

约束强度感知的时序约束服务组合

叶恒舟*, 胡志丹

(桂林理工大学广西嵌入式技术与智能系统重点实验室 广西 桂林 541006)

【摘要】保障全局时序的时序约束分解模型(TCD)可以将时序服务组合分解成约束分解与局部优选两个相对独立的过程,但该模型可能丢失可行组合方案,在用户约束强度较强时可能导致无解。该文提出了一种约束强度感知的时序约束分解模型(CIA-TCD),通过在现有的TCD模型中引入松弛因子,使得用户约束强度较弱时,能保证全局约束,而约束强度较强时,也能够保留一定量的组合方案,从而提高找到可行方案的概率。分析表明,当约束强度较强时,CIA-TCD模型较TCD模型找到可行组合方案的概率明显更大。

关键词 约束分解; 约束强度感知; 模糊推理; 服务组合; 时序约束

中图分类号 TP393 **文献标志码** A **doi**:10.3969/j.issn.1001-0548.2019.06.012

Constraints Strength-Aware Temporal Constraints Service Composition

YE Heng-zhou* and HU Zhi-dan

(Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology Guilin Guangxi 541006)

Abstract The temporal constraints decomposition model (TCD), which guarantees the global temporal, can decompose the temporal service composition into two relatively independent processes: constraints decomposition and local optimization. However, the model may lose the feasible combination scheme, which may lead to no solution when the user constraints strength is strong. This paper proposes a constraints strength-aware temporal constraints decomposition model (CIA-TCD), which introduces a relaxation factor into the existing TCD model. When the user constraint strength is weak, the global constraint can be guaranteed; and when the constraint strength is strong, a certain amount of combination scheme can be reserved, thereby increasing the probability that a feasible solution can be found. The experimental analysis shows that the CIA-TCD model has a significantly better probability of finding a feasible combination scheme than the TCD model when the constraint strength is strong.

Key words constraint decomposition; constraint strength-aware; fuzzy inference; service composition; temporal constraint

面向服务的体系结构(service of architecture, SOA)^[1]作为一种新型软件构架技术,因其允许集成服务和易于开发复杂的商业应用,已被广泛应用于环境智能^[2]、物联网^[3]、无人机控制^[4]、远程医疗^[5]等诸多领域。Web服务及其组合技术,是实现SOA的关键技术。

时序约束是Web服务组合中备受瞩目的非功能特性。多数QoS(quality of service)感知的Web服务组合方法^[6-10]仅考虑了全局用户时序约束。文献[11]考虑了服务之间及服务内部的时序约束,主要关注了组合服务的时序兼容性问题。文献[12]除了考虑全局用户时序约束之外,还考虑了任务之间的时序约束(即具有直接或间接继承关系的两个任务之间的时

序约束),但其建立的数学模型比较复杂,缺乏求解该模型的高效算法。文献[13]不仅考虑了全局用户时序约束而且还考虑了施加于由多个任务组成的某种模式的时序约束,并提出了一种TCD模型来解决这类时序约束问题。TCD模型可以保障全局时序约束,但可能丢失可行组合方案;当用户提出的时序约束条件较强时,可能导致找不到组合方案。

现有Web服务应用场景多为支持多用户、高并发的云平台,其资源分配与任务调度策略对服务的时序约束高度敏感。为此,本文提出一种CIA-TCD模型,其思想是在文献[13]提出的TCD模型中引入松弛因子,使得当用户约束强度较弱时,仍可保障全局时序约束;当用户约束强度较强时,保留一定数

收稿日期: 2018-12-24; 修回日期: 2019-03-19

基金项目: 国家自然科学基金(51365010); 广西自然科学基金(2014GXNSFBA118269)

作者简介: 叶恒舟(1980-),男,博士,教授,主要从事服务组合、最优化方面的研究. E-mail: 18123408@qq.com

量的候选方案(有些可能不满足全局时序约束), 以提升找到可行组合方案的概率。

1 时序服务组合描述

1.1 服务组合

与文献[13-19]类似, 本文仅考虑工作流已知服务组合问题, 且工作流中每个活动的候选服务已知。工作流由常见的4种组合模式即顺序(SEQ)、选择(XOR)、并发(AND)及循环(LOOP)进行有限次迭代而成。为方便表述, 引入虚拟活动的概念来描述工作流。

