

基于知识图谱的服务功能链在线部署算法

徐泽汐, 庄雷, 张坤丽, 桂明宇

(郑州大学计算机与人工智能学院, 河南 郑州 450001)

摘 要: 沉浸式云 XR、全息通信等新型网络业务的出现对网络服务质量提出了更高的要求。为保证网络服务的可获得性, 在实施虚拟网络功能部署时, 必须根据网络功能的属性以及之间的依赖关系, 将网络服务的时延、可靠性等控制在一定的服务质量等级内。然而邻接矩阵、边表等传统的网络表征形式无法涵盖这些关键网络信息, 造成算法在输入阶段的信息缺失, 进而导致计算结果的偏差。因此, 为了准确提取用户需求, 反映网络资源的动态变化, 采用知识图谱对网络及其业务进行表征, 提出了一种基于知识图谱的服务功能链在线部署算法。所提算法对网络业务请求与底层网络分别进行知识提取, 构建或更新相应的知识图谱, 分解出它们的关系集合, 并基于此设计了一种基于编辑距离的关系对齐方法, 指导复杂依赖关系下的服务功能链在线部署。实验表明, 所提算法使复杂网络环境下的服务功能链部署请求接收率提高了 10%~15%, 网络平均能耗降低了约 13%, 且复杂度低, 时效性较强。

关键词: 网络功能虚拟化; 服务功能链; 知识图谱; 实体对齐; 虚拟网络功能部署

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022154

Online placement algorithm of service function chain based on knowledge graph

XU Zexi, ZHUANG Lei, ZHANG Kunli, GUI Mingyu

School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

Abstract: The emergence of new network services such as immersive cloud XR and holographic communication puts forward higher requirements for network service quality. To ensure the availability of network services, the network service delay and reliability must be controlled within a certain quality of service according to the attributes and dependencies of network functions. However, the traditional network representation forms, such as bitmap and matrix, cannot cover these key network information, resulting in the information loss in the input stage of the algorithm, which leads to the deviation of the calculation results. Therefore, in order to accurately extract user needs and reflect the dynamic changes of network resources, knowledge graph was adopted to represent the network and its services, an online placement algorithm of service function chain based on knowledge graph was proposed. Based on this, a relationship alignment method based on editing distance was designed to guide the online placement of service function chains under complex dependency relationships. Experimental results show that the proposed algorithm can improve the placement accuracy of service function chain by 10%~15% and reduce the average network energy consumption by about 13%. The proposed algorithm has low complexity and high timeliness.

Keywords: network function virtualization, service function chain, knowledge graph, entity alignment, virtual network function placement

收稿日期: 2022-04-26; 修回日期: 2022-08-01

基金项目: 国家电网有限公司总部科技基金资助项目 (No.5700-202024176A-0-0-00)

Foundation Item: The State Grid Corporation of China Science and Technology Project (No.5700-202024176A-0-0-00)

0 引言

网络功能虚拟化 (NFV, network function virtualization) [1] 实现了网元功能的软硬解耦, 推动了网络的软件化和虚拟化实现, 为未来网络提供了一种新的网络设计思路。随着互联网中间件的增多, 为满足业务需求, 国际互联网工程任务组 (IETF, Internet Engineering Task Force) [2] 对网络服务提出了更详细的要求——按特定路径, 遍历部署在不同通用设备内的多个虚拟网络功能 (VNF, virtual network function), 被称为服务功能链 (SFC, service function chain) [3]。然而, 在网络业务中, VNF 之间通常具有错综复杂的依赖关系, 例如, 解密服务器应该部署在加密服务器之后, 防火墙与网络监视器则可以被并行部署[4], 图 1 上半部分展示了一个由 4 个 VNF 组成的端到端网络服务, 其中节点代表 VNF, 边代表数据流的流向。因此, 为保证网络服务的可获得性, 当网络数据流需要穿过这组 VNF 时, 必须考虑它们之间的依赖关系, 将网络服务的时延、可靠性等控制在一定的服务质量 (QoS, quantity of service) 等级内。图 1 下半部分展示了一种网络 SFC 的组建情况, 为提高时效性, VNF₂、VNF₃ 被并行部署在 VNF₁ 后, VNF₄ 被部署在 VNF₂、VNF₃ 后, 其中节点代表 VNF, 边代表 VNF 之间的依赖关系。

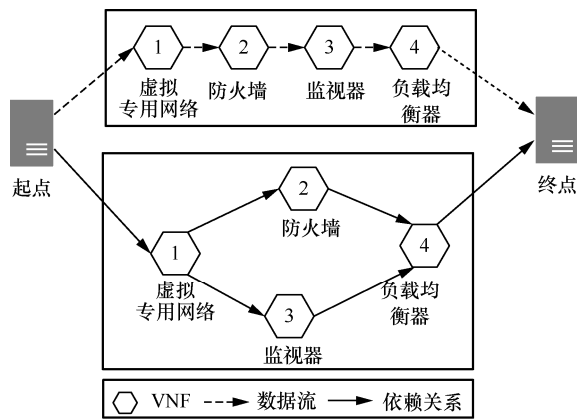


图 1 网络服务功能链

因此, 尽管 NFV 被寄予厚望, 如何链接和放置 VNF, 是 NFV 部署所面对的主要挑战之一, 也被称作 SFC 部署问题[5]。该问题旨在将一组按特定顺序排放的 VNF 高效地部署到底层网络或 NFV 基础设施中, 以适应网络客户提出的个性化要求。该问题的求解主要面临 2 个挑战: 1) 随着信息

网络业务如沉浸式云 XR 等新兴业务的出现, 为精准把控网络服务质量, 如何在遵循 VNF 之间复杂依赖关系下部署服务功能链; 2) 网络的业务需求通常是不可预知的, 对于实时到达且具有特定生存周期的网络业务请求, 如何对其进行动态且满足服务质量等级的资源分配。这对 VNF 的部署带来了持续的挑战。

针对此问题, 国内外学者关注于网络资源的可获得性, 设计了许多经典的启发式 VNF 部署方法, 例如, Pham 等[6]设计了一种启发式方法来协调 VNF 的编排并将其部署到底层网络。Zhang 等[7]设计了 2 种启发式方法来求解 SFC 部署问题, 一种是先路由后部署的方法, 另一种是基于节点优先的路由引导式部署的方法。Freitas 等[8]将 VNF 部署建模为一个综合考虑了部署与能耗的多目标问题, 并使用非支配排序遗传算法和微分进化算法 2 种优化算法来解决该问题。另外, 人工智能的蓬勃发展也引起了学术界和工业界的关注, 研究人员也基于机器学习在 NFV 部署方面进行了卓有成效的研究。例如, Soualah 等[9]将 SFC 编排问题建模为决策树, 基于蒙特卡罗树搜索利用强化学习技术设计了一种 NFV 部署算法。袁泉等[10]利用深度学习方法设计了一种基于多层前馈神经网络的虚拟网络功能资源需求预测方法, 然后根据资源需求预测结果, 基于动态编码遗传算法实现了虚拟网络功能动态部署。Chen 等[11]提出了一种基于深度强化学习的服务质量、体验质量感知的自适应在线编排方法来适应实时网络变化进行 NFV 资源配置。

