

基于深度强化学习的 QoS 感知 Web 服务组合

高文斌,王睿,王田丰,祖家琛,胡谷雨
(陆军工程大学 指挥控制工程学院,江苏 南京 210007)

摘要:传统单体式软件架构由于耦合性高、扩展性差的原因,难以适应如今用户需求频繁变动的开发场景。随着服务化理念的深入推广,利用独立的 Web 服务进行组合成为解决这一问题的可行方案。如何利用功能不同、服务质量(Quality of Service, QoS)迥异的 Web 服务,构建出满足用户功能性需求及非功能性需求的组合服务成为服务计算领域的一个研究热点。提出一种基于马尔可夫决策过程的服务组合模型,并设计了基于深度强化学习的求解算法。应用深度网络提升模型表现,可有效解决大规模服务组合场景中现有服务组合算法寻优能力差的问题;进一步针对传统强化学习 Web 服务组合模型中奖励值估计不准确的问题,提出了一种基于卷积神经网络计算奖励值的方法,对服务历史 QoS 信息加以充分利用,并在公共数据集上做了实验验证。实验结果表明,基于深度强化学习的服务组合 ADR-WSC 算法在大规模服务组合问题中输出的组合服务 QoS 更优,算法运行时间更短。

关键词:服务组合;服务质量;深度强化学习;Web 服务;QoS 感知

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2022)06-0092-07

doi:10.3969/j.issn.1673-629X.2022.06.016

QoS-aware Service Composition Based on Deep Reinforcement Learning

GAO Wen-bin, WANG Rui, WANG Tian-feng, ZU Jia-chen, HU Gu-yu

(School of Command and Control Engineering, Army Engineering University of PLA, Nanjing 210007, China)

Abstract: Traditional monolithic software architecture is difficult to adapt to today's development scenarios where user requirements change frequently due to the high coupling and poor scalability. With the further promotion of the concept of servitization, compositing Web-services with independence to develop software has become a feasible solution to this problem. How to use Web-services with different functions and different quality of service (QoS) to build a composite service that meets the functional and non-functional needs of users has become a hot research topic in the field of service computing. A service composition model based on Markov decision process is proposed with a solution algorithm based on deep reinforcement learning. The problem that existing service composition algorithms are difficult to derive the optimal solution for large-scale service combination scenarios can be effectively solved by using deep networks. Furthermore, to address the problem of inaccurate estimation of reward value in traditional reinforcement learning based Web-service composition models, a method based on convolutional network is proposed to calculate the reward value to make full use of the service's historical QoS information. Finally, a performance simulation is done on the public dataset, and the simulation results show that the deep reinforcement learning-based service combination "Adaptive Deep Reinforcement Learning-Web Service Composition" (ADR-WSC) algorithm has higher efficiency in the large-scale service composition problem. It has better performance in terms of running time and combined-QoS

Key words: service composition; quality of service; deep reinforcement learning; Web service; QoS-aware

0 引言

面向服务的计算(service-oriented computing, SOC)和面向服务的架构(service-oriented architecture, SOA)的出现,为如何实现软件模块的解耦提供了新

的思路^[1]。Web 服务具有独立运行、可互操作的特性^[2]。步入云计算时代以后,相当一部分企业以服务的形式在互联网上发布产品供用户使用,致使 Web 服务的数量迅速增长。截至 2020 年 12 月,仅 Program-

收稿日期:2021-06-16

修回日期:2021-10-19

基金项目:国家自然科学基金(62076251)

作者简介:高文斌(1995-),男,硕士研究生,研究方向为服务计算;通信作者:胡谷雨(1963-),男,博士,教授,研究方向为信息网络、网络管理、网络智能化。

ableWeb 网站上所提供的可用 Web 服务数量已经达到了 22 200 个。这些 Web 服务根据功能不同进行分类,如:支付类、查询类、天气类等。在功能相似的情况下,服务质量(quality of service, QoS)成为衡量 Web 服务优劣的重要指标。用于衡量服务的 QoS 的指标主要包括:响应时间、可用性、可靠性、吞吐量、价格以及调用成功率等^[3]。Web 服务组合不需要完全重新开发满足用户需求的新软件,仅根据用户的功能性需求,完成符合一定逻辑规律的现有服务组合。这种方法在有效降低开发成本的同时,缩短开发周期,进而满足用户时效性要求。但是,如何在规模日趋庞大的服务组合场景下,提高组合服务的 QoS,是服务组合问题的研究热点。

