

刘岸鑫, 程良伦, 王涛. 基于模糊图神经网络的工业互联网服务组件编排方法[J]. 广东工业大学学报, 2025, 42(5): 96-104. doi: 10.12052/gdutxb.240068.
Liu Anxin, Cheng Lianglun, Wang Tao. Fuzzy Graph Neural Network-based Industrial Internet Service Component Orchestration Method[J]. Journal of Guangdong University of Technology, 2025, 42(5): 96-104. doi: 10.12052/gdutxb.240068.

基于模糊图神经网络的工业互联网服务 组件编排方法

刘岸鑫¹, 程良伦¹, 王涛²

(1. 广东工业大学 计算机学院, 广东 广州 510006; 2. 广东工业大学 自动化学院, 广东 广州 510006)

摘要: 工业互联网环境工业软件系统具有规模庞大、涵盖的软件组件与设备实体繁杂异构等特点, 在异构设备与软件对象统一的服务化组件化封装基础上, 通过服务组件编排可实现多样化应用工业软件系统灵活敏捷构建, 但对服务编排效率与服务质量保障方面具有较高要求。为提升工业互联网环境服务组件编排效率并保障服务质量, 本文提出基于图神经网络与模糊理论的服务组件编排方法FGraphSAGE_GA, 将工业互联网服务组件编排问题形式化建模为图结构中的链路预测问题, 并采用监督学习方法实现预测模型训练。针对新的服务组件编排问题, 训练后的模型进行概率预测缩小候选空间, 随后使用遗传算法优化服务组件编排方案。在两个不同的数据集上, 针对3种规模的服务组件编排问题开展实验与分析, 证明了所提出的FGraphSAGE_GA算法在服务质量和编排效率方面具有良好性能。

关键词: 工业互联网; 服务组件编排; 图神经网络; 模糊系统; 遗传算法

中图分类号: TP393

文献标志码: A

文章编号: 1007-7162(2025)05-0096-09

Fuzzy Graph Neural Network-based Industrial Internet Service Component Orchestration Method

Liu Anxin¹, Cheng Lianglun¹, Wang Tao²

(1. School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China; 2. School of Automation, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: In industrial internet environments, industrial software systems are usually large-scale and with a complex and heterogeneous array of software systems and physical devices. Based on the unified service-oriented and component-based encapsulation of heterogeneous devices and software objects, diverse applications of industrial software systems can be flexibly constructed through the orchestration of service components. However, it usually requires high efficiency of service orchestration and the assurance of service quality. To enhance the efficiency of service component orchestration and ensure service quality in industrial internet environments, this paper proposes a service component orchestration method called FGraphSAGE_GA, which formalizes the convert the component orchestration problem in industrial internet environments into a link prediction problem within a graph structure. It employs a supervised learning approach to train the prediction model. For the new service component orchestration problem, the trained model performs probabilistic predictions to narrow down the candidate space, followed by the optimization of the service component orchestration plan using a genetic algorithm. Experiments were conducted on two different datasets for service component orchestration problems of three different scales, and experimental results show that the proposed FGraphSAGE_GA algorithm has good performance in terms of service quality and orchestration efficiency.

收稿日期: 2024-05-15 录用日期: 2024-07-25 网络首发日期: 2024-09-30

基金项目: 国家自然科学基金联合基金资助集成项目(U20A6003)

作者简介: 刘岸鑫(1997-), 男, 硕士研究生, 主要研究方向为工业软件, E-mail: liuanxin10337@163.com

通信作者: 程良伦(1964-), 男, 教授, 博士, 主要研究方向为信息物理融合系统, E-mail: llcheng@gdut.edu.cn

Key words: industrial internet; service component orchestration; graph neural network; fuzzy system; genetic algorithm

工业互联网(Industrial Internet)是新一代信息技术与工业经济深度融合的新型基础设施、应用模式和工业生态,通过对人、机、物、系统等繁杂要素的全面连接,构建一个支持物理过程-信息过程深度融合与动态交互的新型智能系统,实现对复杂工业和城市系统的物理过程数据感知、智能计算、精准控制与信息服务。工业互联网环境具有规模庞大、涵盖的软件组件与设备实体繁杂异构、跨平台、跨网络等典型特征,对繁杂异构软硬件对象(传感器、计算节点、控制器、工业设备等)进行组件化服务化封装形成统一规范的Restful web服务组件,并能根据多样性业务应用需求对服务组件进行灵活高效编排组合,优化服务组件组合,实现面向服务的工业互联网智能应用系统快速构建开发,这将对提升工业互联网环境智能应用构建开发效率有重要价值。

在工业互联网服务组件编排架构研究方面,Ghosh等^[1]提出边缘计算系统服务编排架构,由防火墙、SDN控制器、编排实体和应用程序服务器组成。Muñoz等^[2]提出一个5G场景中的MEC(Multi-Access Edge Computing)平台服务编排方案,满足云边端高灵活性和低延迟。Shekhar等^[3]提出动态资源管理中间件,满足移动性、延迟和功耗要求,在边缘/雾计算侧解决了图像分类优化问题的服务编排。Khansari等^[4]提出一个以云为中心的IoT架构,然而,该机制基于每个请求执行优化服务编排组合。Gary White等^[5]提出一种目标导向服务组合的可靠系统自我管理解决方案,探索大规模服务组件编排过程中的组件关系挖掘与利用问题。

在工业互联网服务组件编排算法方面:启发式算法曾是研究的热点,如EESHHO算法^[6]基于进化策略的基本原则,融入了精英进化策略(Elite Evolutionary Strategy, EES)与哈里斯鹰算法(Harris Hawks Optimization, HHO),有效地加快了收敛速度并提升了避免陷入局部最优的能力;MAACS算法^[7]通过搜索分解和实时种群重置策略来降低问题的复杂性;Etchiali等^[8]设计了蝙蝠算法,分两阶段递进式寻求动态环境下可行解,效率方面优于两种优化算法。启发式方法在合理的时间限制内寻找近似的最优解,但在面对大规模数据时,服务质量和运行时间两者之间难以平衡。随着深度学习的迅猛发展,越来越多的问题可以通过训练合适的神经网络得以解