需要注意的是, 在求解 VNF 部署问题时, 为使那些最初为欧几里得结构数据而开发的智能算法 (机器学习[12]、深度学习[13]、联邦学习[14]等) 能够适用于解决网络功能部署这种图类问题, 需要将网络业务先转换为统一的结构化数据, 而目前几乎所有的研究都采用了简单矩阵 (如邻接矩阵、边表) 的形式来对网络业务进行表征。但是, 使用传统矩阵表示的网络业务有 2 个致命的缺点: 1) 矩阵元素难以表示 VNF 之间错综复杂的关系; 2) 矩阵元素不便对多属性网络业务进行表示。因此利用矩阵表征的网络会使算法在输入阶段就遗失许多重要信息, 进而造成计算结果的偏差。出于同样原因, 利用矩阵表示底层网络则面临更多的局限性, 因为底层网络的资源和状态都是随着 VNF 的部署在不断变化的。因此, 以往基于矩阵表征网络而设计的

NFV 部署方法在应对上述 SFC 部署问题的 2 个挑战时，能力非常有限。

基于此，本文使用知识图谱 (KG, knowledge graph)^[15]对网络业务以及底层网络进行表征，将网络中的元素存储为以“实体-关系-实体”三元组为基本单位的结构化数据，并提出了一种基于关系对齐的 SFC 部署算法。与其他算法相比，本文算法的创新点体现在以下 3 个方面。

- 1) 使用知识图谱对网络以及复杂业务进行表征。
- 2) 定义了关系的聚合属性相似度计算方法。
- 3) 基于知识图谱中实体对齐的思想，设计了一种基于编辑距离的关系对齐方法，用以指导 SFC 的部署。

1 基于知识图谱的 SFC 部署模型

1.1 知识图谱与知识图谱实体对齐

定义 1 知识图谱。本文将一个知识图谱表示为 $KG = (E, R, A, L, T)$ ，其中 E 、 R 、 A 、 L 分别是实体集合、关系集合、属性集合以及属性值集合， $T \subseteq (E \times R \times E) \cup (E \times A \times L)$ 是知识图谱三元组集合，即知识图谱基本组成单位是描述实体之间关系的“实体 1-关系-实体 2”三元组，以及描述实体属性的“实体-属性-值”三元组。

定义 2 知识图谱实体对齐。给定 2 个知识图谱 $KG_0 = (E_0, R_0, A_0, L_0, T_0)$ 和 $KG_1 = (E_1, R_1, A_1, L_1, T_1)$ ，知识图谱实体对齐^[16]被定义为对 E_0 中的每个实体，找出 E_1 中的等效实体，即

$$\text{Align}_{\text{entity}}(KG_0, KG_1) = \{(e_0, e_1) | e_0 \in E_0, e_1 \in E_1\} \quad (1)$$

1.2 问题规划

为实现对网络业务中功能组件、VNF 之间的依赖关系，以及网络资源数目的准确表示，本文将网络业务与底层网络这种图类数据转存为三元组形式的结构化数据，分别得到知识图谱

$KG_f = (E_f, R_f, A_f, L_f, T_f)$ 表示的 SFC 和知识图谱 $KG_s = (E_s, R_s, A_s, L_s, T_s)$ 表示的底层网络，表 1 展示了 KG_f 与 KG_s 的结构化信息。基于此，SFC 部署问题可以表示为

$$\text{Align}_{\text{entity}}(KG_f, KG_s) = \{(e_f, e_s) | e_f \in E_f, e_s \in E_s\} \quad (2)$$

在传统的知识图谱实体对齐方法中，通常通过评估实体的相似度来对齐不同知识图谱中的实体，其本质上为分类问题。然而，在本文提出的在线 SFC 部署问题中，需保证在部署 VNF 时能够兼顾网络的资源属性以及不同 VNF 之间的依赖关系，以实现实时、按需的分配。因此，为使知识图谱实体对齐方法能够适应 SFC 部署问题，本文聚焦于网络中的依赖关系，在结构化 SFC 与底层网络的信息构建与之对应的知识图谱 KG_f 与 KG_s 之后，分解出知识图谱 KG_f 与 KG_s 的关系集合，通过对 2 个关系集合中的所有元素执行对齐操作，即 $\text{Align}_{\text{entity}}(KG_f, KG_s) = \{(R, r) | R \in R_f, r \in R_s\}$ ，以实现在线 SFC 部署。这样，由于 SFC 与底层网络被转换为以“实体 1-关系-实体 2”为单位的知识集合，在部署 SFC 时，物理链路与其两端的物理节点被视为一个整体，用以承载有依赖关系的 2 个 VNF，这就避免了任何额外的隐藏资源被使用，保证了在整个部署过程中消耗的总能量最少。同时也解决了基于“点对点”设计的矩阵表征网络 SFC 部署方法由于忽略了 VNF 之间的依赖关系而造成的 QoS 难以把控等问题。

2 算法设计

在现实网络世界中，服务、应用程序和用户需要与基础设施网络之间进行实时的交互。因此，在网络功能虚拟化背景下，必须对网络业务进行在线分析和部署，进而有效利用共享资源，同时需定期更新底层网络的资源状况，以便在其他网络业务需求到达时自动评估部署的可能性。基于此，本

表 1

KG_f 与 KG_s 的结构化信息

知识图谱	Schema	描述
KG _f	(实体 1_TYPE: VNF _i , 实体 2_TYPE: VNF _j , 关系_TYPE: 依赖)	三元组的主体是 VNF _i ，与主体关系为依赖的实体是 VNF _j ，有向边的另一端的 VNF _j
	(实体_TYPE: 虚拟网络功能 f_i , 属性_TYPE: f_i 的属性, 属性值_TYPE: f_i 的属性)	三元组的主体是 f_i ，属性与属性值分别是 f_i 的属性、属性值
KG _s	(实体 1: 物理节点 a, 实体 2: 物理节点 b, 关系_TYPE: 连接)	三元组的主体是物理节点 a，与主体关系为连接的实体是节点 a 另一端的物理节点 b
	(实体_TYPE: 物理节点 n , 属性_TYPE: n 的属性, 属性值_TYPE: n 的属性值)	三元组的主体是 VNF，属性与属性值分别是 VNF 的属性、属性值

文提出一种针对在线场景设计的算法，其主要目标是在考虑资源受限的网络环境下，实现具有复杂依赖关系的 SFC 的实时部署。本文算法一次处理一个网络业务请求，并定期更新底层网络，以释放更多的资源供新的网络业务请求使用。