定义 1 虚拟活动(VA): 一个VA可由一个三元组(CP, AL, K)描述。其中CP表示工作流模式, 分别为SEQ、XOR、AND或LOOP; AL= $\langle a_1, a_2, \dots, a_n \rangle$ 表示一个有序列表, 其中 $a_i (i = 1, 2, \dots, n)$ 表示一个活动或者VA; K仅当CP为LOOP时有效(无效时K=1), 表示AL中元素的循环次数。

对于文献[13]给出的工作流(图1), 可以由内到外构成如下VA序列, 用以描述该工作流, 其中VA₅代表整个工作流。

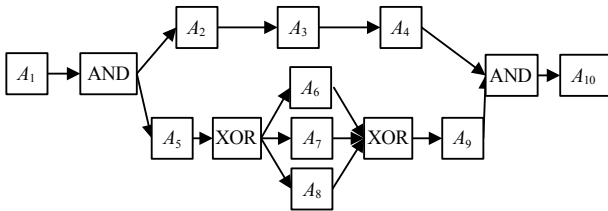


图1 一个工作流例子

VA₁(SEQ, $\langle A_2, A_3, A_4 \rangle$, -1)

VA₂(XOR, $\langle A_6, A_7, A_8 \rangle$, -1)

VA₃(SEQ, $\langle A_5, VA_2, A_9 \rangle$, -1)

VA₄(AND, $\langle VA_1, VA_3 \rangle$, -1)

VA₅(SEQ, $\langle A_1, VA_4, A_{10} \rangle$, -1)

不难验证, 对于一个包含 n 个活动的工作流, 可由不超过 n 个VA组成的序列描述。

1.2 时序约束

时序约束可分为3类: 施加于整个工作流全局的时序约束、施加于由多个活动构成的某种组合模式的时序约束和施加于单个活动的本地时序约束。对于前两类, 可以用针对VA的时序约束描述; 对于第三类, 可以用针对活动的时序约束描述。以图1所示工作流为例, 可用 $tc(VA_5) \leq 12$ 表示整个工作流的完成时间不超过12个单位, $tc(VA_1) \leq 4$ 表示完成由活动 A_2, A_3, A_4 构成的顺序模式的时间不超过4个单位, $tc(A_3)$ 表示应在1个时间单元内完成活动 A_3 。

2 CIA-TCD模型

一个包含 n 个活动(记为: A_1, A_2, \dots, A_n)的工作流, 活动 A_i 拥有 m_i 个候选服务(第 j 个候选服务记为 s_{ij} , 对应的完成时间记为 $s_{ij}.pt, 1 \leq j \leq m_i$), 用于描述该工作流的VA序列有 u 个(记为 $VA_1, VA_2, \dots, VA_u, u \leq n$)。用户对于该工作流的时序约束, 可以施加到某些VA或者某些活动之上。时序服务组合问题可定义为: 当满足用户时序约束时, 使得其某些QoS最优化(例如: 执行价格最低)。

2.1 用户时序约束分解

用户时序约束分解的目标是寻找每个活动应该满足的时序上限, 记为 $ULT(A_i), i = 1, 2, \dots, n$ 。当每个活动应该满足的时序上限确定时, 任何一个VA应该满足的时序上限可由算法UtterLimit(VA)计算所得。为便于描述, 活动 A_i 被看作一个VA(NONE, $\langle A_i \rangle$, -1)。该算法在最坏情况下, 需要递归遍历每个VA和活动, 其时间复杂度为 $O(n)$ 。

算法UtterLimit(VA)

输入: 虚拟任务VA

输出: VA应满足的时序上限

```

If (VA.CP == NONE) {
    获取VA.AL中的元素 $A_i$ ;
    Return ULT( $A_i$ ); }
Else if (VA.CP == SEQ) {
    Set value = 0; //初始化VA的时序上限
    For each element in VA.AL
        value += UtterLimit (element);
    Return value; }
Else if (VA.CP == XOR) {
    Set value = +∞;
    For each element in VA.AL {
        temp = UtterLimit (element);
        If (temp < value)
            value = temp; }
    Return value; }
Else if (VA.CP == AND) {
    Set value = 0;
    For each element in VA.AL {
        temp = UtterLimit (element);
        If (temp > value)
            value = temp; }
    Return value; }
Else if (VA.CP == LOOP) {
    Set value = 0;
    
```

```

For each element in VA.AL
    value += UtterLimit (element);
Return value * VA.K; }
    
```

