONSO_PKP 和算法 4 相比 OLPM_NDelay 在总收益方面的提升较为显著，以上现象可以解释为，ONSO_PKP 和算法 4 均通过设置权重系数的方式综合考虑了部署开销和收益，而 OLPM_NDelay 优先考虑部署开销最小的节点，这可能会导致 SFC 集中部署于若干热点上，难以达到整体收益最优。与 ONSO_PKP 相比，算法 4 使用迁移策略将 VNF 分散部署于最短路径中，降低了基础资源

消耗，达到了提升收益的目标。

如图 5(d)和图 6(d)所示，由于 OLPM_NDelay 算法需要遍历所有的网络节点，搜索满足部署条件的 DC 节点和权重最小的部署路径，因此当网络规模较大时，OLPM_NDelay 算法的部署时间显著增加，在 NSR 请求数量相同的情况下，USNET 消耗的时间约为 NSFNET 的 5 倍。而 ONSO_PKP 和算法 4 由于最短路径可以预先计算，缩短了计算时间，因此在 USNET 中的部署耗时均优于 OLPM_NDelay，且随着 NSR 请求数量的增加呈线性增长趋势，这也与 3.4 节中的分析相符。

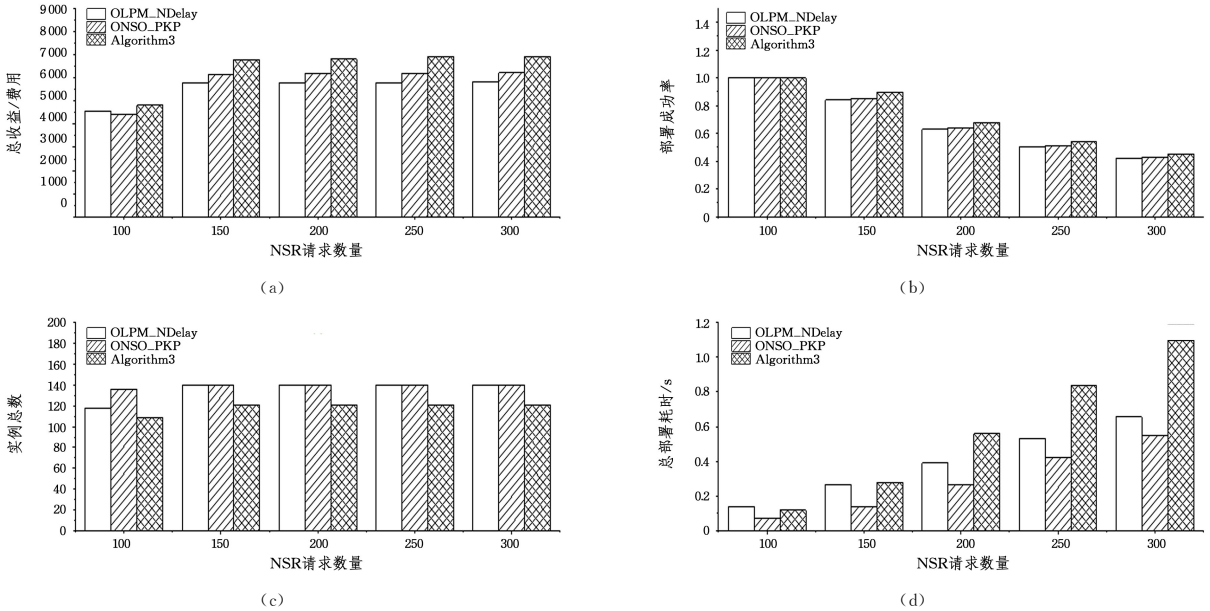

图 5 NSFNET 的仿真结果

Fig. 5 Simulation results of NSFNET

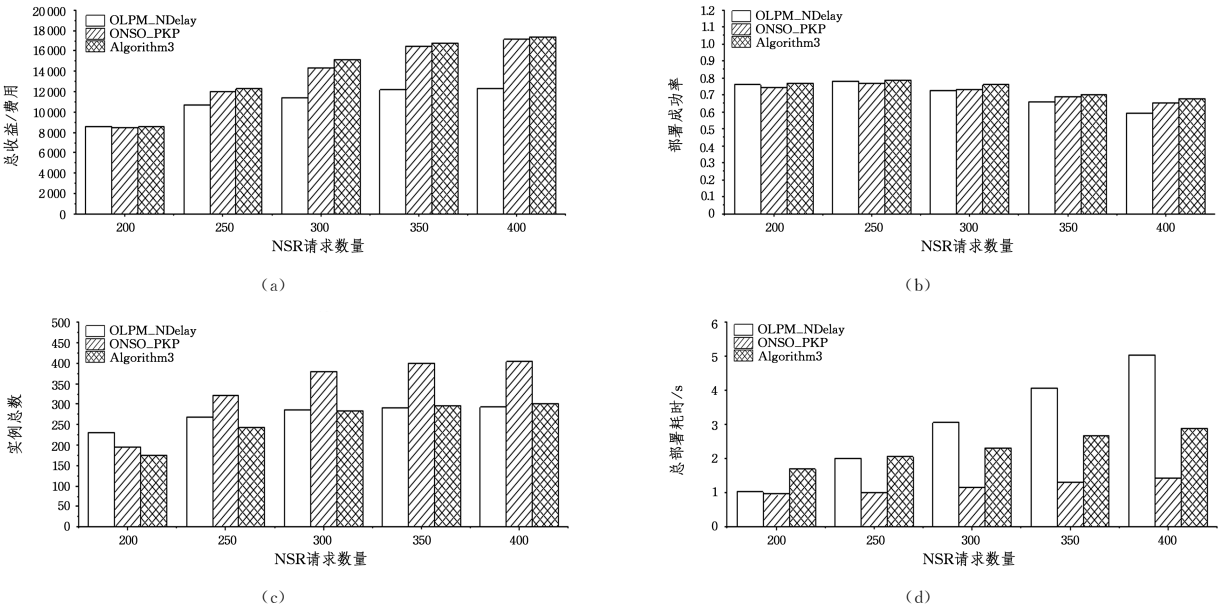

图 6 USNET 的仿真结果

Fig. 6 Simulation results of USNET

结束语 本文研究了多数据中心下的服务功能链编排问题，以服务提供商收益最大化为目标建立了优化模型，并提出了一种两阶段启发式方法。根据权重对路径和节点进行排序，优先选取优先级较高的路径和节点部署 SFC，并在此基础上设计了迁移策略，以减小基础资源消耗，增加部署成功率，提升总体收益。仿真结果表明，本文方法在优化服务链编排效果的同时能够兼顾求解效率及资源的合理利用，为改善基于 NFV 的网络服务能力提供了理论支持。本文方法的不足之处在于迁移策略仍具有一定的保守性，如何对迁移策略进行进一步改进，以提升优化效果是下一步的研究方向。

参 考 文 献

[1] CHATRAS B, OZOG F F. Network Functions Virtualization: The Portability Challenges[J]. IEEE Network, 2016, 30(4): 4-8.

[2] SONG Y, FAN L, STOJAN T, et al. Recent Advances of Re-

source Allocation in Network Function Virtualization[J]. IEEE Transaction on Parallel and Distributed Systems，2021，2(2)：295-314.

[3] HERRERA J G，BOTERO J F. Resource Allocation in NFV：A Comprehensive Survey[J]. IEEE Transactions on Network and Service Management，2017，13(3)：518-532.

[4] GHAZNAVI M，SHAHRIAR N，KAMALI S，et al. Distributed Service Function Chaining[J]. IEEE Journal on Selected Areas in Communications，2017，35(11)：2479-2489.

[5] HAN B，GOPALAKRISHNAN V，JI L，et al. Network function virtualization：Challenges and opportunities for innovations[J]. IEEE Communications Magazine，2015，53(2)：90-97.