学术界对 QoS 感知的服务组合问题已经做了很多深入研究^[4-10],求解方法主要以进化计算算法为主,包括改进的遗传算法及粒子群算法等。

郭星等人^[11]提出了一种基于粒子群算法的动态服务组合方法 FWPSO。通过增加种群多样性来提高搜索性能,并在此基础上进行了防止早熟收敛处理,实验证明该方法相比于传统粒子群算法在搜索效率上有显著提升;张以文等人^[4]提出了一种融合 Skyline 技术的遗传粒子群算法。通过 Skyline 技术降低问题的求解空间。应用种群相似性和遗传操作避免早熟收敛,实验表明,该算法的收敛速度快于传统粒子群算法;鲁城华等^[5]提出了一种基于遗传算法的服务组合方法 MADMAGA,通过局部搜索策略提高算法收敛速度,实验表明,该算法在应对小规模服务组合场景时搜索效率高,模型收敛速度快;Sun 等人^[6]提出了基于时间序列预测和遗传算法结合的模型 ARIMA+GA 解决服务组合问题,通过对服务 QoS 记录做预测来提升服务 QoS 的准确性,并通过遗传算法来求解模型。实验结果表明,组合服务的 QoS 优于传统遗传算法;Rodrigo 等人^[9]考虑了服务 QoS 动态性,提出一种基于自回归积分移动平均(ARIMA)的方法预测服务的 QoS,然而 ARIMA 方法要求数据具有平稳性^[10]。当 QoS 趋于高度动态化后,该方法的预测能力可靠性有待验证。Wang 等人^[12]提出将强化学习应用到服务组合场景中,将 skyline 技术与 Q-learning 算法结合,建立了一种强化学习服务组合模型以解决大规模服务组合问题,但是该方法忽略了服务 QoS 的动态性。

基于进化计算算法的服务组合算法在应对小规模服务组合问题时,算法效率相对较高,但是面对大规模服务组合问题时,仍然存在模型过早收敛、算法寻优能力差的问题,导致输出的组合服务 QoS 性能低下。在历史 QoS 数据利用方面,由于服务调用频率不固定,历史 QoS 数据在时序上不平稳,现有基于预测的方法

对于历史数据的利用准确性不高。该文针对大规模服务组合问题提出一种基于深度强化学习的模型,同时,考虑到服务历史 QoS 数据的利用问题,提出一种基于卷积网络的方法,将服务历史 QoS 数据充分利用,以保证 QoS 数据的准确性。

1 Web 服务组合模型

1.1 服务组合的形式化表示

服务组合的目标是对用户需求进行拆分,而后在服务库中寻找合适的服务组件,并按照一定的逻辑顺序组合成相应的组合服务。用户的输入包括两方面:功能性需求、非功能性需求。如:“功能需求:旅行计划制定”,“非功能性需求:费用低于 5 000 元”。拆分后的功能性需求被描述为一个工作流,工作流有多种结构。如:顺序、并行、循环、选择等。图 1 所示的是包含顺序及并行两种结构的工作流,以及根据该工作流进行服务组合的过程。

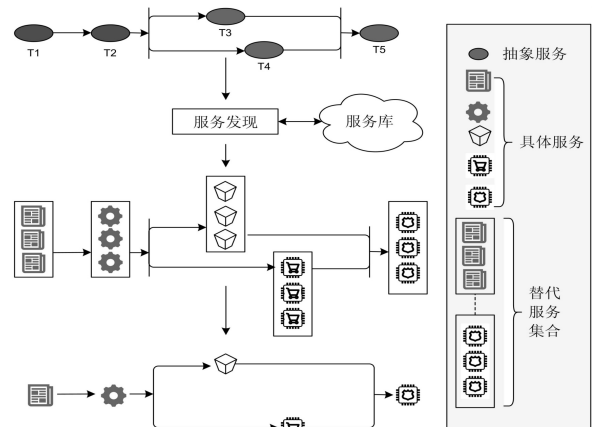


图 1 服务组合场景

定义 1: Web 服务。

Web 服务定义为一个四元组:

$$ws = \langle ID, Inp, Outp, QoS \rangle$$

其中, ID 为 Web 服务的在互联网上的唯一标识。Inp, Outp 分别表示服务的输入规范和输出结果。QoS = $\langle rt, rl, t, p, av \rangle$ 为一个五元组,其中 rt (response time) 为响应时间,表示用户发出调用请求至收到服务响应所消耗的时间,单位为“s”; rl (reliability) 表示服务可靠性,文中将可靠性定义为: {“1”, “2”, “3”, “4”}, 分别对应 {“好”, “较好”, “一般”, “差”} 四个级别。 t (throughout) 表示服务吞吐量,单位为“MB”。 p (price) 表示服务调用的价格,单位为“元”。 av (availability) = {“0”, “1”}, 表示服务可用性,“1”表示服务可用,“0”表示不可用。