2.1 系统模型

2.1.1 底层网络

在一个 NFV 使能网络环境中，物理基础设施网络被表示为一个无向连接图 $G_s=(N,L)$ ，其中 N 表示物理节点集合， L 表示物理链路集合。任一节点 $n \in N$ 都拥有用 $loc(n)$ 表示的位置属性，以及用 $Cap(n)$ 表示的资源容量，这里的资源包含计算资源与存储资源，每个节点都根据场景（例如，虚拟 CPU 核心数、CPU 时间等）选用合适的单位。对于每条连接物理节点 a 和 b 的双向物理链路 $l_{ab} \in L$ ，具有带宽和时延 2 个基本属性。

2.1.2 SFC 请求

如图 2 所示，SFC 请求被描述为一个有向的加权图 $G_f=(F,A)$ ，其中 F 表示 VNF 的集合， $|F|$ 表示该业务中的 VNF 数量； A 表示 SFC 中的依赖关系集合，是 VNF 之间的传输路径，在本文中也称为虚拟链路。对于每个 $VNF \in F$ 都有一定的部署位置要求与计算资源、存储资源需求，分别用 $loc(f)$ 、 $D(f)$ 表示。另外，在线场景中，VNF 之间的虚拟链路通常有着最高数据率与时延的要求，且 SFC 通常包含到达时间与生存时间 2 个属性，它们反映了网络业务的动态特征。

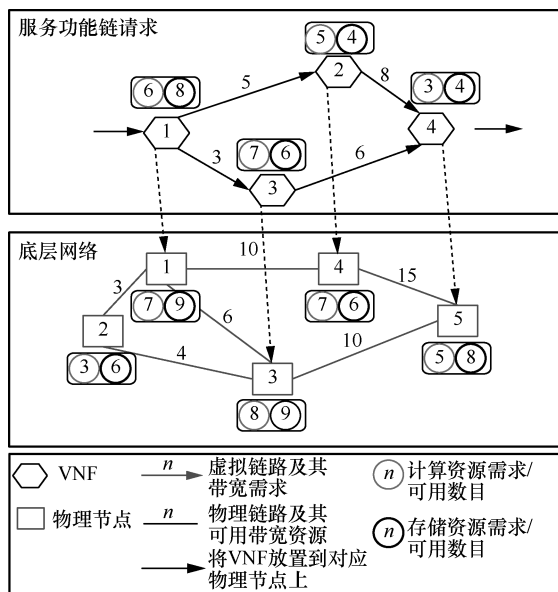


图 2 SFC 部署示例

2.2 基于知识图谱的在线 SFC 部署算法

在任意时刻 t ，为将动态到达的 SFC_{*i*} 部署到满足 QoS 的 NFV 基础设施上，首先，对该网络业务请求与底层网络进行知识提取，构建或更新相应的知识图谱，然后分解出它们的关系集合，分别命名为 S_f 、 S_s 。接下来，定义 2 个关系集合中关系的聚合属性相似度，利用本文设计的基于编辑距离的关系对齐方法求得该 SFC_{*i*} 部署的解。在每次迭代部署中，算法都检查网络业务的生存时间，以便将失效的 SFC 从底层资源中删除，进而更新整个底层网络。基于知识图谱的在线 SFC 部署算法如算法 1 所示，具体操作流程如下。

算法 1 基于知识图谱的在线 SFC 部署算法

输入 动态到达的 SFC 和底层网络

输出 对齐后的 R 与 r

- 1) $t \neq 0$
- 2) 初始化底层网络，构建其知识图谱 KG_s
- 3) 分解 KG_s ，得到其关系集合 S_s
- 4) for 任意时刻 t 到达的 SFC_{*i*}
- 5) 初始化 SFC_{*i*}，构建其知识图谱 KG_f
- 6) 分解 KG_f ，得到其关系集合 S_f
- 7) 根据式(3)计算关系集合 S_s 与 S_f 的关系匹配度矩阵 S_{fs}
- 8) 使用单纯形法求解式(20)的线性规划问题，执行关系对齐
- 9) if S_f 中每一个关系 R 均在底层网络中找到满足式(17)~式(19)约束的 r
- 10) 输出对齐的 R 与 r 的关系 ID 与实体 ID
- 11) 更新 KG_s 的 sheet1，关闭未使用的物理节点
- 12) else 报告失败
- 13) end if
- 14) if SFC 列表非空，跳转到步骤 3)
- 15) end if
- 16) end for

2.2.1 初始化

首先，提取初始状态的底层网络的知识，包括物理节点、物理链路，以及它们初始状态的资源数目，将其存储为以三元组为基本单位的结构化数据，并为其构建知识图谱 $KG_s=(E_s,R_s,A_s,L_s,T_s)$ 。其中， KG_s 的实体指代了物理节点；由于底层网络的拓扑结构在物理上是固定的，即构成底层网络的主要元素节点和链路的数量以及连通性也是固定不变的，因此，本文

中两节点之间的物理链路就对应着 KG_s 的实体关系；根据 2.1.1 节可知， A_s 包括实体属性——物理节点的计算资源 $Cap_{cpu}(n)$ 和存储资源 $Cap_{mem}(n)$ 与关系属性——物理链路的带宽 $Cap_{bw}(l)$ 和时延 $Cap_d(l)$ ，其值用 $L_s = \{R_{s1}, R_{s2}, \dots, R_{s|n|}\}$ 表示， $T_s \subseteq (E_s \times R_s \times E_s) \cup (E_s \times A_s \times L_s)$ 是 KG_s 的基本单位三元组集合。

然后，提取动态到达的 SFC 的知识，即组成该网络业务的 VNF、VNF 之间的依赖关系，以及资源需求，用知识图谱 $KG_f = (E_f, R_f, A_f, L_f, T_f)$ 表示。 KG_f 中， E_f 代表的实体集合即网络业务中的所有 VNF 组件； R_f 代表的关系集合即 VNF 之间的依赖关系集合，它是从起点开始，按照广度优先顺序遍历得到的该 SFC 弧的有序集合。对于每个关系 R_i ，实体 1 是其对应弧的起点的 ID，实体 2 是其对应弧的终点的 ID。根据 2.1.2 节可知， A_f 包括实体属性——VNF 的计算资源需求量 $D_{cpu}(f)$ 和存储资源需求量 $D_{mem}(f)$ 与关系属性——虚拟链路的最高数据率 $D_{bw}(L)$ 与时延 $D_d(L)$ 的集合，其值用 $L_f = \{D_{f1}, D_{f2}, \dots, D_{f|F|}\}$ 表示， $T_f \subseteq (E_f \times R_f \times E_f) \cup (E_f \times A_f \times L_f)$ 是 KG_f 的基本单位三元组集合。

另外，为方便操作，本文将结构化数据分别存储在其知识图谱的命名为 sheet1 和 sheet2 的实体集合与关系集合列表中。表 2~表 5 展示了根据图 2 中底层网络与 SFC 所构建的知识图谱 KG_s^1 与 KG_f^1 的实体属性结构化信息。

表 2 知识图谱 KG_s^1 的实体列表 sheet1

实体 ID	实体名	实体属性($Cap_{cpu}(n)$, $Cap_{mem}(n)$)
1	n_1	(7,9)
2	n_2	(3,6)
3	n_3	(8,9)
4	n_4	(7,6)
5	n_5	(5,8)