2.2 时序约束强度

全局QoS约束分解策略是研究QoS感知的Web服务组合的重要方法。现有QoS约束分解模型^[13,20-26]可分为3类：依据经验公式或规则建立的分解模型^[20-21]，其优点是易于实现，但它不能保障全局QoS约束，也不能保证不丢失可行组合方案；以保障全局QoS约束建立的分解模型^[13,22-25]，它在用户约束强度较强时，容易丢失可行组合方案；保障以不丢失可行组合方案建立的分解模型^[26]，其目的是过滤不可行的候选服务，在用户约束强度较弱时过滤效果较差，也不能保障全局约束。由此可见，用户约束强度会影响QoS约束分解的效果。

用RA(VA)描述VA涉及到的活动的集合：

$$RA(VA) = \begin{cases} \{a | a \text{ in } VA.AL\} & VA.CP = NONE \\ \bigcup_{VA \in VA.AL} RA(VA) & VA.CP \neq NONE \end{cases} \quad (1)$$

若为RA(VA)中的每个活动选择一个候选服务，其执行时间为对应的活动的所有候选服务的平均执行时间，由这些服务组合成的执行方案的平均执行时间称为VA的平均执行时间，记为APT(VA)。设用户对VA的时序约束为 $tc(VA) \leq \tau(VA)$ ，用CI(VA)表示用户对VA的时序约束强度：

$$CI(VA) = \frac{\tau(VA)}{APT(VA)} - 1 \quad (2)$$

2.3 CIA-TCD模型

构建CIA-TCD模型的基本思想是在TCD模型的基础上引入松弛因子。当约束强度较弱时，取松弛因子为0以保障全局约束；当约束强度较强时，松弛因子取一个大于0的合适值，以保留一定数量的组合方案和降低丢失可行组合方案的概率。

CIA-TCD模型可描述如下：

目标：

$$\max \prod_{i=1}^n \# \{s_{ij} | s_{ij}.pt \leq ULT(A_i) \quad 1 \leq j \leq m_i\} \quad (3)$$

约束条件：

$$ULT(A_u) \leq \tau(A_u) \quad 1 \leq u \leq n \quad (4)$$

$$ULT(VA_v) \leq (1 + \lambda_v)\tau(VA_v) \quad 1 \leq v \leq n \quad (5)$$

$$\min(A_u.pt) \leq ULT(A_u) \leq \max(A_u.pt) \quad 1 \leq u \leq n \quad (6)$$

式中，#{}表示集合中元素的个数； $\tau(A_u)$ 、 $\tau(VA_v)$ ($1 \leq u \leq n$, $1 \leq v \leq n$)分别为用户对各个活动、VA的时序约束(若无约束，设为 $+\infty$)； $\min(A_u.pt)$ 、 $\max(A_u.pt)$

($1 \leq u \leq n$)分别表示 A_u 的候选服务的执行时间的最小值与最大值； λ_v ($1 \leq v \leq n$)表示 VA_v 的时序约束所需引入的松弛因子； $ULT(A_u)$ ($1 \leq u \leq n$)为模型变量，表示 A_u 的时序约束上限。式(3)表明CIA-TCD模型的优化目标为最大化剩余组合方案的个数。

2.4 松弛因子的自适应调节方法

若能确定CIA-TCD模型中 λ_v ($1 \leq v \leq k$)的值，该模型可以采用文献[13]中给出的贪心算法求解。 λ_v 的取值主要与 $RA(VA_v)$ 中活动个数(记为 $|RA(VA_v)|$)和 $CI(VA_v)$ 相关，可采用模糊推理规则来实现 λ_v 的自适应调节。图2a、图2b、图2c分别描述了 $|RA(VA_v)|$ 、 $CI(VA_v)$ 和 λ_v 的隶属函数。采用表1所示的推理规则，可得 λ_v 的投影曲面如图3所示。