定义 2: 组合 Web 服务。

组合 Web 服务定义为一个四元组:

$$WSCS = \langle N, CIn, COut, CQoS \rangle$$

其中, N 为该组合中包含的独立服务总数。 CIn 表示组合服务执行时的输入规范。 $COut$ 表示组合服务执行结束后的输出结果。 $CQoS = \langle CRT, CRL, CT, CP, CA \rangle$ 为一个五元组, 表示组合服务的服务质量。其中 $CP = \sum_1^N p_i$ 表示每个服务的价格总和; $CP = \sum_1^N rt_i$ 表示每个服务的响应时间总和。文中根据 workflow 结构的不同, 为组合服务的服务质量设计了不同的计算方法, 如

表 1 所示。

Web 服务分布式部署在各个服务提供商的服务器上, 用户通过网络访问部署在各个服务器上的服务。这样的“部署-使用”方式带来的问题是部分 QoS 指标会受到服务器负载、通信链路状态的影响^[12]。例如: 当服务器负载过大或者通信链路拥塞时, Web 服务的响应时间便会增加。因此, 该文将服务组合问题建模为一个不确定环境下的多阶段决策优化问题。

表 1 不同 workflow 顺序组合服务 QoS 属性计算方法

workflow 结构	响应时间: CRT	价格: CP	可靠性: CRL	可用性: CA	吞吐量: CT
顺序结构	$CRT = \sum_1^N rt_i$	$CP = \sum_1^N p_i$	$CRL = \prod_1^N rl_i$	$CA = \prod_1^N av_i$	$CT = \text{Min}(t_1, \dots, t_N)$
并行结构	$CRT = \text{Max}(rt_1, \dots, rt_N)$	$CP = \sum_1^N p_i$	$CRL = \text{Min}(rl_1, \dots, rl_N)$	$CA = \prod_1^N av_i$	$CT = \sum_1^N t_i$
选择结构	$CRT = \sum_i^N K * rt_i$	$CP = \sum_1^N k * p_i$	$CRL = \text{Min}(rl_1, \dots, rl_N)$	$CA = \prod_1^N av_i$	$CT = \text{Min}(t_1, \dots, t_N)$
循环结构	$CRT = N * rt_i$	$CP = N * p_i$	$CRL = rl_i^N$	$CA = \prod_1^N av_i$	$CT = N * t_i$

注: k 为选择分支中服务 ws 被选中的概率。

1.2 马尔可夫决策过程

马尔可夫决策过程 (Markov decision process, MDP) 是一种通过交互式学习来实现目标的理论框架, 也是强化学习 (reinforcement learning, RL) 问题在数学上的理想化形式^[13]。在一个马尔可夫决策过程中, 进行学习及实施决策的对象称为智能体 (agent); 智能体之外, 所有与其相互作用的事物都被称为环境 (environment)。智能体与环境持续进行交互, 根据从环境中获得的反馈来不断调整自身行动策略, 以此最大化长期收益。一个马尔可夫决策过程可以用图 2 表示。

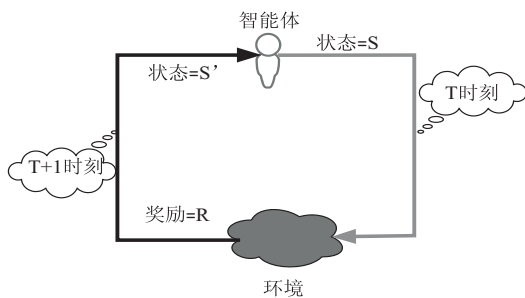


图 2 马尔可夫决策过程抽象

定义 3: 马尔可夫决策过程。

马尔可夫决策过程可以形式化定义为一个五元组:

$$MDP = \langle S, A, P, R, G \rangle$$

(1) $S = \{s_1, s_2, \dots, s_n\}$ 表示状态集, 表示环境中智能体所能观察到的所有状态, n 为状态的总数。

(2) $A = \bigcup_1^m a_i$ 为智能体的动作集, 表示智能体所能采取的所有动作, m 为动作集中的动作总数。

(3) P 为状态转移函数, $P(s' | s, a)$ 表示智能体在状态 s 下执行动作 $a (a \in A)$ 后, 状态转移到 s' 的概率值; s 表示智能体当前状态, s' 表示智能体在执行动作后, 跳转到的下一个状态。