[6] MIJUMBI R，SERRAT J，GORRICHO J L，et al. Management and Orchestration Challenges in Network Functions Virtualization[J]. IEEE Communications Magazine，2016，54(1)：98-105.

[7] SUN G，XU Z，YU H，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.

[8] LI G，REN Y，LIU Y. A Cross-Domain Service Function Chain Deployment Scheme Based on Bargaining Game[J]. Mathematical Problems in Engineering，2021，2021：1-12.

[9] WANG Y，LU P，LU W，et al. Cost-Efficient Virtual Network Function Graph（vNFG）Provisioning in Multidomain Elastic Optical Networks[J]. Journal of Lightwave Technology，2017，35(13)：2712-2723.

[10] ZHAI D，MENG X R，KANG Q Y，et al. Service Function Chain Deployment Method for Delay and Reliability Optimization[J]. Journal of Electronics & Information Technology，2020，42(10)：2386-2393.

[11] HUANG M G，WANG T，LIU L，et al. Virtual Network Function Deployment Strategy Based on Software Defined Network Resource Optimization [J]. Computer Science，2020，47(1)：404-408.

[12] WANG K，QU H，ZHAO J H. Multi-object Optimization Method Based on Reinforcement Learning in Multi-domain SFC deployment[J]. Computer Science，2021，48(12)：324-330.

[13] RACHEG W，GHRADA N，ZHANI M. Profit-driven resource provisioning in NFV-based environments[C]// IEEE International Conference on Communications. 2017：1-7.

[14] ZENG M，FANG W，ZHU Z. Orchestrating tree-type VNF forwarding graphs in inter-DC optical networks[J]. Journal of Lightwave Technology，2016，34(14)：3330-3341.

[15] MA Y，LIANG W，XU Z，et al. Profit Maximization for Admitting Requests with Network Function Services in Distributed Clouds[J]. IEEE Transactions on Parallel and Distributed Systems，2019，30(5)：1143-1157.

[16] FANG W，ZENG M，LIU X，et al. Joint Spectrum and IT Resource Allocation for Efficient VNF Service Chaining in Inter-Datacenter Elastic Optical Networks[J]. IEEE Communication Letters，2016，20(8)：1539-1542.

[17] KUO T，LIOU B，LIN K C，et al. Deploying chains of virtual network functions：On the relation between link and server usage[J]. IEEE Transactions on Networking，2018，26(4)：1562-1576.

[18] LI D，LAN J L，WANG P，et al. Service Function Chain Deployment Algorithm Based on Longest Effective Function Sequence [J]. Journal of Electronics and Information Technology，2019，41(3)：680-686.

[19] YU H，CHEN Z，SUN G，et al. Profit Maximization of Online Service Function Chain Orchestration in an Inter-Datacenter Elastic Optical Network[J]. IEEE Transactions on Network and Service Management，2021，18(1)：973-985.

[20] ZHAO D，REN J，LIN R，et al. On Orchestrating Service Function Chains in 5G Mobile Network[J]. IEEE Access，2019，7：39402-39416.

[21] MANZANARES-LOPEZ P，MUOZ-GEA J P，MALGOSA-SANAHUJA J. VNF Placement for Service Function Chains with Strong Low-Delay Restrictions in Edge Computing Networks[J]. Applied Sciences，2020，10(18)：6573-6595.

[22] YEN J. Finding the k shortest loopless paths in a network[J]. management science，1971，17(11)：712-716.

[23] MA Y，LIANG W，XU Z，et al. Profit Maximization for Admitting Request with Network Function Services in Distributed Clouds[J]. IEEE Transactions on Parallel and Distributed Systems，2018，30(5)：1143-1157.

HUANG Hua，born in 1983，Ph.D，senior engineer. His main research interests include network virtualization，business process management and service computing.

JIANG Jun，born in 1983，Ph.D，associate professor. His main research interests include artificial intelligence，data mining and big data processing.

（责任编辑：喻藜）