决。例如,Wang等^[9]提出了一种基于Q-learning的组合算法,并在数年后推出了该算法的改进版本^[10],采用基于长短期记忆(Long Short-Term Memory, LSTM)网络的深度Q网络(Deep Q-Network, DQN)解决组合优化问题。Elsayed等^[11]探索遗传算法与Q-learning的结合应用,通过Q-learning算法来初始化遗传算法的种群。Yi等^[12]将预训练与基于策略梯度的强化学习结合起来(Pretraining-and-Policy-Based Deep Reinforcement Learning, PPDRL),通过迭代寻找近似最优服务组合方案。Wang等^[13]结合GNN(Graph Neural Networks)、指针网络和WOA(Whale Optimization Algorithm)算法对基于QoS(Quality of Service)的服务组合问题进行研究。Wei等^[14]提出的FGNN方法,在模糊关系小样本学习问题上具有高效性。面对工业互联网服务规模大、服务间依赖关系复杂和服务质量要求高等挑战,如何充分发挥图神经网络在关系挖掘方面的优势,实现高效稳定服务组件编排依然是研究关注点。

为实现工业互联网环境面向多样性应用需求的服务组件高效编排与服务质量优化,本文提出基于图神经网络与模糊理论结合的服务组件编排方法(FGraphSAGE_GA)。提出一个面向多样性工业互联网应用的QoS感知的服务组件编排方法,将工业互联网服务组件编排问题形式化建模为图结构数据中链路预测问题并用监督学习方法学习约束图数据;针对新应用需求的服务组件编排,使用训练后的模型进行概率预测缩小候选空间,利用遗传算法优化服务组件编排方案,在保证服务质量的前提下实现快速服务组件编排,为工业互联网环境多样性应用的高效灵活构建提供技术支撑。

1 问题定义

图1展示了工业互联网服务组件编排架构,底层是物理过程接入层,包括传感器、边缘计算控制设备和生产制造设备等。中间平台层的服务组件池中有海量软件服务组件,包括工业生产过程管控所需的感知节点、计算节点、控制节点。平台借助请求管理器、发现引擎、服务分配器、SLA(Service Level Agreement)监控器等实现服务动态部署、监控等。顶层是应用层,根据不同业务场景进行服务的编排组合,以满足用户实际需求。底层硬件设备通过实体映

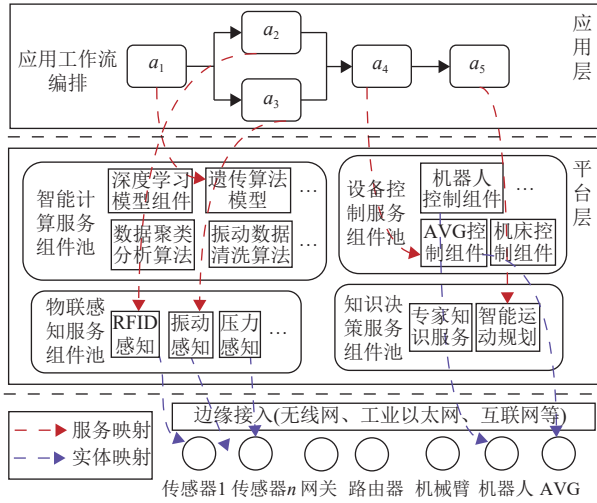


图1 工业互联网服务组件编排架构

Fig.1 Architecture of Industrial Internet Service Orchestration

射为服务,平台集成有各种封装的节点作为服务,应用层根据应用工作流调取映射服务进行编排实现工业需求。

服务组件编排算法的目标是在尽可能少的时间内根据用户任务构建一个服务质量最优的服务编排方案。用户任务包含所需抽象服务的标识符以及这些服务之间的工作流结构,要在每个候选服务集合中选取一个服务,以便组合成一个综合性能服务质量(QoS)接近最佳的整体服务。

工业软件服务组件 s 是指服务组件池中可用的具体服务。一个工业软件服务组件可以被定义为一个四元组: $s = (\text{ID}, \text{In_P}, \text{Out_P}, \text{QoS})$,其中, ID是服务的唯一标识符; In_P和Out_P分别表示服务的输入参数和输出; QoS是服务的服务质量属性,它由多个不同的服务质量指标组成,本文选取以下四种:响应时间(Response Time) q_d 、吞吐量(Throughput) q_t 、可靠性(Reliability) q_r 和可用性(Availability) q_a 。本文通过对各指标进行加权来计算服务质量:

$$\text{QoS} = \delta_1 q_d + \delta_2 q_t + \delta_3 q_r + \delta_4 q_a \quad (1)$$

式中: δ_i 是第 i 个服务质量所占权重,且 $\sum_{i=1}^4 \delta_i = 1$ 。

抽象服务 a_i 是为了研究方便设定的,它是提供特定业务功能服务的代称。用户不需要了解软件系统有哪些具体服务,他们只需要从任务角度出发,用户任务 \mathcal{T} 可以被分解为子任务 $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$, t_i 任务所需的具体服务类别就是抽象服务 a_i 。

候选服务集是从工业软件系统角度定义的名词,也可以表示为 a_i 。抽象服务是构建系统所需的具体业务功能的服务代表,也就是候选服务集,而实际

服务池中可能存在多个提供此功能的服务,这些服务所属的集合称为候选服务集。 $a_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,n_i}\}$,其中 $s_{i,j}$ 是对应任务 t_i 的第 j 个具体服务, n_i 是 a_i 的候选服务数量。因此,任务的解决方案 $\text{sol} = \{s_1, s_2, \dots, s_n\}$,其中 s_i 是从 a_i 中选择的具体服务。

目标函数 F 用于评估服务解决方案的质量。研究的目的是最小化或最大化目标函数的值,而最大化问题和最小化问题可以相互转化,这里选取最大化问题。除了目标函数外,服务解决方案还必须满足一系列的约束条件 C 。约束条件是场景对于任务的限制,这些约束可能包括执行时间、可靠性和资源使用等方面的限制。一个解决方案必须满足所有约束才能被认为是可行的。用 $C = \{c_1, c_2, \dots, c_p\}$ 表示约束集合,其中 c_p 是第 p 个约束。如果关注最大化非功能属性,最小化总体时间成本,目标函数可以定义为

$$\begin{aligned} & \max \sum_{i=1}^n \sum_{j=1}^{k_i} \text{QoS}(s_{ij}), \\ & \min T, \\ & \text{s.t.} \begin{cases} x_{ij} \in \{0, 1\}, i \in \{1, 2, \dots, k\}, j \in \{1, 2, \dots, k_i\} \\ \sum_{j=1}^{k_i} x_{ij} = 1, \forall i \in \{1, 2, \dots, k\} \\ c_i = q_l < k_i \end{cases} \quad (2) \end{aligned}$$