(4) R 为奖励函数, 当动作 $a (a \in A)$ 被执行后, 环境从状态 s 转移到后继状态 s' , 智能体所获得的奖励值 $R = r(s' | s, a)$ 。

(5) G 表示回报, 是智能体所能获得的奖励的总和。

定义 4: 策略 (policy) 与值函数 (value function)。

策略 π 是状态 S 到动作 A 的映射 ($S \rightarrow A$)。 $\pi(s) = a$ 表示环境状态为 s 条件下, 智能体依据策略 π 进行决策, 执行动作 a 。

值函数表示在一个确定的策略 π 及给定奖励函数 R 的情况下, 当前状态的奖励以及未来根据策略的决策所能得到的总奖励的期望。数学表达见公式 (1):

$$V_\pi(s | R) = R(s) + \gamma V^\pi(s') \tag{1}$$

其中, 参数 $\gamma \in [0, 1]$ 被称为折扣率, 用来权衡未来奖励对累积奖励的影响。

马尔可夫决策过程中, 智能体的目标就是在所有的策略中找出最优策略以在每个状态获得最大的值函数值。形式化表达如公式 (2):

$$\pi^* = \underset{\pi}{\operatorname{argmax}} V^\pi(s), \forall s \in S \tag{2}$$

1.3 基于马尔可夫决策的服务组合模型 MDPWSC

基于马尔可夫决策模型, 服务组合可以抽象为一个五元组 $MDPWSC = \langle T, A, P, R, G \rangle$ 。各参数定义如下:

$T = \{t_1, t_2, \dots, t_n\}$ 为工作流,是智能体的状态集合。工作流是对用户功能性需求拆分成 n 个子功能后的抽象描述,其中 t_i 表示第 i 个子功能, n 表示用户功能性需求所拆分成的子功能的总数。例如:用户功能性需求为“旅行计划制定”。可将其拆分为: $t_1 =$ “订车票”、 $t_2 =$ “订酒店”、 $t_3 =$ “景点购票”、 $t_4 =$ “订车票”四个流程。

$A = \bigcup_1^N A(t_i)$ 为所有可用服务的集合,是智能体所有动作的集合。其中 $A(t_i) = \{ws_{i1}, ws_{i2}, \dots, ws_{ik}\}$ 表示满足流程 t_i 功能性需求的所有可用服务集合。 k 为满足流程 t_i 功能性需求的所有可用服务总数,

$P(t' | ws, t)$ 表示状态转移概率,表示智能体在状态 t 下,选择动作 ws ,状态转移到 t' 的概率。 t' 表示智能体在执行完任意动作后跳转到的后续状态。

$R = r(t' | ws, t)$ 表示奖励函数,表示智能体在状态 t 下,选择动作 ws ,状态转移到 t' ,环境给予智能体的奖励。

G 表示组合服务的 QoS,也是服务组合问题的优化目标。

2 算法描述

在动态网络环境中,服务的 QoS 指标将随着网络的动态性改变^[14]。如何通过现有的调用记录发掘服务最真实的 QoS 值是决定能否完成最符合用户 QoS 需求的服务组合的关键。同时,随着应用场景越来越复杂,如何应对大规模服务组合也是急需解决的问题。

2.1 基于 CNN 的奖励函数设计

奖励函数是强化学习中非常重要的变量之一,奖励函数的设定对最终优化的目标起重要作用。在基于强化学习 QoS 感知的服务组合问题中,为了最终得到最优的组合服务 QoS,常把奖励函数设定为先对子服务的 QoS 值归一化,而后对归一化后的 QoS 值加权求和^[10]。由于不同的 QoS 属性有不同的取值范围,现有方法都是首先利用公式(3)对各个属性值做归一化处理,而后采用公式(4)对多个 QoS 属性加权求和得到结果作为奖励值。

$$r_i = \begin{cases} \frac{Qos_i - \min}{\max - \min} \dots\dots (a) \\ \frac{\max - Qos_i}{\max - \min} \dots\dots (b) \end{cases} \quad (3)$$

(a):当 QoS 和 R 成正相关;(b)当 QoS 和 R 成负相关。

$$R = \sum_i^k \omega_i * r_i \quad (4)$$

其中, r_i 为 QoS 指标对应的奖励值, ω_i 为指标对应的权值, k 为 QoS 指标总数。

这种方法存在的问题是需要人为设定各个指标之间的权重关系 ω_i ,准确性不高,不能反映各 QoS 之间的真实联系。同时,Web 服务的每次调用都会产生相应的 QoS 记录,传统的方法只选取最近一条调用记录作为计算奖励值的依据,对于服务历史 QoS 数据的利用过于粗糙。以上两点原因将导致计算出的奖励值可靠性不足,并最终导致组合服务的 QoS 可靠性差。