表 3 知识图谱 KG_s^1 的关系列表 sheet2

关系 ID	实体 1ID	实体 2ID
r_1	n_1	n_2
r_2	n_1	n_3
r_3	n_1	n_4
r_4	n_2	n_3
r_5	n_3	n_5
r_6	n_4	n_5

表 4 知识图谱 KG_f^1 的实体列表 sheet1

实体 ID	实体名	实体属性($D_{cpu}(f)$, $D_{mem}(f)$)
1	f_1	(6,8)
2	f_2	(5,4)
3	f_3	(7,6)
4	f_4	(3,4)

表 5 知识图谱 KG_f^1 的关系列表 sheet2

关系 ID	实体 1ID	实体 2ID
r_1	n_1	n_2
r_2	n_1	n_3
r_3	n_1	n_4
r_4	n_2	n_3

2.2.2 分解知识图谱

与知识图谱传统的实体对齐方法不同，在 SFC 部署问题中，只有所有的 VNF 都按照既定的依赖关系部署到底层网络中，才能得到一个有效的部署解，即

$$\text{Align}_{\text{entity}}(KG_f, KG_s) = \{(R, r) \mid R \in R_f, r \in R_s\} \quad (3)$$

在本文中，对齐 KG_f 与 KG_s 中的关系，需为 R_f 中的每一个 R 找到 R_s 中与之匹配的 r ，即将虚拟/物理链路及其两端节点看作一个整体，作为关系对齐的基本单位。因此，本文在初始化 SFC 与底层网络后，分解它们对应的知识图谱，分别得到 KG_f 与 KG_s 的用 $n \times n$ 矩阵表示的关系集合 \mathcal{S}_f 与 \mathcal{S}_s ，其中，关系集合的对角线元素 f_{ii} 与 s_{ii} 分别是关系 R_i 与 r_i 的聚合属性值，用 $f(R_i)$ 与 $s(r_i)$ 表示。关系的聚合属性包含了其对应的虚拟/物理链路的资源属性，以及关系中实体 1、实体 2 的资源属性，即实体属性与关系属性的聚合。集合中其他元素表示关系之间是否存在相继关系，若有则为 1，反之为无穷。例如，图 3 是分解 KG_s^1 与 KG_f^1 得到的关系集合。

2.2.3 定义关系聚合属性相似度

为对齐实体，一般会定义 2 个实体的属性相似性函数，用以评估 2 个实体的相似度^[17]。同样地，在评估 2 个关系的相似度时，本文基于一种广泛使用的 Jaccard 相似度^[18]计算方法，设计了 2 个关系集合中关系的聚合属性相似度。

$$s_{ii} - f_{ii} = \frac{|s(r_i) \odot f(R_i)|}{|s(r_i) \oplus f(R_i)|} \quad (4)$$

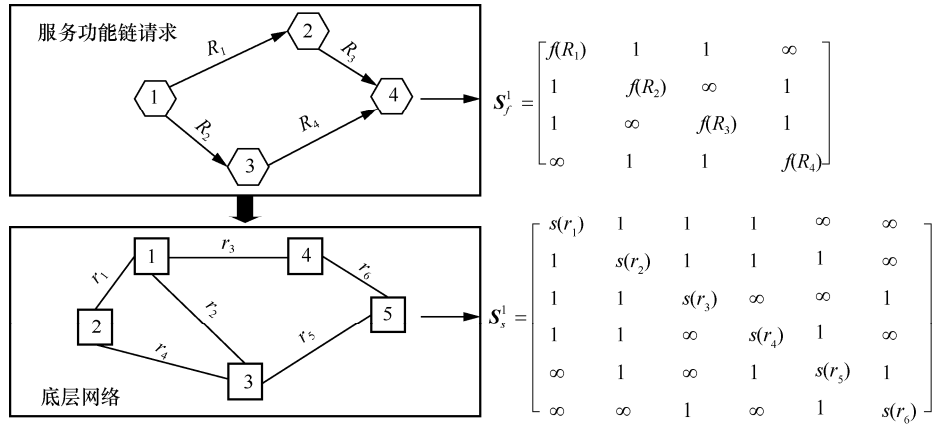


图 3 分解 KG_s^1 与 KG_f^1 得到的关系集合

其中，2 个关系集合对角线元素之间的减操作代表 2 个关系的聚合属性相似度计算， $|s(r_i) \oplus f(R_i)|$ 代表 2 个关系集合中待匹配元素的聚合属性中相同属性的个数，为简化操作，本文仅讨论底层 NFV 设备与 VNF 的资源属性一致的情况； $|s(r_i) \odot f(R_i)|$ 代表对 r_i 与 R_i 的所有属性值执行 \odot 操作之后的值，若 r_i 中所有元素的资源剩余量均大于 R_i 需求量，则 $|s(r_i) \odot f(R_i)|$ 与分母数值相等，否则为正无穷。例如图 3 的 2 个关系集合中， R_1 与 r_1 均考虑计算与存储资源 2 个节点属性以及时延与带宽 2 个链路属性，因此， $|s(r_1) \oplus f(R_1)| = 4$ ；另外，根据图 2 中所标的资源数目值可知， r_1 中的 2 个节点及其之间的链路剩余资源数目均大于 R_1 的资源需求量，因此， $|s(r_1) \odot f(R_1)| = |s(r_1) \oplus f(R_1)| = 4$ ；则 R_1 与 r_1 的聚合属性相似度为

$$s_{11} - f_{11} = \frac{|s(r_1) \oplus f(R_1)|}{|s(r_1) \odot f(R_1)|} = \frac{4}{4} = 1。$$

2.2.4 线性规划求解关系对齐

基于实体对齐中广泛使用的基于编辑距离^[19]的相似性度量方法，本文将关系集合 S_s 与 S_f 之间的距离定义为最小化目标函数，将式(2)的关系对齐问题表示为

$$J(\Phi) = \sum_{i=1}^n \sum_{j=1}^n (s(r_i) - \Phi(f(R_j)))^2 \quad (5)$$

其中， Φ 是关系 r 与 R 之间的映射关系。该匹配问题旨在最小化距离 $J(\Phi)$ ，因此，可以重新将该问题表示为

$$\min_M \|S_s - MS_f M^T\| \quad (6)$$

其中， M 是本文算法的部署函数，维度为 $n \times n$ ， M 的行对应 SFC 的 VNF，列对应底层网络中的物理节点， M 中的每一行和每一列都只有一个元素为 1，其他元素均为 0 (元素值为 1 代表 VNF 被放置在该列的物理节点上)； $S_s - MS_f M^T$ 是经过函数 M 作用后 S_s 与 S_f 的差； $\|\cdot\|$ 是 L1 范式。

根据文献[19]可知，问题式(6)是多项式时间无解的，并且一个中等规模的底层网络通常具有数百个节点，若使用暴力枚举法，例如分支界限和枚举搜索，得到整数解的时间复杂度将为指数级^[20]。因此，本文利用一种图匹配^[21]方法将该问题化简为线性问题。具体操作如下。