在公式(2)中, QoS表示解决方案的整体服务质量。服务组件编排问题的第二个优化目标中, T 表示算法的运行时间。 n 是组合服务中抽象服务的总数, k_i 是抽象服务 a_i 对应的候选服务集 S_i 中的具体服务数量。 $x_{i,j}$ 是一个二元变量,表示是否选择候选服务集 C_i 中的具体服务 $s_{i,j}$ 。约束条件确保每个抽象服务 a_i 只有一个对应的具体服务被选择, l 是第 l 个服务质量指标, k_i 是 c_i 的具体限制值。

2 FGraphSAGE_GA模型

如图2所示,本文所提出的FGraphSAGE_GA算法框架由初始化模块、模糊图神经网络模块和遗传算法模块组成。其中初始化模块用任务和服务分别构建图数据模型;模糊图神经网络模块提出了算法具体方法;遗传算法模块在图神经网络方法预测的基础上对编排方案进一步优化。

2.1 初始化模块

首先,对这些任务和解决方案进行预处理,包括从服务组件编排任务中提取功能(子任务)和非功能(约束)属性,形成约束图;同时,从服务解决方案中提

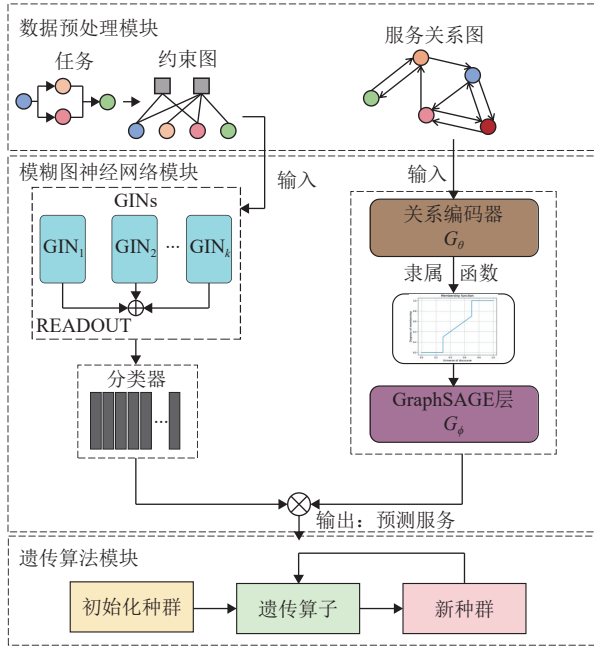


图2 FGraphSAGE_GA模型框架

Fig.2 Framework of FGraphSAGE_GA model

取服务共现关系,形成服务关系图。然后,将服务关系图输入到候选服务缩减模型中,并使用服务组件编排任务(即训练样本)和服务解决方案(即训练标签)来训练模型。

2.1.1 约束图

约束图将用户任务的功能和非功能属性转化为图的形式,以便更好地处理服务组件编排问题。为了减少图冗余,保持图紧凑性,采取以下2个规则:

- (1) 如果两个约束条件 c_i 和 c_j 约束相同的子任务集,那么它们可以被合并为一个节点。
- (2) 如果一个约束条件 c_i 只针对一个特定的子任务 t ,那么该约束和子任务可以合并为一个节点。

对于节点特征,参考Multi-hot编码表示为

$$\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2] = [k, q_{\min}, q_{\max}] \in \mathbb{R}^{1+2|QoS|} \quad (3)$$

式中: \mathbf{n}_1 为0表示这是一个约束节点,否则表示子任务的索引; \mathbf{n}_2 表示QoS属性编码; q_{\min}, q_{\max} 分别表示约束条件在各QoS属性上的最小值和最大值。

2.1.2 服务关系图

服务类别特征是用于区分不同服务的一个基本属性,方式编码与约束图类似。每个服务 s_i 分配一个唯一的类别标签 c_i ,该标签是一个整数,可以根据服务的功能和特性进行分配,用向量 \mathbf{a}_1 来表示。对于服务 s_i ,定义了一个QoS向量 \mathbf{a}_2 ,其中包含了该服务的所有QoS属性值。综合服务类别和QoS属性,得到了服务的完整特征表示:

$$s_i = [\mathbf{a}_1, \mathbf{a}_2] \in \mathbb{R}^{1+|QoS|} \quad (4)$$

利用上述构建的服务特征,可以构建服务关系图(Service Relation Graph, SRG)。在SRG中,节点代表服务,边代表服务之间的共现关系。构建SRG的过程首先计算每个服务 s_i 出现的次数 n_i ,以及每对服务 s_i 和 s_j 共现的次数 $n_{i,j}$ 。边的权重由条件概率 $p(s_j|s_i)$ 给出,表示在服务 s_i 被使用的情况下,服务 s_j 也被使用的概率。这个概率可以通过公式(5)计算:

$$p(s_j|s_i) = \frac{n_{i,j}}{n_i} \quad (5)$$

这个概率值表示在服务 s_i 被使用的情况下,服务 s_j 被使用的可能性。在图中,这个概率值作为边的权重,从而构建了服务之间的关联关系。例如, s_1 到 s_2 的边权重和 s_2 到 s_1 是不同的,即当 s_1 出现在解决方案中 s_2 紧接着出现的概率与当 s_2 出现在解决方案中 s_1 紧接着出现的概率不同。

2.2 模糊图神经网络模块

2.2.1 图同构网络

在服务组件编排的过程中,历史数据中蕴含的模式和规则对于高效地选择和组合服务至关重要。

经过初始化模块构建了约束图(Constraint Graph),为了从约束图中学习有效的表示,本文采用了图同构网络(Graph Isomorphism Network, GIN)。GIN^[15]是一种强大的图神经网络架构,它能够捕捉图中节点的拓扑结构和节点特征。GIN的核心思想是通过一系列图卷积层来更新节点的特征表示,其中每一层都会考虑节点自身及其邻居的特征。

GIN的更新规则如下:

$$\mathbf{z}_i^{(k+1)} = h_{\theta} \left(\alpha \mathbf{z}_i^{(k)} + \beta \sum_{j \in \mathcal{N}(i)} \mathbf{z}_j^{(k)} \right) \quad (6)$$

式中: $\mathbf{z}_i^{(k)}$ 是第 k 层中节点 i 的特征向量, $\mathcal{N}(i)$ 是节点 i 邻居节点集合, h_{θ} 代表一个全连接神经网络,用于进一步处理节点特征。 α 、 β 是标量参数,用于平衡节点自身特征与其邻居特征的聚合结果。

