

基于时序约束分解的 QoS 感知的 Web 服务组合

叶恒舟^{1,2}, 李陶深³, 关云慧²

(1. 广西大学电气工程学院, 广西南宁 530004; 2. 桂林理工大学信息科学与工程学院, 广西桂林 541000;
3. 广西大学信息与工程学院, 广西南宁 530004)

摘要: 基于时序约束的 QoS 感知的 Web 服务组合 (TC_QSC) 问题是在考虑时序约束的基础上寻找满足 QoS 约束或效用最大化的 Web 服务组合问题, 受到了越来越多的关注. 本文提出了一种时序约束分解方法, 把施加于整个或部分工作流的时序约束分解为施加于每个活动的局部时序约束, 从而将 TC_QSC 问题转换为一般的 QoS 感知的 Web 服务组合 (QSC) 问题, 并通过过滤不满足局部时序约束的候选服务, 一定程度上减小原问题的规模. 这种时序约束分解过程主要依赖于 workflow 及其涉及的活动, 而与各活动的候选服务关联不大, 复杂度较低. 实验测试了该方法的效果与时间开销, 验证了其对于局部优选算法的必要性.

关键词: 时序约束; 约束分解; QoS 感知; Web 服务组合; 贪心算法

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2017)05-1150-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.05.018

QoS-Aware Web Service Composition Based on Temporal Constraints Decomposition

YE Heng-zhou^{1,2}, LI Tao-shen³, GUAN Yun-hui²

(1. College of Electrical Engineering, Guangxi University, Nanning, Guangxi 530004, China;
2. College of Information Science and Engineering, Guilin University of Technology, Guilin, Guangxi 541000, China;
3. College of Information and Engineering, Guangxi University, Nanning, Guangxi 530004, China)

Abstract: The problem of Temporal Constraints based QoS-aware web Service Composition (TC_QSC) is becoming increasingly critical in recent years. Its objective is to select the Web service composition which maximizes the QoS utility while preserving QoS constraints and temporal constraints. This paper proposes a temporal constraints decomposition method that formulates the TC_QSC problem into a general QoS-aware Web service composition (QSC) problem by means of decomposing the temporal constraints imposed on the whole or part of the workflow to a set of local temporal constraints for each activity. So the problem scale is reduced by filtering out those candidate web services violating local temporal constraints. The temporal constraints decomposition process is mainly dependent on the workflow and activities involved in it, while it has little to do with the candidate services. Therefore, its complexity is lower. The experiments have demonstrated the effectiveness and time cost of the proposed method and verify its necessity to local selection algorithm.

Key words: temporal constraint; constraint decomposition; QoS-aware; Web service composition; greedy algorithm

1 引言

近年来, 很多研究致力于解决 QoS 感知的 Web 服务组合 (QoS-aware Web service composition, QSC) 优化问题. 这些研究可分为两类: 一类认为 workflow 是未知的, 需要探究服务之间的依赖关系^[1,2]; 另一类认为 workflow 已知, 重点关注服务的优化选择问题^[3-14]. 本文的研究属

于后一类.

一般 QSC 问题是 NP 难的^[3], 简单的全局搜索算法将导致指数级的时间复杂度. 基于贪心策略的局部优选策略^[4-6,15]具有很好的时空性能, 但不能保证全局约束. 另一类研究将 QSC 问题建模为一个已知模型, 如整数线性规划 (ILP)^[7]、多维多选择背包问题 (MMKP)^[6]、混合整数线性规划 (MILP)^[3]等, 这些模型

的求解仍面临着指数级时间复杂度的挑战. 进化类算法(如遗传算法、粒子群算法)^[8-12,16]也广泛应用于求解 QSC 问题,这类算法适应性强,容易满足全局约束,可平衡时间开销与优化效果,但存在收敛速度慢、稳定性不强等缺点.

为了兼顾局部优选和全局统筹的优点,采用全局 QoS 约束分解以降低时间复杂度成为了一种有效的选择. 文献[13]先将全局 QoS 约束分解为满足用户偏好的局部约束,然后利用局部最优策略寻找合适的组合服务,但其不能保证全局约束,也不能直接支持复杂 workflow. 文献[14]采用遗传算法来求解 QoS 约束分解(QCD)模型,以确定每个活动的每个 QoS 属性参考值,在进行局部选择时选择与这些参考值最为接近的候选服务,该方法支持复杂 workflow,因采用遗传算法,效率偏低,也不能保证全局约束. 文献[17]也基于分解思想,但它分解的不是约束而是全局效用,是通过寻找全局效用与局部效用之间的关联逐步调整局部选择方案来实现的,体现的是一种贪心策略. 文献[18]提出了一种把全局 QoS 约束分解为局部约束的模型,并采用了一种改进的遗传算法求解该模型,但该文只讨论了模型的优化目标而没有讨论相关约束条件,其实验也仅针对包含 9 个活动的给定 workflow. 比较而言,本文所采用的时序约束分解(TCD)模型建立在包含顺序、并发、选择等模式的常见 workflow 之上,其目标是获取每个活动应满足的时序约束,并使每个活动满足时序约束的候选服务的个数尽量多且分布均衡,并保证当局部约束得到满足时,全局约束也一定可以满足.