深度学习 (deep learning, DL) 和神经网络 (neural network) 是当下学术研究的热点,并且在多个领域取得了显著的成果。深度学习算法通过多个神经元组成神经层,再由多个神经层组成复杂的神经网络,进而解决复杂的问题。从数据集^[15]中可以获得单个服务关于多个 QoS 指标的历史数据。该文利用卷积神经网络 (convolutional neural network, CNN) 在特征提取上的突出优势^[16],将单个服务的多次调用记录以及相应 QoS 指标组合成二维矩阵输入网络,通过卷积网络提取服务多条调用记录之间的相互联系特征,以及各个 QoS 指标之间的相互关系特征,进而输出服务对应的真实 QoS 值。

网络结构如图 3 所示。

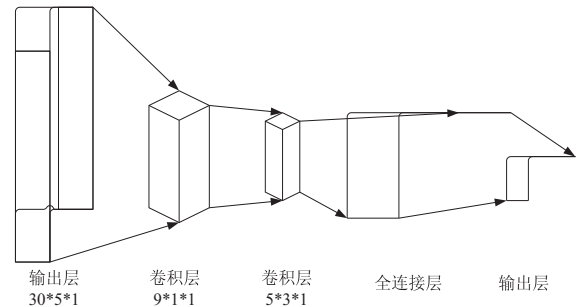


图 3 CNN 网络结构

输入层为在状态 t' 下被选中的服务的多次调用记录与 QoS 指标组成的二维矩阵,实验中选择 30 条服务调用记录中五个 QoS 指标的值组成 $30 * 5$ 的输入矩阵,经过一层卷积核大小为 $9 * 1 * 1$ 的卷积层提取多条记录之间的特征和一层卷积核大小为 $5 * 3 * 1$ 的卷积层进一步提取多条记录之间的特征以及提取各个 QoS 指标之间的特征,而后经过全连接层,最终输出 t' 状态下,被选中服务对应的奖励值 R 。相比起传统的利用某一条 QoS 记录计算奖励值的方法,基于 CNN 的方法可以将多条 QoS 属性信息利用起来,更加能反映服务的真实 QoS 情况。

2.2 基于深度强化学习的服务组合算法 ADR-WSC

Q-Learning 算法^[12]是求解强化学习问题的基本方法之一,其基本思想是通过一张 Q 表来存储“动作-值”函数 $Q(t, ws)$ 的值。对应的 Q 值是对真实奖励的评估。算法通过比较 Q 值的估计值和实际值,不断更新 Q 表,直至估计值接近真实值。在服务组合问题

中,其迭代公式如下:

$$Q(t, ws) = Q(t, ws) + \alpha(r(t | t, ws) + \gamma \max_{ws'} Q(t', ws') - Q(t, ws)) \quad (5)$$

其中, α 为学习率, ws' 为 t' 状态下选择的动作。

当面临解决大规模离散空间或连续状态空间的问题时,例如:待解决问题的状态空间大小为 5, 每个状态对应的动作空间大小为 10 时,这时的 Q 表是一个 $5 * 10$ 的矩阵,计算和更新开销相对较小,训练和后续查表开销不大。但应用到实际生产领域当中,一个子功能对应的可选服务往往成百上千个,工作流的长度也数以十甚至百计。对于一个工作流长度为 30,即状态空间大小为 30,每个状态空间对应的动作空间大小为 500 时,将会形成 500^{30} 种组合方案,这是个巨大的搜索空间。并且,一次训练过程就需要经过 500^{30} 次计算。在网络波动的情况下,光是计算开销就足以导致计算结果失效。因此,在 Q-Learning 的基础上,加入了深度网络(deep neural network, DNN)替代 Q 表,以解决 Q-Learning 在面对大规模问题时的瓶颈^[17]。

基于强化学习良好的决策能力和卷积网络的特征提取能力以及神经网络本身强大的表征能力,该文提出了一种基于深度强化学习的 QoS 自适应服务感知算法:“Adaptive Deep Reinforcement Learning - Web Service Composition(ADR-WSC)”。为了解决奖励值 R 的可靠性问题,采用了卷积神经网络代替传统的直接对 QoS 记录加权求和的方式;为了解决大规模“状态-动作”值存储于 Q 表中导致算法性能低下的问题,采用了深度神经网络替代传统的 Q 表。综合上述两点改进,传统的 Q-Learning 中 Q 值的迭代更新公式应转化为:

$$Q(t, ws) = Q(t, ws) + \alpha(\text{CNN}(t | t, ws) + \gamma \max_{ws'} \text{DNN}(t', ws') - Q(t, ws)) \quad (6)$$