求和聚合节点特征并获得图表示向量:

$$\mathbf{z}_G = \sum_{i \in V} \mathbf{z}_i \quad (7)$$

通过GIN,可以学习到每个子任务的嵌入表示,进而捕捉任务中的功能和非功能属性之间的关系。这些学习到的表示随后被用于预测每个服务在构建服务解决方案时的使用概率。

2.2.2 FGraphSAGE

该部分构建了由关系编码器、隶属函数、GraphSAGE

组成的FGraphSAGE框架。

(1) 关系编码器: 关系编码器是FGNN中的核心组成之一, 其主要作用是计算样本间的相似度得分, 并通过迭代更新边标签来构建图结构。它通过对输入的SRG进行边和节点学习来输出初步学习的关系表征, 然后隶属函数将关系表示转换为图中的强关系归纳偏差进行节点分类。区别于一般的GNN算法, 关系编码器充分利用初始化阶段构建的服务关系图, 通过节点对的方式对边特征条件概率进行学习。

在关系编码器的每一层, 首先通过边更新函数 f_e^l 来更新边特征 e_{ij} , 公式为

$$e_{ij}^{l+1} = f_e^l(v_i^l, v_j^l) \quad (8)$$

式中: v_i, v_j 分别是边两端节点的特征。接着, 通过节点更新函数 f_v^l 来更新节点特征 v_i^l , 公式为

$$v_i^{l+1} = f_v^l \left(\frac{\sum_j e_{ij}^{l-1} v_j^{l-1}}{\sum_j e_{ij}^{l-1}}, v_i^l \right) \quad (9)$$

式中: $\sum_j e_{ij}^{l-1} v_j^{l-1}$ 表示积累通过边特征加权后的邻居节点特征。

损失计算: 通过计算边预测的损失 L 来优化关系编码器, 公式为

$$L(\theta) = \frac{1}{|E|} \sum_{i,j} l_e(e_{ij}^l, \hat{y}_{ij}; \theta) \quad (10)$$

式中: L_1 是关系编码器中的层数。边预测的真实值定义为

$$\hat{y}_{ij} = \begin{cases} 0, & \text{if } y_i = y_j \\ 1, & \text{if } y_i \neq y_j \end{cases} \quad (11)$$

如果两个节点 i, j 属于同一个类别, 即它们的类别标签 y_i, y_j 相等, 那么边预测的真值 \hat{y}_{ij} 被定义为0, 这表示边连接的是相似的节点, 即它们之间存在较强的关系或它们属于同一个组或类别; 反之被定义为1, 表示边连接的是不同的类别, 即它们之间存在较弱的关系或它们属于不同的组或类别。

(2) 隶属函数: 隶属函数见图3。通过模糊隶属函数 μ 将关系表示映射到 $[0,1]$ 区间, 以生成优化后的图结构, 公式为

$$\mu(e_{ij}) = \begin{cases} 1, & \text{if } e_{ij} > \zeta \\ f_\mu(e_{ij}), & \text{if } \zeta \geq e_{ij} \geq \eta \\ 0, & \text{if } e_{ij} < \eta \end{cases} \quad (12)$$

式中: ζ 和 η 是设定的阈值, f_μ 是模糊隶属函数的具体形式, 展示了在FGNN中用于推断关系表示的隶属函

数设计。考虑到关系编码器的输出具有模糊性, 模型对极其相似的样本间的联系进行了人为加强, 并剪断两个节点之间差异显著的边。对于相似度得分介于 ζ 和 η 之间的关系, 关系编码器难以给出可靠的评分。为了处理这些边缘情况, 引入了线性函数 $f_\mu(e_{ij})$ 。由这些保留下来的边所构成的图不再是完全连接的, 而是具有了强烈的关系归纳偏置。

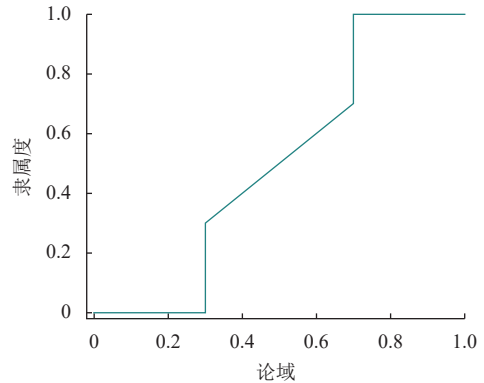


图3 隶属函数

Fig.3 Membership function

(3) GraphSAGE层: GraphSAGE层的核心在于如何有效地更新节点的特征表示, 将关系编码器和模糊隶属函数的输出作为输入, 根据SRG边存储的节点 i, j 之间的连接关系, 在第 k 层的更新函数中, 节点特征的更新公式为

$$u_i^{(k+1)} = \psi \text{Aggregate}_{s_j \in \mathcal{N}(s_i) \cup \{s_i\}} \frac{p(s_j|s_i)}{\sqrt{d_i d_j}} u_i^{(k)} \quad (13)$$

式中: $d_i = 1 + \sum_{s_j \in \mathcal{N}(s_i)} p(s_j|s_i)$, ψ 是可训练参数矩阵, Aggregate是聚合函数。

损失 L 通过交叉熵损失来优化, 公式为

$$L(\phi) = \sum_i l_v(v_i^{L_2}, y_i; \phi) \quad (14)$$

式中: L_2 是关系编码器中的层数。

整体损失函数是在训练过程中同时优化节点的分类准确性和边的预测准确性, 从而提高模型在少量样本学习任务中的性能。

$$L([\theta, \phi]) = - \sum_{k=1}^{|\mathcal{C}|} y_k \log(P(Y^* = y_k)) \quad (15)$$

式中: θ, ϕ 分别是关系编码器和GraphSAGE层的参数, $|\mathcal{C}|$ 是子任务对应候选服务集大小, P 是节点 Y^* 属于类别 y_k 的概率。

算法1 FGraphSAGE算法

- (1) 输入服务关系图 $G=(\mathcal{V}, \mathcal{E})$, 任务 \mathcal{T} , 学习率 γ, ε , 权重 λ
- (2) 随机初始化关系编码器和GraphSAGE层

- (3) 对于训练迭代次数进行以下操作:
- (a) 采样任务 t 和对应解决方案 sol 。
 - (b) 对于关系编码器的每一层:
 - i. 更新边特征: $\mathcal{E}' \leftarrow \text{EdgeUpdate}(\mathcal{V}^{l-1})$;
 - ii. 更新节点特征: $\mathcal{V}' \leftarrow \text{NodeUpdate}(\mathcal{E}', \mathcal{V}^{l-1})$ 。
 - (c) 根据公式(9)评估关系编码器的损失 L' 。
 - (d) 根据公式(11)隶属函数进行边 \mathcal{E} 转换。
 - (e) 对临时迭代次数:
 - i. 对GraphSAGE层每一层:

更新节点特征: $\mathcal{V}' \leftarrow \text{NodeUpdate}(\mathcal{E}', \mathcal{V}^{l-1})$;
 - ii. 根据公式(13)评估GraphSAGE层损失 L ;
 - iii. 梯度下降优化GraphSAGE层参数 ϕ : $\phi \leftarrow \phi - \gamma \nabla L(\phi)$;
 - (f) 根据公式(14)评估整体损失 $L([\theta, \phi])$;
 - (g) 梯度下降优化整体损失: $[\theta, \phi] \leftarrow [\theta, \phi] - \gamma \nabla (L([\theta, \phi]) + \lambda L')$ 。
- (4) 输出: 关系编码参数 θ , GraphSAGE参数 ϕ 。

2.2.3 分类学习

利用学习到的节点表示和任务部分的输出 z_G , 可以预测每个服务在特定任务中被使用的概率。

$$\hat{y} = \sigma(Uz_G) \quad (16)$$

为了训练GraphSAGE模型, 需要定义一个损失函数来衡量预测概率和真实标签之间的差异。对于多标签分类问题, 通常使用二元交叉熵损失函数:

$$\mathcal{L} = -\frac{1}{|S|} \sum_{i=1}^{|S|} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (17)$$

式中: $|S|$ 是节点的总数, y_i 是节点 s_i 是否被使用的标签(取值为1或0), \hat{y}_i 是模型预测的概率。通过最小化损失函数, 可以调整模型参数, 使得预测概率更接近真实的标签分布。

模型参数的优化通常通过随机梯度下降(Stochastic Gradient Descent, SGD)来实现。在每次迭代中, 计算损失函数关于模型参数的梯度, 并更新参数以减少损失。

2.3 遗传算法模块

初始阶段, 通过整数编码方式创建染色体。对于一个由 n 个抽象服务组成的服务组件编排问题, 每个抽象服务都关联一个包含 k_i 候选服务的集合, i 的范围是1到 n 。每个整数范围是 $[0, k_i - 1]$, 这些整数标识了在候选服务集中被选定的具体服务的序号。

其次, 将组合服务的服务质量(QoS)作为适应度函数。适应度函数的值越大, 表示个体越接近最优解。

初始种群创建个体染色体的编码时, 根据模糊图神经网络模型预测概率来选择服务, 具有较高预测概率的服务更有可能被选为染色体的一部分。这

种方法不仅确保了遗传算法从较好的起点开始优化, 提高了优化效率; 同时每个服务都有机会被选中也保持了算法的多样性。

随后, 本文采用传统的二进制锦标赛选择机制来筛选参与后续的交叉与变异过程的染色体。在执行交叉时, 引入均匀交叉策略, 确保每对父代染色体有50%的几率进行基因片段的交换。在变异阶段, 则运用均匀变异策略, 导致每个新一代个体的染色体在某一位置上发生改变, 转变为代表同一候选服务集中另一具体服务的编号。最后, 采用精英保留策略, 将新生成的子代群体与原有的父代群体合并, 以此确保最优秀个体的保留。

3 实验

本文将提出的方法在2个数据集上进行实验, 并与其他基准方法进行对比, 并通过参数调优和消融实验验证该模型的性能。

3.1 数据集与衡量指标

此次实验的数据基于服务计算领域常见的Quality of Web Service(QWS)公开数据集来构建实验数据集。QWS基于实际公开数据, 收集了服务描述和属性信息, 包括服务的名称、类型、功能描述以及QoS指标等。

本文参考文献[16]数据处理方法, 将QWS数据集加工为两种数据集分别进行验证。第一种: QWS数据集经过文本聚类处理, 形成了QWS_grouped数据集。通过这一方法, 所有Web服务名称被分成250个聚类结果, 每个结果被视为一个潜在服务集, 代表了一种抽象服务。这些服务集的规模在2~128不等。第二种: 扩展数据集QWS_enlarged。考虑到QWS数据集中Web服务数量的限制, 为避免实验结果受到数据分布的限制, 采用了服务计算领域内通用的数据集扩展方法。这些策略依赖于QWS中服务质量属性的取值范围, 随机生成了68 430个虚拟服务, 并将它们随机分配到每个潜在服务集中。其中包含了5 320个潜在服务集, 每个服务集的规模在2~128之间, 模拟工业互联网服务动态变化环境和大规模服务环境。

通过以上步骤, 本文构建了一个全面的数据集, 在本次实验中, 服务组件编排用时和解决方案质量是所有算法的共同衡量标准。解决方案质量的比较可以通过公式(18)来量化:

$$\text{sq}(\text{sol}) = \frac{F(\text{sol})}{F(\text{sol}^*)} \quad (18)$$

式中: sol^* 表示最佳解决方案, 其质量为1, 其余解质量均在0到1之间。sq值越高, 服务质量越好。此外, 为了进行参数校准实验, 本文首先定义了 $p@k$ 指标, 这是一个用于评估极端多标签分类问题的常用指标。具体地, $p@k$ 指标计算了模型预测为最高概率的 k 个标签中, 实际正确的标签数量的比例。数学上, $p@k$ 可以表示为

$$p@k = \frac{1}{k} \sum_{l \in \text{rank}_k} y_l \quad (19)$$

式中: rank_k 表示预测概率最高的 k 个标签的集合, y_l 是第 l 个标签的实际值。当 k 等于1时, $p@1$ 实际上就是准确率(accuracy)。

3.2 基准实验

本节对提出的服务质量感知的服务组件编排方法进行了全面的比较分析, 选取了两个类别的共三种方法作为对比, 分别为启发式算法(遗传算法(GA)结合天鹰策略和鲸鱼优化算法(ESWOA)^[17]和基于学习的方法(深度强化学习算法PPDRL^[13])。

3.2.1 算法有效性

在不同规模下验证本文算法的有效性, 问题的规模是根据抽象服务的数量来定义的, 本文在两个不同的数据集上生成了包含10个、50个和100个抽象服务的测试集。

如图4、图5所示, 在较小规模的服务组件编排问题中, FGraphSAGE_GA和PPDRL算法能够保持较高的服务质量, 相比于其他算法优势不够明显。这一现象可以归因于数据集的有限规模, 包含的服务数量较少, 导致服务组件编排任务的复杂度相对较低, 各类算法均能够有效地识别出高服务质量的服务组件编排。但随着抽象服务数量的增加, FGraphSAGE_GA和PPDRL算法开始与其他算法在性能上拉开差距。在规模更大的QWS_enlarged数据集中, 尽管PPDRL算法性能下降速度增加显著, 总体来说, 基于学习的方法效率明显优于启发式算法。本文通过构