在一些业务流程中会带有各种时序约束^[19,20]. 如图 1 所描述的某电子产品的研发流程(A1 到 A10 分别代表“需求分析”、“主板设计”、“外设设计”、“硬件测试”、“软件系统分析”、“软件外包”、“升级现有软件”、“重构软件”、“软件测试”、“软硬件集成测试”)可能存在如下时序约束:“整个研发时间不超过 12 个月,软硬件开发的时间不超过 4 个月,主板设计时间不超过 1 个月,且应在软件系统设计之前完成”. 这些约束可符号化为: $t(1_14) \leq 12, t(2_13) \leq 4, t(3_3) \leq 1, t(6_12) \leq 3$,其中 $t(i_j)$ 表示完成从 i 到 j 之间的流程需要的时间. 尽管 QoS 也把服务的响应时间作为考虑对象,但它仅包含一条全局约束,而时序约束往往是覆盖全局及部分的多个约束,因此一些研究关注于时序约束. 这些研究除了关注时间本体^[21]、时序约束描述^[22,23]等基础理论外,更多的是关注时序约束的兼容性验证^[19,20,23-26],而很少考虑时序约束的服务组合问题. 本文的目标综合时序约束与 QSC 问题,采用 TCD 方法把 TC_QSC 问题转化为一般的 QSC 问题.

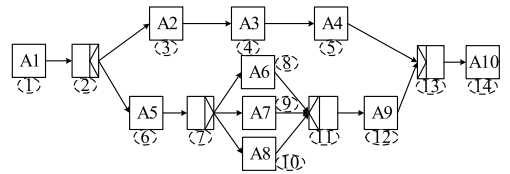


图1 一个 workflow 例子

2 工作流模型

一个组合服务可由 workflow 描述,而一个 workflow 由一些活动及模式构成. 在 workflow 中,一个活动代表具有相同功能、不同 QoS 的一组 Web 服务,这组服务称为该活动的候选服务集;模式用于描述活动之间的依赖关系,比较常见的模式有:顺序(SEQ)、并发(AND)、选择(XOR)^[3]. 本文所讨论的 workflow 是比较常见的一类 workflow,它们由这三种模式经过有限次连接与嵌套而成(见定义 1).

定义 1 设 \odot 为 SEQ、AND 或 XOR 中的任意一种, workflow 可以递归定义如下:

- (1) 若 A_1, A_2 是活动,则 $A_1 \odot A_2$ 是 workflow;
- (2) 若 A_1 是活动、 F_1 是 workflow,则 $A_1 \odot F_1$ 及 $F_1 \odot A_1$ 是 workflow;
- (3) 若 F_1, F_2 是 workflow,则 $F_1 \odot F_2$ 是 workflow;
- (4) 若 F_1, F_2 是 workflow,且 F_1 是 F_2 的一部分,则称 F_1 是 F_2 的子 workflow.

由定义 1 可知,一个 workflow F 可与一棵二叉树 T 对应,叶子节点为 F 涉及的所有活动,非叶子节点为 F 涉及的所有模式,各个子树代表 F 的一个子 workflow,树根为 F .

显然,我们可以得到以下两个性质:

- (1) 设 T 是 F 对应的一棵二叉树, v 为 T 的一个非叶子节点,则 v 的左子树($v.lc$)和 v 的右子树($v.rc$)均不为空;
- (2) 设 T 是 F 对应的一棵二叉树, r 为 T 中叶子节点的个数, s 为 T 中非叶子节点的个数,则 r 为 F 中涉及的活动个数,且 $r = s + 1$.

通常, F 的时序约束施加于 F 的各个子 workflow,因此 F 的时序约束施加于 T 的各个节点,用 $t(v) \leq \tau_v$ 表示 T 的节点 v 应满足的时序约束. 例如图 1 所示的 workflow 可以用图 2 所示的二叉树表示,相应的时序约束可表示为 $t(v19) \leq 12, t(v15) \leq 4, t(v1) \leq 1, t(v14) \leq 3$.