首先，为了符合图匹配问题中顶点数相同的要求，在矩阵中增加参数为 0 的元素，将关系矩阵 S_f 扩充到与 S_s 相同的维度。如图 3 中给出的一组待对齐的 SFC 与底层网络，由于 SFC 的关系数为 4 而底层网络为 6，因此需要在 SFC 的关系矩阵 S_f 中增加 2 行 0 元素和 2 列 0 元素，使其与物理邻接矩阵维度相同。

接下来，定义一个 $|R_s| \times |R_s|$ 的矩阵 H_1

$$H_1 = S_s - MS_f M^T \quad (7)$$

其中， M 为正交矩阵， $MM^T = I$ ， I 为单位矩阵，则式(7)右乘 M 可得

$$H_1 M = S_s M - MS_f \quad (8)$$

因为 M 是置换矩阵，式(8)的 L1 范式为

$$\|H_1 M\|_1 = \|H_1\|_1 = \|S_s M - MS_f\| \quad (9)$$

将残差矩阵表示为 H

$$H = S_s M - MS_f \quad (10)$$

接下来，将矩阵 $H=\{h_{ij}\}$ 和 $M=\{m_{ij}\}$ 按列划分为

$$\text{VEC}(H)=\{h_{11}, h_{21}, \dots, h_{1n}, h_{2n}, \dots, h_{nn}\}^T \quad (11)$$

$$\text{VEC}(M)=\{m_{11}, m_{21}, \dots, m_{1n}, m_{2n}, \dots, m_{nn}\}^T \quad (12)$$

基于此， H 可被表示为

$$\text{VEC}(H)=S_{fs} \text{VEC}(M) \quad (13)$$

其中， S_{fs} 是根据关系矩阵 S_f 与 S_s 派生的 $|R_s|^2 \times |R_s|^2$ 的关系匹配度矩阵，定义为

$$S_{fs} = \begin{bmatrix} \{S_s - f_{11} I_{|R_s|}\} & \{-f_{21} I_{|R_s|}\} & \dots & \{-f_{|R_s|1} I_{|R_s|}\} \\ \{-f_{12} I_{|R_s|}\} & \{S_s - f_{11} I_{|R_s|}\} & \dots & \{-f_{|R_s|2} I_{|R_s|}\} \\ \vdots & \vdots & \ddots & \vdots \\ \{-f_{|R_s|1} I_{|R_s|}\} & \{-f_{2|R_s|} I_{|R_s|}\} & \dots & \{S_s - f_{11} I_{|R_s|}\} \end{bmatrix} \quad (14)$$

其中， $I_{|R_s|}$ 是一个 $|R_s| \times |R_s|$ 的单位矩阵，则关系匹配度矩阵的对角线元素 $\{S_s - f_u I_{|R_s|}\}$ 为

$$S_s - f_u I_{|R_s|} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{|R_s|1} \\ s_{21} & s_{22} & \dots & s_{2|R_s|} \\ \vdots & \vdots & \ddots & \vdots \\ s_{|R_s|1} & s_{|R_s|2} & \dots & s_{|R_s||R_s|} \end{bmatrix} - f_u \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \begin{bmatrix} s_{11} - f_{ii} & s_{12} & \dots & s_{|R_s|1} \\ s_{21} & s_{22} - f_{ii} & \dots & s_{2|R_s|} \\ \vdots & \vdots & \ddots & \vdots \\ s_{|R_s|1} & s_{|R_s|2} & \dots & s_{|R_s||R_s|} - f_{ii} \end{bmatrix} \quad (15)$$

其中，对角线元素 $s_{jj} - f_{ii}$ 的值为 2.2.3 节中定义的 2 个关系属性的相似度。

因此，目标函数就等价于

$$\min_M \|\text{VEC}(H)\|_1 = \min_M \|S_{fs} \text{VEC}(M)\|_1 = \min_m \|S_{fs} m\|_1 \quad (16)$$

其中， $m = \text{VEC}(M)$ 是 $n^2 \times 1$ 的向量。且该目标函数的求解需受到如下所示的位置约束 ($\forall R_i \in \text{KG}_f, r_j \in \text{KG}_s$)

$$\text{若 } \Phi(R_i) = r_j, \text{ 则有 } e_{f1}, e_{f2} \in R_i, e_{s1}, e_{s2} \in r_j \quad (17)$$

$$\text{若 } \Phi(e_{f1}) = e_{s1}, \text{ 则有 } \text{loc}(e_{f1}) = \text{loc}(e_{s1}) \quad (18)$$

$$\text{若 } \Phi(e_{f2}) = e_{s2}, \text{ 则有 } \text{loc}(e_{f2}) = \text{loc}(e_{s2}) \quad (19)$$

式(17)~式(19)确保了被部署到底层网络上的节点对之间均无额外的边。若有关系 R_i 与 r_j 通过关系

对齐成功匹配， e_{s1} 与 e_{s2} 分别是它们的实体 1 与实体 2，则 e_{f1} 与 e_{f2} 代表的 VNF 需分别部署到 e_{s1} 与 e_{s2} 所代表的物理节点的位置上，且一个 VNF 只能被部署到一个物理节点上。

因此，SFC 的在线部署问题可被表示为

$$\begin{aligned} & \min_m \|S_{fs} m\|_1 \\ & \text{s.t. 式(17)~式(19)} \end{aligned} \quad (20)$$

对于某时刻 t 到达的 SFC，分解该 SFC 与底层网络对应的知识图谱，分别得到关系矩阵，作为本文算法的输入，使用单纯形法求解目标函数，得到部署解。

2.2.5 更新底层网络

一旦成功部署了 t 时刻到达的 SFC，算法需更新 KG_s 的 sheet1，以更新当前底层网络的剩余资源数目，并根据到达时间开始处理下一个 SFC。但当该 SFC 中有任意一个的关系元组未能在底层网络中匹配到满足资源需求量的物理设备时，则标记对齐失败，算法拒绝该 SFC，开始处理下一时刻到达的 SFC。

2.2.6 复杂度分析

通常服务功能链包含 3~8 个不同的网络功能，因此，在不考虑网络业务数目的情况下，本文算法根据构成底层网络的节点总数 $|M|$ 和边总数 $|L|$ ，从其知识图谱中分解出关系矩阵，即需在 $O(|M| + |L|)$ 处理时间内搜索并列出了所有关系。一旦列出了所有关系，便可进入关系对齐阶段，该过程消耗的处理时间几乎可以忽略不计，因为经过了线性化的处理之后，对齐阶段的主要工作是比较待匹配关系中所有元素的属性值，其复杂度为 $O(1)$ 。

3 实验仿真与分析

3.1 实验环境

仿真采用 TiNet^[22] 网络拓扑，包含 54 个节点和 89 条全双工链路。网络资源容量均服从均匀分布，其中，节点的 CPU 资源容量与存储资源容量的分布区间均为 $[10, 15]$ ，链路的带宽分布区间为 $[50, 100]$ ，传输时延分布区间为 $[1, 50]$ 。节点和链路能量消耗中常量值设置为 $P_l = 150, P_b = 150, P_n = 15$ 。网络可以提供 8 种功能，每个需要部署的服务功能链由 3~8 个不同的网络功能组成，VNF 的主要参数同文献[23]，如表 6 所示。