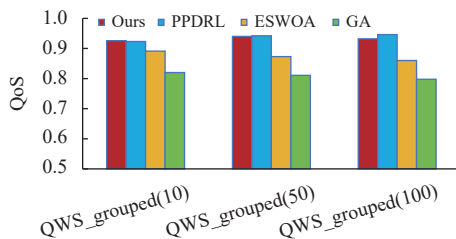


图4 4种算法在QWS_grouped上对比结果

Fig.4 Comparative results of four algorithms on the QWS_grouped dataset

建2种不同规模的数据集, 成功模拟了不同复杂度的服务组件编排场景。

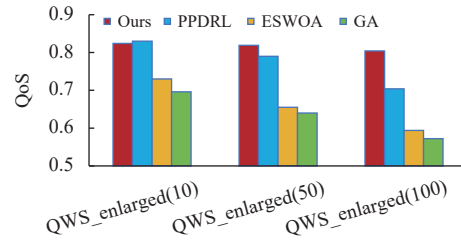


图5 4种算法在QWS_enlarged上对比结果

Fig.5 Comparative results of four algorithms on the QWS_enlarged dataset

根据实验结果分析, FGraphSAGE_GA算法相较于PPDRL算法QoS提升了2.63%, 与ESWOA算法相比提升了15.54%, 与GA相比提升了22.09%。说明基于图学习的方法在挖掘潜在拓扑关系方面天然的优势, 可以有效利用历史组合的经验, 提升服务编排的质量。此外, 实验结果还揭示了一个趋势: 随着服务组件编排问题中涉及的抽象服务数量增加, 各种算法实现的优化QoS值呈现下降趋势, FGraphSAGE_GA算法在这种情况下表现出相当的稳定性。综上所述, FGraphSAGE_GA算法在服务组件编排问题上的有效性得到了充分的验证。

3.2.2 算法高效性

为了确认所提出方法的效率, 通过实验追踪并记录了各算法在生成服务质量最优的组合服务过程中的时间消耗(以s为单位)。

实验结果如表1、表2所示, 服务组件编排的规模是影响算法效率的关键因素, 在较为简单的数据集一(QWS_grouped)中, 所有算法的处理时间普遍较短; 而在数据集二(QWS_enlarged)中, 随着服务数量的增加, 各算法的处理时间相应延长。

表1 4种算法在QWS_grouped上高效性实验结果

Table 1 Efficiency experimental results of four algorithms on the QWS_grouped

算法	10节点		50节点		100节点	
	服务质量	时间/s	服务质量	时间/s	服务质量	时间/s
Ours	0.926	0.600	0.940	11.240	0.932	36.537
PPDRL	0.923	7.365	0.942	102.600	0.946	279.800
ESWOA	0.891	1.562	0.873	31.669	0.860	68.397
GA	0.820	0.929	0.811	29.460	0.798	100.135

在处理包含较少抽象服务的数据集一时, FGraphSAGE_GA算法展现出了较低的时间消耗, 并且随着服务数量的增长, 其时间消耗的增幅相对较小。在面对更大规模的组合问题时, FGraphSAGE_GA

算法依然保持了其时间效率上的优势,ESWOA和GA算法的效率位于中等水平,而与FGraphSAGE_GA服务质量最为接近的PPDRL算法耗时始终居高不下,在数据集二最大节点情况下甚至接近本文算法的10倍。

表2 4种算法在QWS_enlarged上高效性实验结果
Table 2 Efficiency experimental results of four algorithms on the QWS_enlarged

算法	10节点		50节点		100节点	
	服务质量	时间/s	服务质量	时间/s	服务质量	时间/s
Ours	0.824	2.450	0.819	15.068	0.804	32.989
PPDRL	0.830	16.860	0.790	130.500	0.704	309.800
ESWOA	0.730	6.005	0.655	37.577	0.594	87.781
GA	0.696	5.407	0.640	39.891	0.572	105.640

通过对比分析,FGraphSAGE_GA算法在减少时间消耗方面实现了显著的改进,相较于GA减少了57.74%,相较于ESWOA算法减少了约59.03%,相较于PPDRL算法减少了约88.52%。这表明,相比于在线训练的常规方法,离线预训练的FGraphSAGE模型可以在不影响服务组件编排效果的前提下有效减少时间消耗,表明FGraphSAGE_GA算法在提升服务组件编排效率方面的优越性。

综上所述,实验结果表明,本文的方法在解决方案质量上具有显著优势,并且在组合时间上也表现出较高的效率。这些结果突出了本文方法在快速构建高质量服务解决方案方面的有效性。此外,还讨论了每种方法的优缺点,为未来的研究提供了方向,例如进一步探索更高效的算法结构,或者将服务组件编排方法扩展到更广泛的应用场景中。

3.3 参数k选择

在服务质量感知的服务组件编排中,选择合适的服务数量(top-k)对于减少搜索空间和提高组合效率至关重要。本节通过实验确定最佳的k值,以平衡解决方案的质量。

实验中,本文选取了不同的k值(从2到10),并记录了每个k值下的平均解决方案质量。如图6所示,随着k值的增加,解决方案的质量初期有所提高,但当k值超过某个阈值后,质量的提升趋于平缓。这表明已经选择了足够多的高概率服务,进一步增加k值不会带来显著的效益。

基于这些观察,QWS_grouped数据集选择了k=5,而QWS_enlarged数据集选择了k=10作为各自实验的最佳k值。这一决策平衡了解决方案的质量和搜索空间的大小,为后续的服务组件编排优化奠

定了基础。

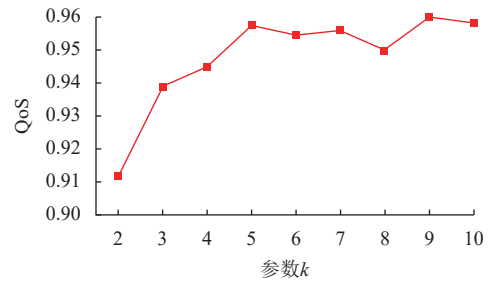


图6 不同k值对方法表现的影响

Fig.6 The performance of the method versus different values of k

3.4 消融实验

3.4.1 带与不带GraphSAGE的模型在候选服务减少任务中的效果对比

本文进行了一系列的消融实验,以验证所提出方法中每个部分的有效性。本文首先比较了是否使用模糊图神经网络(FGraphSAGE)的候选服务减少方法,旨在展示所提出的FGraphSAGE算法在挖掘服务组件编排历史数据中的隐含模式和关系时的能力。本文采用了多标签分类问题中的p@5指标来定量地分析评估不同方法的性能。