算法具体过程如下:

算法:服务组合算法 ADR-WSC。

(1) 初始化“动作-值”函数 $Q(T, ws, \omega)$ 的参数 ω 、加载 CNN(T, ws) 网络已保存的训练参数。加载随机探索率 ϵ 、学习率 α 、工作流长度 T 、初始化经验回放区域 D 、确定训练轮数 MAX_EPISODE,以及网络训练的 BATCH_SIZE

(2) For each episode in MAX_EPISODE:

(3) For t_i in T :

(4) 在工作流 t_i 对应的可用服务集 $A(t_i)$ 中根据 ϵ -greedy 策略选取相应的服务 ws

(5) 绑定服务,获得下一个状态 t' ,并根据 CNN(t_i, ws) 网络计算出对应的奖励 $r(t' | ws, t)$

(6) $t_i = t'$,根据 t_i, ws 计算 Q 值,公式如下:

$$y(t_i, ws) = \begin{cases} r_{t_i} & \dots \dots \dots t_i \text{ 是终止流程} \\ r_{t_i} + \gamma \max_{ws'} Q(t', ws, \omega) & \dots t_i \text{ 不是终止流程} \end{cases}$$

(7) 将 $[t_i, ws, y(t_i, ws)]$ 存入经验回放区

(8) 判断缓冲区长度是否达到 BATCH_SIZE,若达到,则将经验区中的数据喂入网络训练,利用梯度下降算法更新 $Q(T, ws, \omega)$ 中的所有参数。后清空缓冲区。否则,结束此次循环

(9) End for

(10) End for

3 实验与分析

3.1 实验设置

实验数据集的基础为公共有效数据集 QWS^[15], QWS 数据集中的所有数据均为真实网络中可用 Web 服务被调用后获得的统计数据。包含 2 500 个 Web 服务及其对应的 11 个 QoS 属性值,实验中选用了数据集中的四个 QoS 属性:响应时间、可用性、可靠性、吞吐量,并扩充了服务的价格属性。网络使用深度学习框架 Tensorflow 搭建,实验在一台操作系统为 Win10, CPU 为 E3-1226,内存为 16G, GPU 为 NVIDIA GT-1030 的机器上运行。

3.2 结果分析

(1) 初始探索率对总收益的影响。

用于预测奖励 R 的 CNN 网络是在 ADR-WSC 算法运行之前就已经完成训练的,因此算法迭代过程中,不需要再对其做额外的训练。而用于拟合‘动作-价值’函数的 DNN 网络则是在线训练的,经验回放区域的容量大小、智能体的随机探索率等超参数会很大程度上影响实验结果,因此,实验中先验证了初始探索率 ϵ 的大小对总收益(即组合模型的整体 QoS)的影响。

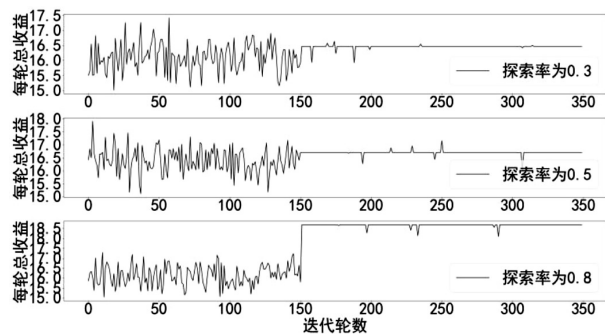


图 4 探索率对总收益值的影响

图 4 显示的是服务组合模型中流程数为 10,每个流程对应有 300 个备选服务时,探索率 $\epsilon = 0.3$ 、 $\epsilon = 0.5$ 以及 $\epsilon = 0.8$ 时,算法收敛时最大收益的对比。实验中固定了学习率 $\alpha = 0.9$ 。经验区容量为 150。为了模型更快的收敛,实验中还采用了探索率递减的方法。算法迭代的过程中,探索率会随着迭代轮数逐渐降低,实验中设置的探索率值仅仅是算法开始迭代时的探索率。结果表明,当模型收敛时,在 $\epsilon = 0.3$ 和 $\epsilon = 0.5$ 情况下,智能体获得的总的收益值只有不到 17,而 $\epsilon = 0.8$ 时,收益值达到 18.5 以上。笔者认为,让智能体