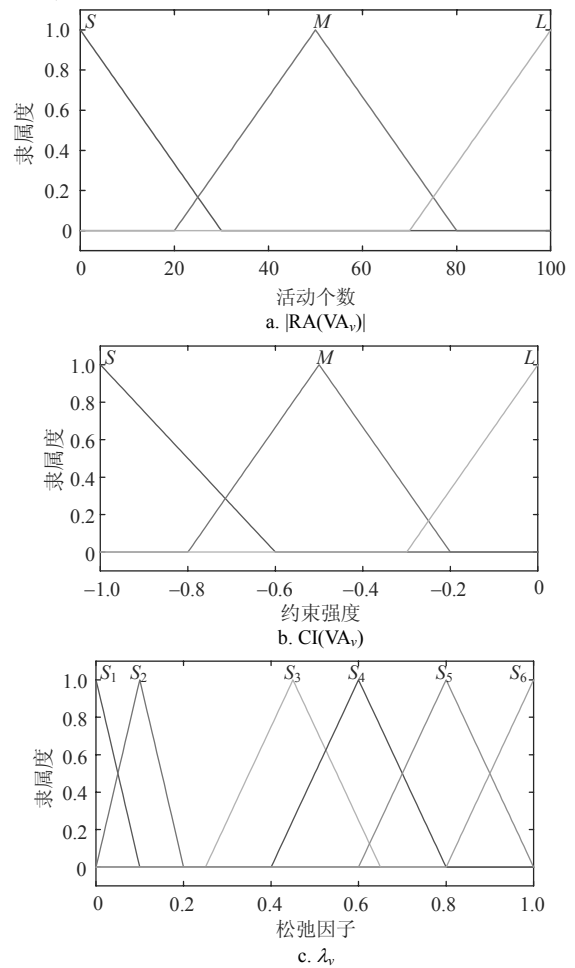


图2 隶属度函数

表1 模糊推理规则

| RA(VA _v) | CI(VA _v) | | |
|----------------------|----------------------|----------------|----------------|
| | S | M | L |
| S | S ₅ | S ₃ | S ₁ |
| M | S ₅ | S ₃ | S ₂ |
| L | S ₆ | S ₄ | S ₂ |

其中 $|RA(VA_v)|$ 和 $CI(VA_v)$ 的模糊子集均为{S, M,

$L\}$; λ_v 则划分为6个等级, 从小到大依次记为 S_1, S_2, \dots, S_6 。

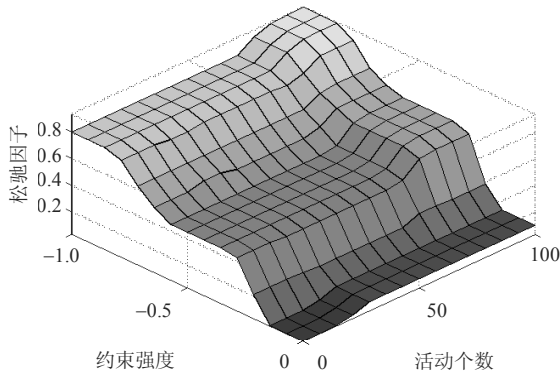


图3 模糊规则的曲面投影

3 实验分析

本文采用CIA-TCD模型和文献[13]中的TCD模型找到可行组合方案的概率和剩余组合方案个数进行对比, 以及仅在CIA-TCD模型下找到可行组合方案的概率随约束个数变化的仿真实验。实验与文献[13]类似, 考虑了3种规模的服务组合问题: G1(活动个数100, 每个活动拥有的候选服务个数100), G2(活动个数100, 每个活动拥有的候选服务个数200), G3(活动个数200, 每个活动拥有的候选服务个数100)。 workflow根据活动个数而随机生成, 其中AND、XOR及SEQ模式的比例为1:1:2。以QWS数据集^[27-28]中的响应时间模拟服务的执行时间。在求解模型所采用的贪心算法中, 参数步长StepNumber取为50, 参数测试次数TestTimes取值100, 时序约束个数为活动个数的0.3倍。实验代码用java 7编写, 运行环境是Intel(R)Core(TM) i5-2430M、2.50 GHz Intel Xeon双核处理器, 4.0 GB内存及win7操作系统。

3.1 找到可行组合方案的概率对比

与TCD模型相比, 本文提出的CIA-TCD模型由于引入了松弛因子, 使得当时序约束较强时, 仍能保留一定量的候选服务, 因此在时序约束较强时, 采用CIA-TCD模型应比TCD模型更可能找到可行组合方案。图4给出G1和G2规模时, 分别采用CIA-TCD及TCD模型时找到可行组合方案的概率随约束强度变化的情况(设定所有约束的约束强度相同, 约束强度从-0.8以步长0.1增加到-0.1)。从图4中可以发现: 约束强度较强时(-0.7~-0.3之间), 采用CIA-TCD模型比TCD模型更可能找到可行组合方案; G2与G1相比, 活动个数相同, 候选服务个数增加一倍, 在同