实验结果显示,使用FGraphSAGE的方法明显优于不使用FGraphSAGE的方法。这一结果强调了FGraphSAGE在捕捉服务间复杂关系和提高服务选择准确性方面的重要作用。FGraphSAGE的引入使得模型能够更好地理解和利用服务组件编排历史数据中的隐含结构,从而在服务组件编排任务中取得更好的性能。是否有GNN模型的服务质量对比如图7所示。

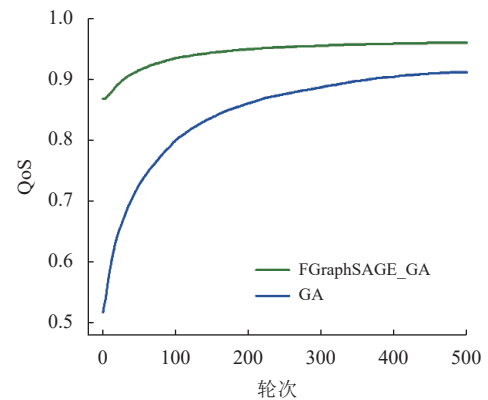


图7 是否有GNN模型的服务质量对比

Fig.7 Comparison of service quality with and without GNN

3.4.2 FGraphSAGE各模块作用分析

该实验旨在评估FGraphSAGE部分算法中各个组件对整体性能的贡献。在规模较大的数据集二上,通过逐一移除或修改关键组件,可以观察到这些变

化对模型性能的具体影响,从而深入理解各个组件的作用。本节设计了4种不同的网络结构,分别是移除GraphSAGE层 $[\theta]$ 、移除关系编码器 $[\phi]$ 、移除隶属函数 $[\theta; \phi]$ 、移除关系编码和隶属函数 $[\phi]$ 以及包含三者的完整算法 $[\theta; \phi]$ 。

从表3中可以看出,相比于完整算法,移除GraphSAGE层对整体模型效果影响很大,而随着抽象节点数增加,模型整体效果衰减很快。因为GraphSAGE层消失后,隶属函数也无法发挥作用,相当于FGraphSAGE算法只剩关系编码器在工作。但值得注意的是,尽管如此,整体方法效果依然表现出相比于其他方法的有效性,侧面证明图方法在服务组件编排中的有效性。引入关系编码器和隶属函数的完整算法 $[\theta; \phi]$ 相比于 $[\phi]$ 在三种节点任务中服务质量明显提升。这说明,通过模糊理论设计的隶属度函数能够有效地处理样本间的模糊关系,生成更优的关系归纳偏差,从而提高模型对新任务的适应能力。最后,与使用二进制矩阵而非隶属度函数的模型 $([\theta; \phi])$ 相比,完整模型的性能略有提升,这进一步验证了隶属度函数在处理模糊关系时的有效性。

表3 在QWS_enlarged上消融实验结果

Table 3 Experimental results of ablation algorithms on the QWS_enlarged

模型配置	10节点 服务质量	50节点 服务质量	100节点 服务质量
$[\theta]$	0.800	0.789	0.762
$[\phi]$	0.817	0.816	0.797
$[\theta; \phi]$	0.809	0.798	0.789
$[\phi]$	0.814	0.810	0.792
$[\theta; \phi]$	0.824	0.819	0.804

综上所述,消融实验的结果清晰地展示了FGNN模型中各个组件的贡献,特别是关系编码器和模糊理论的应用。

3.4.3 服务关系图在缩小候选服务中的作用评估

在服务组件编排的研究中,服务关系图(SRG)的构建和应用是提高候选服务筛选准确性的关键步骤。SRG通过分析历史服务组件编排数据,揭示了服务之间的共现关系和相互作用,为服务选择提供了有价值的信息。本节通过实验比较了利用SRG的FGraphSAGE模型与不利用SRG信息的模型在多标签分类任务中的性能。

如图8所示,利用SRG的FGraphSAGE模型在预测精度上显著优于不使用SRG的模型。这一实验突出了SRG在捕捉服务间复杂关系和提升模型预测能力方面的重要作用。通过SRG,模型能够更深入地理

解服务之间的相互依赖性,从而在服务组件编排任务中实现更精准的候选服务筛选。

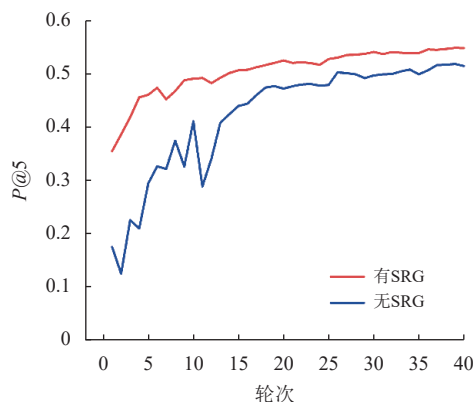


图8 是否有SRG的 $p@5$ 对比

Fig.8 Comparison of $p@5$ with and without SRG

4 结论

针对工业互联网服务规模大、服务间依赖关系复杂和服务质量要求高等特点,为解决服务组件编排的效率和稳定性不足问题,本文提出了FGraphSAGE_GA方法。首先形式化描述服务组件编排问题;然后将任务和服务构建为图数据,将问题建模为图数据中的链路预测问题并用监督的方法学习约束图数据。训练后的模型被用于概率预测缩小候选空间,随后使用遗传算法进一步优化服务组件编排方案。在2个不同数据集、3种不同规模上,证明了所提出的FGraphSAGE_GA方法在服务质量和编排效率以及稳定性方面优势显著。

参考文献:

- [1] GHOSH S, UGWUANYI E, DAGIUKLAS A, *et al.* BlueArch-an implementation of 5G testbed[J]. Journal of Communication, 2019, 14: 1110-1118.
- [2] MUÑOZ R, NADAL L, CASELLAS R, *et al.* The ADRENALINE testbed: an SDN/NFV packet/optical transport network and edge/core cloud platform for end-to-end 5G and IoT services[C]//2017 European Conference on Networks and Communications (EuCNC). Oulu, Finland: IEEE, 2017: 1-5.
- [3] SHEKHAR S, CHHOKRA A, SUN H, *et al.* Supporting fog/edge-based cognitive assistance IoT services for the visually impaired[C]//Proceedings of the International Conference on Internet of Things Design and Implementation. Montreal Quebec. Montreal: ACM, 2019: 275-276.
- [4] KHANSARI M E, SHARIFIAN S, MOTAMEDI S A. Virtual sensor as a service: a new multicriteria QoS-aware cloud service composition for IoT applications[J]. The Journal of Supercomputing, 2018, 74(10): 5485-5512.