表 6 VNF 的主要参数

类型	CPU	MEM
1	4	4
2	3	3
3	3	3
4	2	2
5	2	2
6	2	2
7	1	1
8	1	1

任意 VNF 之间具有依赖关系的概率为 0.5, 编排方式同文献[4], VNF 之间的虚拟链路对最高数据率与时延的要求服从[20,50]的随机分布。为了提高在线算法评估的准确性, 共执行 2 000 个服务功能链的部署测试, 其中网络业务请求的到达时间服从泊松分布, 平均每 1 000 个时间单元约到达 40 个; 生存时间服从指数分布, 平均为 40 个时间单元。实验将数据采集点设置在时间窗的开端, 一个时间窗等于 5 000 个时间单元, 实验仿真在 10 个时间窗内执行完毕。

3.2 初始化结果

TiNet 拓扑被转换为 197 个三元组, 其中包括 89 个关系三元组和 108 个属性三元组。具体来说,

本文考虑的底层网络的知识包括物理节点、物理链路和资源数目, 分别为它们打上节点、链接和属性的标签。同样地, 每个在线到达的网络业务也根据标签被抽取相应的知识。图 4 展示了本文利用文献[24]中的 Echarts 工具所呈现的部分知识图谱的可视化结果, 其中图 4(a)是 TiNet 的网络拓扑结构, 图 4(b)是对网络执行知识抽取后部分三元组的可视化结果。与仅能包含部分信息的其他传统网络表征方法相比, 知识图谱表征的网络能够提取网络中的全部重要信息(关系与属性值), 避免了算法在输入阶段重要信息遗失而造成的计算结果的偏差。

3.3 实验分析

3.3.1 SFC 部署评价指标

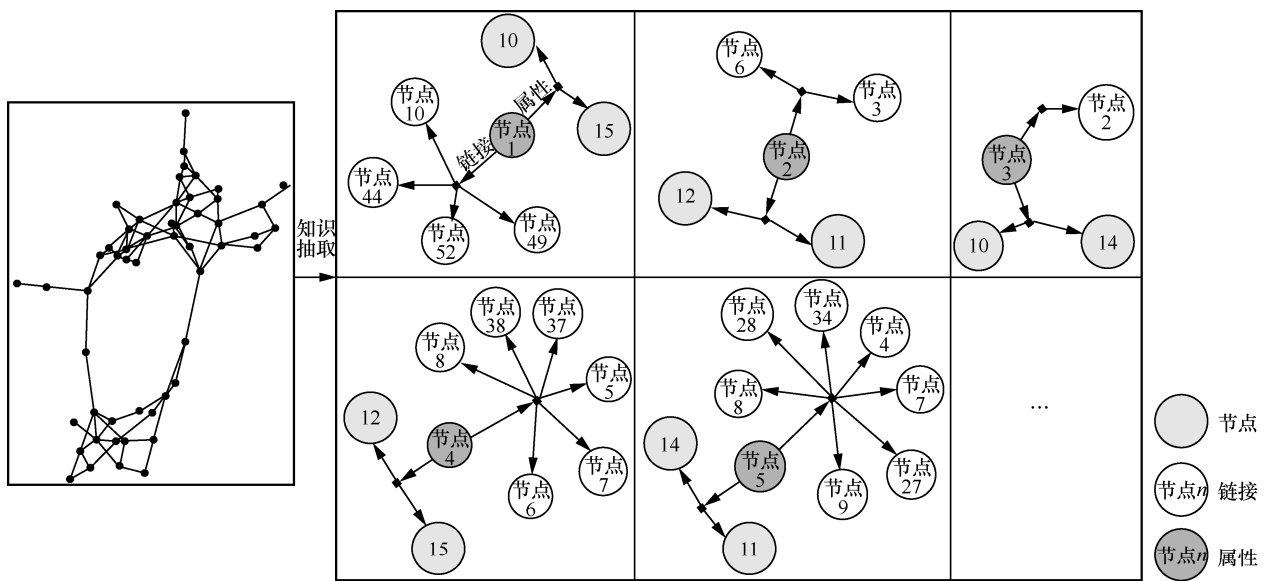
本文基于如下 4 个指标^[11]评估算法的性能。

1) 请求接收率: 一段时间内成功部署的 SFC 个数占实际总网络业务请求数量的比例。

$$P = \frac{\sum_{i=1}^{n_o(T)} \sigma(\text{SFC}_i)}{n_o(T)} \quad (21)$$

其中, $n_o(T)$ 表示 T 时间段内到达网络业务请求数, $\sigma(\text{SFC}_i)$ 是一个取值为 0 或 1 的二进制变量, 表示第 i 个 SFC 是否部署成功。

2) 总收益: 至 t 时刻, 部署网络请求所产生的



(a) TiNet的网络拓扑结构

(b) 部分三元组的可视化结果

图 4 TiNet 网络知识抽取可视化结果

总收益。

$$R_t = \sum_{f \in F} D_{\text{cpu}}^{\text{accept}}(f) + \sum_{a \in A} D_{\text{bw}}^{\text{accept}}(a) \quad (22)$$

其中, $D_{\text{cpu}}^{\text{accept}}(f)$ 和 $D_{\text{bw}}^{\text{accept}}(a)$ 分别是成功部署的 VNF 的计算资源以及这些 VNF 之间的虚拟链路的带宽资源。

3) 平均能耗: 每个时间间隔 T 后所有物理节点和链路的总功耗的平均值, 单位为 W 。

$$E = \frac{1}{T} \left(\sum_{n \in N} E(n) + \sum_{l \in L} E(l) \right) \quad (23)$$

若处于开启状态, $E(n) = p_n + \tau(p_b - p_n)$, p_n 、 p_b 分别为物理节点的基本能耗和满负荷时的能耗, τ 为节点的 CPU 负载率, $E(l) = p_l$, p_l 为物理链路能耗, 一般为常量; 若处于关闭状态, 则无能量消耗。

4) 运行时间: 客户端发出网络业务请求到其被成功部署到底层网络上所需的时间。

3.3.2 部署结果分析

为验证所提算法的性能和有效性, 本文选取 SFC-MCT 算法及 First-Fit (FF) 算法进行对比实验。SFC-MCT 算法^[9]将部署问题转化为决策树搜索, 树中的每个节点对应于一个 VNF 的匹配方案, 再用蒙特卡罗树搜索算法找到一个奖励值最大的 SFC 部署方案, 是一种基于强化学习的 SFC 部署算法。FF 算法采用 First-Fit 算法设计了一种启发式的 SFC 部署算法, 先为每一个 VNF 匹配资源数目最为充足的物理节点, 然后在这些节点之间使用 Dijkstra 在算法部署传输路径, 是 SFC 部署的经典基准线算法^[11,22]。