等约束强度时, 两种模型在G2时都比G1时更能找到可行组合方案, 这在一定程度上表明松弛因子与候选服务个数关联不大。图5对比了G1和G3规模时, 两种模型找到可行组合方案的概率随约束强度变化的情况。该图同样表明: 当约束强度较强时, CIA-TCD模型比TCD模型更可能找到可行组合方案。从图5可以发现: 对于G1和G3两种问题规模, 两种模型找到可行方案的概率变化的约束强度区间基本一致(CIA-TCD模型为(-0.7, -0.4), TCD模型为(-0.5, -0.1)), 这表明松弛因子的选取对活动个数具有较强适应能力(G1与G3的区别在于活动个数不同)。

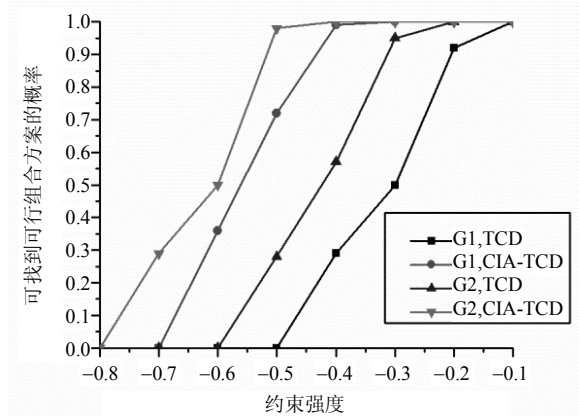


图4 找到可行组合方案的概率对比(G1和G2)

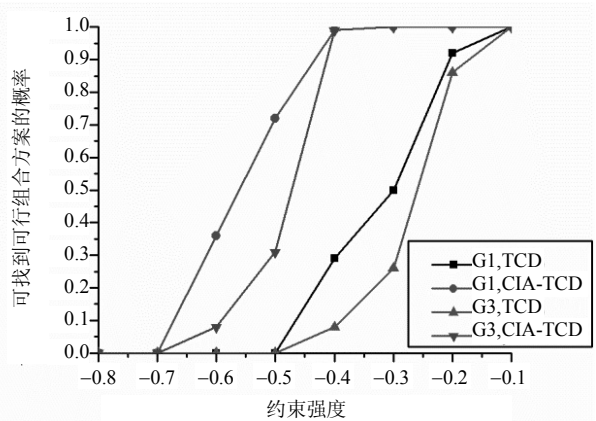


图5 找到可行组合方案的概率对比(G1和G3)

3.2 剩余组合方案个数对比

引入松弛因子的另一个目的是使得剩余组合方案(不一定是可行组合方案)个数保持在一定范围内。图6给出了在G1、G2及G3规模时, 采用CIA-TCD及TCD模型时, 剩余组合方案个数随约束强度变化的情况(由于剩余组合方案个数很大, 纵坐标为其采用科学计数法时的指数值)。在同样规模时, 与TCD模型相比, CIA-TCD模型可以在更大约束强度区间上基本保持稳定(以G1为例, 采用CIA-TCD时的稳定区

间为 $[-0.7, -0.1]$ ，而采用TCD时的区间为 $[-0.4, -0.1]$ 。对于CIA-TCD模型，对比G1和G2时的情形可以发现，它们的变化区间一致，但由于G1时候选服务个数较少，剩余组合方案个数也较少，这表明若要更好的控制剩余组合方案个数，松弛因子的选取还应考虑候选服务个数的影响；对比G1和G3时的情况，活动个数较多时，稳定区间较窄，表明在同等约束强度时，活动个数越多(约束个数也越多，其值是活动个数的0.3倍)，越不容易找到可行组合方案。