(下转第120页)

- [19] HUANG G, LIU Z, VAN DER MAATEN L, *et al.* Densely connected convolutional networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2017: 4700-4708.
- [20] ZHOU L, CAI H, GU J, *et al.* Efficient image super-resolution using vast-receptive-field attention[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 256-272.
- [21] ZHANG Y, LI K, LI K, *et al.* Image super-resolution using very deep residual channel attention networks[C]//Proceedings of the European conference on computer vision (ECCV). Berlin: Springer Verlag, 2018: 286-301.
- [22] WOO S, PARK J, LEE J Y, *et al.* CBAM: convolutional block attention module[C]//Proceedings of the European Conference on Computer Vision (ECCV). Berlin: Springer Verlag, 2018: 3-19.
- [23] FAN C M, LIU T J, LIU K H. Compound multi-branch feature fusion for real image restoration[EB/OL]. arXiv: 2206.02748(2022-06-02)[2024-07-06]. <https://doi.org/10.48550/arXiv.2206.02748>.
- [24] WANG Z, BOVIK A C, SHEIKH H R, *et al.* Image quality assessment: from error visibility to structural similarity[J]. *IEEE Transactions on Image Processing*, 2004, 13(4): 600-612.
- [25] JAGALINGAM P, HEGDE A V. A review of quality metrics for fused image[J]. *Aquatic Procedia*, 2015, 4: 133-142.
- [26] LIN T Y, MAIRE M, BELONGIE S, *et al.* Microsoft coco: common objects in context[C]//Computer Vision—ECCV 2014: 13th European Conference. Zurich: Springer International Publishing, 2014: 740-755.
- [27] NEJATI M, SAMAVI S, SHIRANI S. Multi-focus image fusion using dictionary-based sparse representation[J]. *Information Fusion*, 2015, 25: 72-84.
- [28] FARID M S, MAHMOOD A, AI-MAADEED S A. Multi-focus image fusion using content adaptive blurring[J]. *Information Fusion*, 2019, 45: 96-112.
- [29] LIU Y, CHEN X, PENG H, *et al.* Multi-focus image fusion with a deep convolutional neural network[J]. *Information Fusion*, 2017, 36: 191-207.
- [30] ZHANG Y, LIU Y, SUN P, *et al.* IFCNN: a general image fusion framework based on convolutional neural network[J]. *Information Fusion*, 2020, 54: 99-118.
- [31] MA B, ZHU Y, YIN X, *et al.* SESF-Fuse: an unsupervised deep model for multi-focus image fusion[J]. *Neural Computing and Applications*, 2021, 33: 5793-5804.
- [32] LIU Y, WANG L, LI H, *et al.* Multi-focus image fusion with deep residual learning and focus property detection[J]. *Information Fusion*, 2022, 86: 1-16.
- [33] ZHAI H, ZHENG W, OUYANG Y, *et al.* Multi-focus image fusion via interactive transformer and asymmetric soft sharing[J]. *Engineering Applications of Artificial Intelligence*, 2024, 133: 107967.

(责任编辑: 杨耀辉 英文审核: 熊荣斌)



(上接第104页)

- [5] WHITE G, PALADE A, SIOBHÁN CLARKE. QoS prediction for reliable service composition in IoT[C]//International Conference on Service-Oriented Computing. Hangzhou, China: Springer, 2018: 149-160.
- [6] LI C, LI J, CHEN H, *et al.* Memetic harris hawks optimization: developments and perspectives on project scheduling and QoS-aware web service composition[J]. *Expert Systems with Applications*, 2021, 171: 114529.
- [7] DAHAN F. An effective multi-agent ant colony optimization algorithm for QoS-aware cloud service composition[J]. *IEEE Access*, 2021, 9: 17196-17207.
- [8] ETCHIALI A, HADJILA F, BEKKOUCHE A. An intelligent bat algorithm for web service selection with QoS uncertainty[J]. *Big Data and Cognitive Computing*, 2023, 7(3): 140.
- [9] WANG H, ZHOU X, ZHOU X, *et al.* Adaptive service composition based on reinforcement learning[C]//International conference on service-oriented computing. Francisco, CA, USA: Springer, 2010: 92-107.
- [10] WANG H, GU M, YU Q, *et al.* Large-scale and adaptive service composition using deep reinforcement learning[C]//International Conference on Service-Oriented Computing. Malaga, Spain: Springer, 2017: 383-391.
- [11] ELSAYED D H, NASR E S, ALAA EL DIN M, *et al.* A new hybrid approach using genetic algorithm and Q-learning for QoS-aware web service composition[C]//International Conference on Advanced Intelligent Systems and Informatics. Cairo, Egypt: Springer, 2017: 537-554.
- [12] YI K, YANG J, WANG S, *et al.* PPDDL: a pretraining-and-policy-based deep reinforcement learning approach for QoS-aware service composition[J]. *Security and Communication Networks*, 2022, 2022: 8264423.
- [13] WANG X, XU H, WANG X, *et al.* A graph neural network and pointer network-based approach for QoS-aware service composition[J]. *IEEE Transactions on Services Computing*, 2022, 16(3): 1589-1603.
- [14] WEI T, HOU J, FENG R. Fuzzy graph neural network for few-shot learning[C]//2020 International Joint Conference on Neural Networks (IJCNN). Glasgow, UK: IEEE, 2020: 1-8.
- [15] XU K, HU W, LESKOVEC J, *et al.* How powerful are graph neural networks?[J]. arXiv: 1810.00826(2019-02-22) [2024-05-01]. <https://arxiv.org/pdf/1810.00826>.
- [16] 任笑. 基于图神经网络的服务组合方法的研究与实现[D]. 西安: 西安电子科技大学, 2022.
- [17] GAVVALA S K, JATOTH C, GANGADHARAN G, *et al.* QoS-aware cloud service composition using eagle strategy[J]. *Future Generation Computer Systems*, 2019, 90: 273-290.

(责任编辑: 张玮欣 英文审核: 费伦科)