前期更多地探索环境知识,积攒更多的训练数据,更加有利于模型的准确性。因此,后续实验将采取探索率 $\varepsilon = 0.8$ 。

(2) 经验区容量对算法收敛速度的影响。

经验区容量直接决定了喂入网络的训练数据量,对算法收敛起了重要作用,因此,文中接着验证了经验区容量大小对于模型收敛的影响。图 5 显示的是服务

组合模型中流程数为 10,每个流程对应有 300 个备选服务时,不同经验区容量对算法收敛过程的影响。实验中固定了探索率 $\varepsilon = 0.8$,学习率 $\alpha = 0.9$ 。结果表明,在探索率为 0.8 的情况下,经验区容量取 150、200、250、300 均可以保证算法收敛到最优解。从节省资源的角度上来说,选择 150 更合适。因此,后续实验中,选择经验区容量为 150。

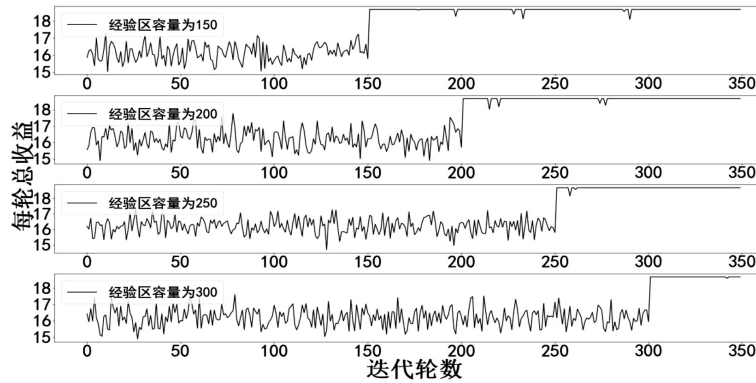


图 5 经验区容量对总收益的影响

(3) 组合服务 QoS 对比。

在确定超参数以后,还对模型的整体性能做了验证。首先是算法收敛后的 QoS 值对比。算法的核心目的是提供 QoS 更优的组合服务,因此组合服务的 QoS 是模型最重要的指标。实验中选取了基于粒子群算法的 FWPSO 算法^[11]、基于时间序列预测和遗传算法的 ARIMA+GA 算法^[6]做对比。同时,为了获取实际最优组合服务 QoS,实验中还使用了穷举法求解作为参照。

断更新网络,即使问题规模扩大,也可以找出最优解。

(4) 算法时间开销对比。

由于每个算法收敛时迭代轮数不同,实验中对比的是当模型收敛时,所需要的最小时间开销。实验中没有将穷举法的运行时间绘制在图上,因为其时间开销跟其他三种算法不在一个量级上。从图 7 中可以看出,当单个流程的备选服务数量越来越多的时候,三种算法的时间开销都是随着问题规模递增的,但是 ADR-WSC 算法的时间开销均低于其他两种算法,ADR-WSC 算法运行速度快可能的原因是:当完成网络训练以后,算法执行过程中,只需要通过网络完成特定数据的运算即可,是针对固定数据的运算,而不像其他两种算法每次运行都必须迭代所有数据。

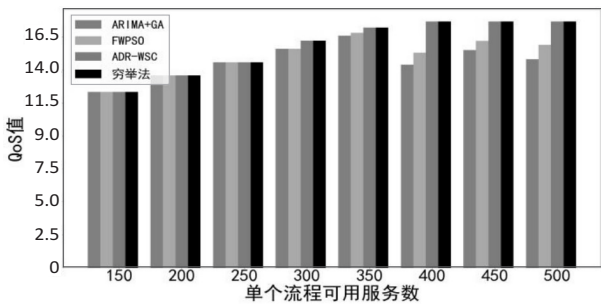


图 6 各算法在不同规模服务组合场景中 QoS 值对比

图 6 显示的是 ARIMA+GA、FWPSO、ADR-WSC 以及穷举算法在求解服务组合问题模型后,输出的组合服务 QoS 值。需要说明的是:实验中对每种算法运行了多次,取收敛后最低的 QoS 值作为对比数据,这样做的好处是:可以明确获知算法在最坏的情况下输出的结果。从图中可以看出,当单个流程的候选服务数小于等于 250 时,ARIMA+GA、FWPSO、ADR-WSC 均可以求得最优解。当备选服务大于 250 时,ARIMA+GA、FWPSO 出现提前收敛至局部最优的情况,无法找到全局最优解。而 ADR-WSC 算法迭代的过程中,不