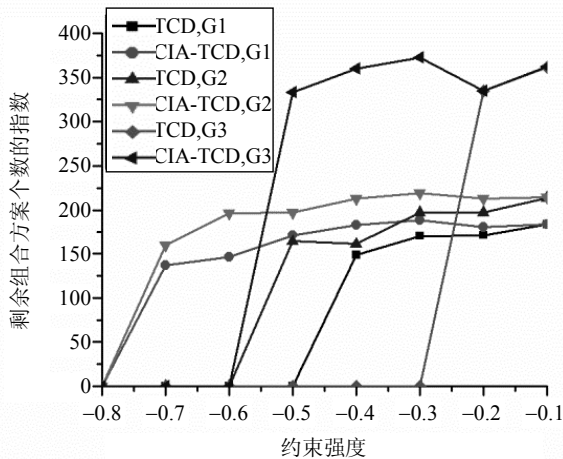


图6 剩余组合方案个数的指数随约束强度变化

3.3 约束个数的影响

图7给出了在G1规模、CIA-TCD模型、约束个数从20以步长20递增至100时，找到可行方案的概率随约束强度变化的情况。可以看出：无论哪种约束个数，当约束强度小于等于 -0.7 时找不到可行组合方案，当约束强度大于等于 -0.4 时几乎总能找到可行组合方案，在区间 $(-0.7, -0.4)$ 上，约束个数对可找到组合方案的概率有些影响，但影响不大，表明松弛因子的选取受约束个数的影响不大。

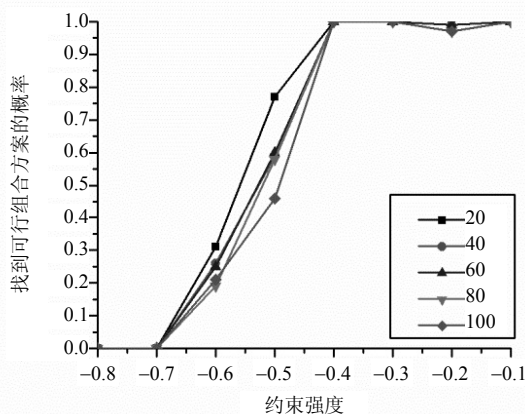


图7 找到可行方案的概率随约束强度变化

4 结束语

应用于服务组合的时序约束分解模型对用户约束强度较为敏感，尤其是当用户约束强度较强时，保障全局约束的分解模型容易丢失本就不多的可行组合方案。通过在保障全局约束的时序约束分解模型中引入松弛因子建立的CIA-TCD模型，在约束强度较大时，仍然能够保留一定量的可行组合方案，从而明显提高找到可行组合方案的概率。在采用模糊推理自适应调节松弛因子的值时只关注了约束强度及任务个数的影响。而仿真实验结果表明候选服务个数、约束个数等对松弛因子的取值也有影响，这将是以后研究的一个方向。

参考文献

- [1] WANG P W, DING Z J, JIANG C J, et al. Automatic web service composition based on uncertainty execution effects[J]. IEEE Transactions on Services Computing, 2016, 9(4): 551-565.
- [2] MANUEL S, JOSE A, ERNESTO E. Integrating SOA and MAS in intelligent environments[J]. Dyna, 2018, 85(206): 268-282.
- [3] ZHANG Y, CHEN J L, CHENG B. Integrating events into SOA for IoT services[J]. IEEE Communications Magazine, 2017, 55(9): 180-186.
- [4] RODRIGUES D, PIRES R D M, MARCONATO E A, et al. Service-oriented architectures for a flexible and safe use of unmanned aerial vehicles[J]. IEEE Intelligent Transportation Systems Magazine, 2017, 9(1): 97-109.
- [5] TRAORE B B, KAMSU-FOGUEM B, TANGARA F. Integrating MDA and SOA for improving telemedicine services[J]. Telematics & Informatics, 2016, 33(3): 733-741.
- [6] 叶恒舟, 李陶深, 关云慧. 一种自治愈合的QoS感知的Web服务组合方法[J]. 电子科技大学学报, 2017, 46(3): 618-624.
- [7] 叶恒舟, 陆湘鹏. 基于离散粒子群优化的鲁棒Web服务组合[J]. 电子科技大学学报, 2018, 47(3): 443-448.
- [8] XU J J, GUO L, ZHANG R X, et al. QoS-aware service composition using fuzzy set theory and genetic algorithm[J]. Wireless Personal Communications, 2018, 102(2): 1009-1028.
- [9] LU W, WANG W D, BAO E D, et al. FAQs: Fast web service composition algorithm based on QoS-aware sampling[J]. IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences, 2016,