图 5 显示了请求接收率的对比结果。结果表明, 本文算法能够接收所有到达系统的网络请求, 当系统达到稳定状态后, FF 算法与 SFC-MCT 算法的成功率在 85% 左右, 因此, 本文算法性能明显优于对比算法。这是因为无论是基于启发式的 FF 算法还是基于强化学习的 SFC-MCT 算法, 这些 SFC 部署方法都是将矩阵表征的 SFC 与底层网络作为算法的输入, 缺少了网络中的关系以及网络属性等关键信息, 复杂的网络环境便对它们的有效性提出了考验。随着网络资源逐渐被分配, SFC 匹配到既满足依赖关系又能达到服务质量的物理资源的请求接收率便有所下降。

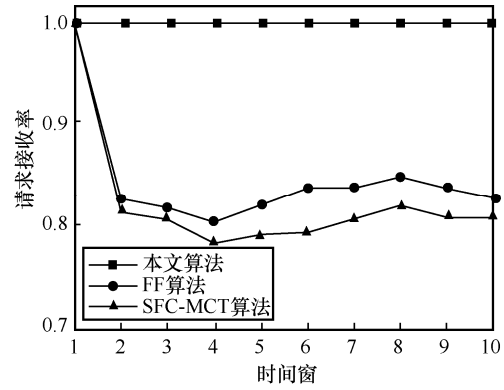


图 5 请求接收率的对比结果

图 6 显示了总收益的对比结果。从图 6 中可以看出, 与其他算法相比, 本文算法使互联网供应商取得了最高的总收益, 这与请求接收率的结果相吻合。

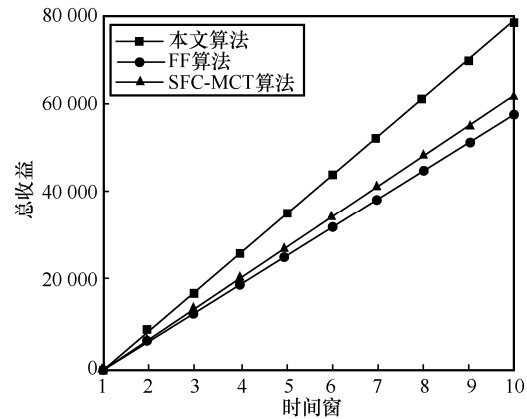


图 6 总收益的对比结果

图 7 显示了运行时间的对比结果。从图 7 可以看到, 本文算法平均部署时间约为 0.01 ms, FF 算法与 SFC-MCT 算法约为 0.04 ms 与 0.21 ms, 因此, 本文算法在运行时间上取得了明显的优势。这是因为 FF 算法与 SFC-MCT 算法的输入数据中未包含 VNF 之间的依赖关系, 因此匹配过程是基于点对点的, 即先部署 VNF 再连接它们之间的传输路径, 这也是目前大多部署算法所面临的问题。这些算法每部署一个 VNF 都需要对整个底层网络中的节点进行比对, 复杂度高, 并且需要额外的链路部署时间。而本文算法仅需从输入的结构化数据中搜索并列出现所有关系后, 便可进入关系比对阶段, 并且因为经过了线性化的处理, 对齐过程消耗的处理时间几乎可以忽略不计。本节实验是在 3.4 GHz CPU、16 GB 内存的计算机上通过 Pycharm 进行的仿真实验。

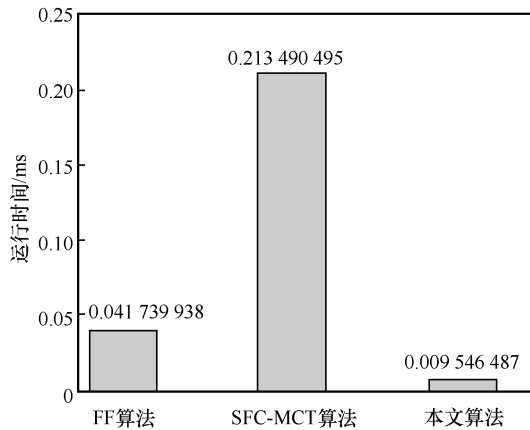


图 7 运行时间的对比结果

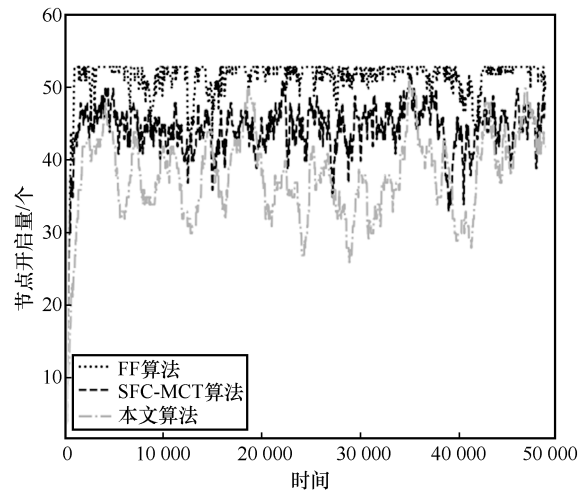


图 9 节点开启量的对比结果

图 8 是平均能耗的对比结果，结果表明，本文算法的平均能耗比 FF 算法低约 30%，比 SFC-MCT 算法的低约 13%。这是因为 FF 算法是先根据节点资源数目启发式地计算得到网络业务中 VNF 的部署位置，VNF 的部署位置通常较为分散，因此，通过 Dijkstra 算法计算得到的 VNF 之间的传输路径就要占用较多的链路资源；SFC-MCT 算法虽然利用强化学习计算得到了奖励值最高的部署方案，但节点与链路的关系仍然是被割裂地考虑的，在部署过程中不可避免地会开启一些隐藏节点。而本文算法在部署 SFC 时，SFC 与底层网络被转换为以“实体 1-关系-实体 2”为单位的知识集合，以节点和链路组成的三元组为单位进行部署，这就避免了任何额外的隐藏资源被使用，因此在整个部署过程中消耗的总能量最少。图 9 与图 10 也证实了这一结果，随着服务功能链的持续部署，本文算法的节点开启量与链路开启量在整个过程中一直是最少的。