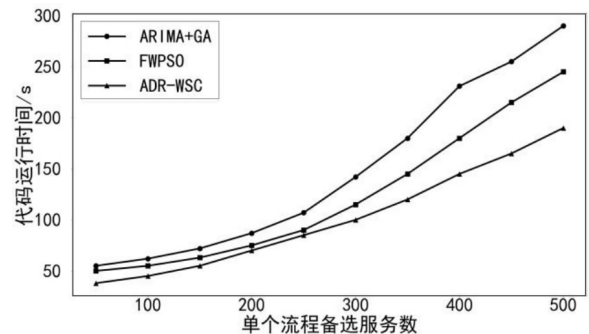


图 7 GA, PSO, ADR-WSC 运行时间对比

4 结束语

提出了一种基于深度强化学习的自适应优化方法 ADR-WSC 来解决大规模 QoS 感知的 Web 服务组合优化问题,考虑了多个 QoS 属性之间的相互关系以及如何利用 Web 服务多条历史调用记录分析 Web 服务

QoS 可靠性,并改进了基于传统 Q-Learning 的算法,以解决大规模服务组合问题。实验结果表明,该方法在组合服务 QoS 上优于传统寻优方法,并且在遇到大规模服务组合场景时,具有更高的可扩展性,能够有效地处理大规模服务组合问题。

云计算发展速度越来越快,Web 服务的数量在迅猛增长,对于服务组合问题来说,每个子流程的候选服务数量越多,完成组合的难度越大,算法运行开销也越久。如果用户对于时效要求进一步提高,比如:当应用场景为高速智能驾驶时,对于算法的时效要求更为严格。今后将进一步研究如何提高超大规模服务组合低时耗的问题。

参考文献:

- [1] 王纪军,张 斌,顾永生,等. 云环境中 Web 应用的微服务架构评估[J]. 计算机系统应用,2017,26(5):9-15.
- [2] 邓水光,吴朝晖. Web 服务组合方法综述[J]. 中国科技论文在线,2008,3(2):79-84.
- [3] ZHENG Z,LYU M R. WS-DREAM:a distributed reliability assessment mechanism for web services[C]//IEEE international conference on dependable systems & networks with FTCS & DCC. Anchorage:IEEE,2009.
- [4] 张以文,吴金涛,郭 星,等. 一种基于动态 Skyline 和遗传粒子群优化的云服务组合方法[J]. 小型微型计算机系统,2016,37(11):2552-2557.
- [5] 鲁城华,寇纪淞. 求解 Web 服务组合 QoS 优化的多属性决策及自适应遗传算法[J]. 计算机科学,2019,46(2):187-195.
- [6] SUN X,WANG S,XIA Y, et al. Predictive-trend-aware composition of web services with time-varying quality-of-service[J]. IEEE Access,2020,8:1910-1921.
- [7] 杨文青,刘广钟. 智慧城市中 QOE 感知服务组合策略的研究[J]. 计算机技术与发展,2021,31(4):176-181.
- [8] 李 强,周井泉,张严凯. 基于参数自适应差分进化算法的 Web 服务组合[J]. 计算机技术与发展,2018,28(3):101-104.
- [9] CA LHEIROS R N,MASOUMI E,RANJAN R, et al. Workload prediction using ARIMA model and its impact on cloud applications' QoS[J]. IEEE Transactions on Cloud Computing,2015,3(4):449-458.
- [10] SONG H,DU D,LI Y. Research on precise demand forecast of retail commodities based on LSTM-ARIMA joint model[J]. Journal of Physics: Conference Series,2021,1865(4):042049.
- [11] 郭 星,陈姗姗,张以文,等. 烟花粒子群优化算法在 Web 服务组合上的应用[J]. 小型微型计算机系统,2018,39(6):1312-1316.
- [12] WANG H,HU X,YU Q, et al. Integrating reinforcement learning and skyline computing for adaptive service composition[J]. Information Sciences,2020,519(2):141-160.
- [13] SUTTON R S,BARTO A G. Reinforcement learning[J]. A Bradford Book,1998,15(7):665-685.
- [14] ABDULLAH M N,BHAYA W S. Predication of quality of service (QoS) in cloud services:a survey[J]. Journal of Physics: Conference Series,2021,1804(1):012049.
- [15] AL-MASRI E,MAHMOUD Q H. Discovering the best web service:a neural network-based solution[C]//IEEE international conference on systems. San Antonio:IEEE,2009.
- [16] GORBAN A N,MIRKES E M,TUKIN I Y. How deep should be the depth of convolutional neural networks;a backyard dog case study[J]. Cognitive Computation,2020,12(1):388-397.
- [17] XIE A,HARRISON J,FINN C. Deep reinforcement learning amidst lifelong non-stationarity[J]. arXiv:2006.10701v1,2020.