- 99(4): 826-834.
- [10] 谭文安, 赵尧. 基于混沌遗传算法的Web服务组合[J]. 计算机集成制造系统, 2018, 24(7): 1822-1829.
TAN Wen-an, ZHAO Yao. Web service composition based on chaos genetic algorithm[J]. Computer Integrated Manufacturing Systems, 2018, 24(7): 1822-1829.
- [11] DU Y H, TAN W, ZHOU M C. Timed compatibility analysis of web service composition: a modular approach based on petri nets[J]. IEEE Transactions on Automation Science & Engineering, 2014, 11(2): 594-606.
- [12] GUIDARA I, GUERMOUCHE N, JMAIEL M, et al. Time-dependent QoS aware best service combination selection[J]. International Journal of Web Services Research, 2015, 12(2): 1-25.
- [13] 叶恒舟, 李陶深, 关云慧. 基于时序约束分解的QoS感知的Web服务组合[J]. 电子学报, 2017, 45(5): 1150-1157.
YE Heng-zhou, LI Tao-shen, GUAN Yun-hui. QoS-aware web service composition based on temporal constraints decomposition[J]. Acta Electronica Sinica, 2017, 45(5): 1150-1157.
- [14] 汪潇洒, 付晓东, 刘骊, 等. 随机QoS感知的Web服务组合概率分析[J]. 计算机工程与应用, 2017, 53(14): 70-75.
WANG Xiao-sa, FU Xiao-dong, LIU Li, et al. Probabilistic analysis for stochastic QoS of web service composition [J]. Computer Engineering and Applications, 2017, 53(14): 70-75.
- [15] WANG D D, YANG Y, MI Z Q. A genetic-based approach to web service composition in geo-distributed cloud environment[J]. Computers and Electrical Engineering, 2015, 43: 129-141.
- [16] GUO X, CHEN S S, ZHANG Y W, et al. Service composition optimization method based on parallel particle swarm algorithm on spark[J]. Security & Communication Networks, 2017: 10.1155/2017/9097616.
- [17] MEZNI H, SELLAMI M. A negotiation-based service selection approach using swarm intelligence and kernel density estimation[J]. Software Practice & Experience, 2018, 48(6): 1285-1311.
- [18] BAO L, ZHAO F, SHEN M Q, et al. An orthogonal genetic algorithm for QoS-aware service composition[J]. Computer Journal, 2016, 59(12): 1857-1871.
- [19] DING Z J, SUN Y Q, LIU J J, et al. A genetic algorithm based approach to transactional and QoS-aware service selection[J]. Enterprise Information Systems, 2015, 11(3): 339-358.
- [20] 王尚广, 孙其博, 杨放春. 基于全局QoS约束分解的Web服务动态选择[J]. 软件学报, 2011, 22(7): 1426-1439.
WANG Shang-guang, SUN Qi-bo, YANG Fang-chun. Web service dynamic selection by the decomposition of global QoS constraints[J]. Journal of Software, 2011, 22(7): 1426-1439.
- [21] HWANG S Y, HSU C C, LEE C H. Service selection for web services with probabilistic QoS[J]. IEEE Transactions on Services Computing, 2017, 8(3): 467-480.
- [22] SUN S X. A decomposition-based approach for service composition with global QoS guarantees[J]. Information Sciences, 2012, 199(15): 138-153.
- [23] SURIANARAYANAN C, GANAPATHY G, SETHUNARAYANAN M. An approach for selecting best available services through a new method of decomposing QoS constraints[J]. Service Oriented Computing & Applications, 2014, 9(2): 1-32.
- [24] LIU Z Z, XUE X, SHEN J Q, et al. Web service dynamic composition based on decomposition of global QoS constraints[J]. International Journal of Advanced Manufacturing Technology, 2013, 69(9-12): 2247-2260.
- [25] YE H Z, LI T S, JING C. Decomposition of global constraints for QoS-aware web service composition[J]. International Journal of Innovative Computing, Information and Control, 2016, 12(6): 2053-2066.
- [26] GUIDARA I, JAOUHARI I A, GUERMOUCHE N. Dynamic selection for service composition based on temporal and QoS constraints[C]//IEEE International Conference on Services Computing. [S.l.]: IEEE, 2016: 267-274.
- [27] AI-MASRI E, MAHMOUD Q H. Discovering the best web service[C]//International Conference on World Wide Web(WWW). New York: ACM, 2007: 1257-1258.
- [28] AI-MASRI E, MAHMOUD Q H. QoS-based discovery and ranking of web services[C]//International Conference on Computer Communications and Networks(ICCCN). New York: IEEE, 2007: 529-534.