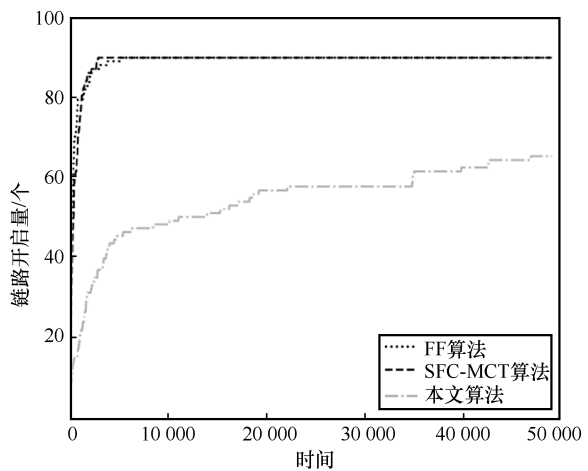


图 10 链路开启量的对比结果

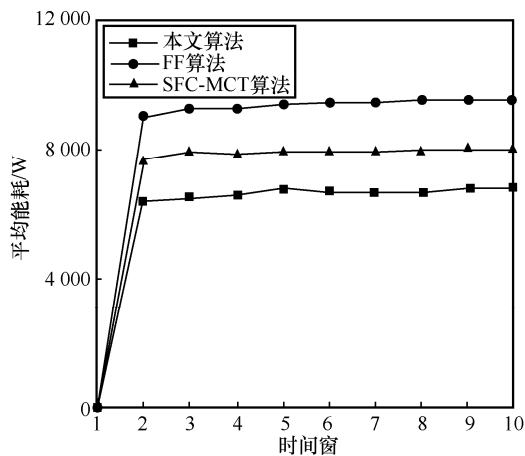


图 8 平均能耗的对比结果

4 结束语

新型网络业务的出现对服务功能链的部署提出了更高的要求，即必须在资源受限的前提下遵循网络功能之间的依赖关系执行部署。因此，针对具有复杂依赖关系服务功能链在线部署问题，本文利用知识图谱对网络业务及底层网络进行表征，提出了一种基于知识图谱的服务功能链部署方法。与基于传统矩阵表征的部署算法相比，由于本文算法将网络中的属性与关系信息转化为了以实体-关系-实体的结构化数据，没有造成重要信息的缺失，因此在线部署网络功能时，其精确性与时效性均有所提高。此外，虽然将知识图谱引入了服务功能链部署中，但使用知识图谱对网络进行表征学习还有深入研究的空间。因此，下一步将继续在知识图谱表征服务网络上展开工作，挖掘更加精准、有效的表征方法。

参考文献：

- [1] 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.
- [2] ETSI. European telecommunications standards institute[R]. 2016.
- [3] SHERRY J, RATNASAMY S. A survey of enterprise middlebox deployments[R]. 2012.
- [4] YANG S, LI F, YAHYAPOUR R, et al. Delay-sensitive and availability-aware virtual network function scheduling for NFV[J]. IEEE Transactions on Services Computing, 2022, 15(1): 188-201.
- [5] YANG S, LI F, TRAJANOVSKI S, et al. Recent advances of resource allocation in network function virtualization[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(2): 295-314.
- [6] PHAM C, TRAN N H, REN S L, et al. Traffic-aware and energy-efficient VNF placement for service chaining: joint sampling and matching approach[J]. IEEE Transactions on Services Computing, 2020, 13(1): 172-185.
- [7] ZHANG Q, QIU X H, ZHU X R. A novel resource optimization algorithm for dynamic networks combined with NFV and SDN[C]//International Conference on Wireless and Satellite Systems. Berlin: Springer, 2019: 283-296.
- [8] FREITAS B D, SANTOS G L, GONÇALVES G, et al. Optimizing NFV placement for distributing micro-data centers in cellular networks[J]. The Journal of Supercomputing, 2021, 77(8): 8995-9019.
- [9] SOUALAH O, MECHTRI M, GHRIBI C, et al. An efficient algorithm for virtual network function placement and chaining[C]//Proceedings of IEEE Annual Consumer Communications & Networking Conference. Piscataway: IEEE Press, 2017: 647-652.
- [10] 袁泉, 游伟, 季新生, 等. 虚拟网络功能资源容量自适应调整方法[J]. 电子与信息学报, 2021, 43(7): 1841-1848.
- YUAN Q, YOU W, JI X S, et al. Adaptive scaling of virtualized network function resource capacity[J]. Journal of Electronics & Information Technology, 2021, 43(7): 1841-1848.
- [11] CHEN J, CHEN J, ZHANG H K. 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.
- [12] 陈康, 向勇, 喻超. 大数据时代机器学习的新趋势[J]. 电信科学, 2012, 28(12): 77-85.
- CHEN K, XIANG Y, YU C. New trend of machine learning in the age of big data[J]. Telecommunications Science, 2012, 28(12): 77-85.
- [13] 郭丽丽, 丁世飞. 深度学习研究进展[J]. 计算机科学, 2015, 42(5): 28-33.
- GUO L L, DING S F. Research progress on deep learning[J]. Computer Science, 2015, 42(5): 28-33.
- [14] 周传鑫, 孙奕, 汪德刚, 等. 联邦学习研究综述[J]. 网络与信息安全学报, 2021, 7(5): 77-92.
- ZHOU C X, SUN Y, WANG D G, et al. Survey of federated learning research[J]. Chinese Journal of Network and Information Security, 2021, 7(5): 77-92.
- [15] PAULHEIM H. Knowledge graph refinement: a survey of approaches and evaluation methods[J]. Semantic Web, 2016, 8(3): 489-508.
- [16] 刘峤, 李杨, 段宏, 等. 知识图谱构建技术综述[J]. 计算机研究与发展, 2016, 53(3): 582-600.
- LIU Q, LI Y, DUAN H, et al. Knowledge graph construction techniques[J]. Journal of Computer Research and Development, 2016, 53(3): 582-600.
- [17] 庄严, 李国良, 冯建华. 知识库实体对齐技术综述[J]. 计算机研究与发展, 2016, 53(1): 165-192.
- ZHUANG Y, LI G L, FENG J H. A survey on entity alignment of knowledge base[J]. Journal of Computer Research and Development, 2016, 53(1): 165-192.
- [18] MONGE A E, ELKAN C. The field matching problem: algorithms and applications[C]//Proceedings of ACM International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 1996: 267-270.
- [19] NAVARRO G. A guided tour to approximate string matching[J]. ACM Computing Surveys, 2001, 33(1): 31-88.
- [20] MECHTRI M, GHRIBI C, ZEGHLACHE D. A scalable algorithm for the placement of service function chains[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 533-546.
- [21] ALMOHAMAD H A, DUFFUAA S O. A linear programming approach for the weighted graph matching problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, 15(5): 522-525.
- [22] The University of Adelaide. The Internet topology zoo[R]. 2012.
- [23] 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.
- [24] 奥德玛, 杨云飞, 穗志方, 等. 中文医学知识图谱 CMeKG 构建初探[J]. 中文信息学报, 2019, 33(10): 1-9.
- BYAMBASUREN O, YANG Y F, SUI Z F, et al. Preliminary study on the construction of Chinese medical knowledge graph[J]. Journal of Chinese Information Processing, 2019, 33(10): 1-9.

[作者简介]



徐泽汐(1997-), 女, 河南驻马店人, 郑州大学博士生, 主要研究方向为网络虚拟化、NFV 部署技术等。



庄雷(1963-), 女, 山东日照人, 博士, 郑州大学教授, 主要研究方向为网络虚拟化、模型检测等。



张坤丽(1977-), 女, 河南巩义人, 博士, 郑州大学讲师, 主要研究方向为人工智能、自然语言处理。



桂明宇(2000-), 男, 河南长葛人, 郑州大学硕士生, 主要研究方向为知识图谱融合、实体对齐、知识图谱补全、链接预测等。