3 TCD 模型

TCD 的思想是将施加于 F 的各个子 workflow 上的时序约束分解为针对各个活动的时序约束,再从各个活动中除去不满足时序约束的候选服务,从而将 TC_QSC

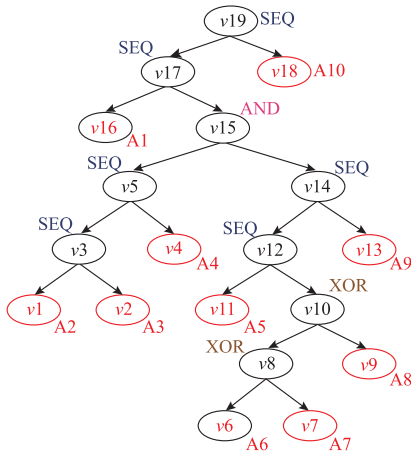


图2 用二叉树表示的工作流

问题转变为 QSC 问题,同时也降低了候选服务集的规模。

设 F 涉及活动 $A = \{a_1, a_2, \dots, a_n\}$, 活动 $a_i (i = 1, 2, \dots, n)$ 满足时序约束 $t(a_i) \leq x_i$, 其候选服务集为 $CS(a_i) = \{c_{i1}, c_{i2}, \dots, c_{im}\}, c_{ij}$, t 表示完成 c_{ij} 所需要的时间, $p_i (i = 1, 2, \dots, n)$ 为集合 $\{c \mid c \in CS(a_i) \ \&\& \ c.t \leq x_i\}$ 中元素的个数, $P = \{p_1, p_2, \dots, p_n\}$ 的均值与方差分别为 $E(P)$ 和 $D(P)$, 则时序约束分解的目标是寻找 x_1, x_2, \dots, x_n , 使其满足 F 的时序约束, 且使 $E(P) - \alpha D(P)$ 最大化, 其中 α 是可调节的正常数。

下面讨论时序约束转化问题, 即把施加于 T 的非叶子节点的约束转化为用叶子节点表示的约束。设 v 是 T 的一个非叶子节点, 给出的转换规则如下:

(1) 若 v 代表 SEQ 模式, 则 $t(v) \leq \tau_v \Leftrightarrow t(v.lc) + t(v.rc) \leq \tau_v$;

(2) 若 v 代表 AND 模式, 则 $t(v) \leq \tau_v \Leftrightarrow \max(t(v.lc), t(v.rc)) \leq \tau_v$;

(3) 若 v 代表 XOR 模式, 则 $t(v) \leq \tau_v \Leftrightarrow \min(t(v.lc), t(v.rc)) \leq \tau_v$;

因此, 对于施加于 T 的非叶子节点 v 上的约束, 可以用算法 1 (TC_Transfer) 转化为用叶子节点表示的约束, 即可将 $t(v) \leq \tau_v$ 转化为 $f(x_1, x_2, \dots, x_n) \leq \tau_v$ 的形式。该算法仅需对以 v 为根的子树作一次遍历, 其时空复杂度为 $O(n)$ 。

算法 1 TC_Transfer (Node v)

input: T and v

output: The value of $f(x_1, x_2, \dots, x_n)$

if v is a leaf node

return x_v ;

if pattern(v) is SEQ then

return $TC_Transfer(v.lc) + TC_Transfer(v.rc)$;

if pattern(v) is AND then

return $\max(TC_Transfer(v.lc), TC_Transfer(v.rc))$;

if pattern(v) is XOR then

return $\min(TC_Transfer(v.lc), TC_Transfer(v.rc))$;

我们以 $t(v14) \leq 3$ 为例观察算法 1 的工作过程。

① $v14$ 是 SEQ 节点, $t(v14)$ 转化为 $t(v12) + t(v13)$;

② $v13$ 是叶子节点, $t(v13)$ 转化为 x_9 ;

③ $v12$ 是 SEQ 节点, $t(v12)$ 转化为 $t(v11) + t(v10)$;

④ $v11$ 是叶子节点, $t(v11)$ 转化为 x_5 ;

⑤ $v10$ 是 XOR 节点, $t(v10)$ 转化为 $\min(t(v8), t(v9))$;

⑥ $v9$ 是叶子节点, $t(v9)$ 转化为 x_8 ;

⑦ $v8$ 是 XOR 节点, $t(v8)$ 转化为 $\min(t(v6), t(v7))$;

⑧ $v6, v7$ 是叶子节点, $t(v6), t(v7)$ 分别转化为 x_6, x_7 ;

最终, $t(v14) \leq 3$ 转换为 $x_5 + \min(\min(x_6, x_7), x_8) + x_9 \leq 3$ 。

由此, 对于工作流 F , 含有 k 个时序约束的 TCD 模型可表示为:

object: $\max(E(P) - \alpha D(P))$

s. t. $f_i(x_1, x_2, \dots, x_n) \leq \tau_i, i = 1, 2, \dots, k$

$\min(c_{ij}, t) \leq x_i \leq \max(c_{ij}, t), i = 1, 2, \dots, n$

其中 $\min(c_{ij}, t), \max(c_{ij}, t)$ 分别表示 a_i 的候选服务所需最小与最大完成时间。

定理 1 设 T 有 m 个非叶子节点 $v_1, v_2, \dots, v_m, v_m$ 为根节点, $v_i (i = 1, 2, \dots, m)$ 对应的全局约束为 $t(v_i) \leq \tau_v$ (若 v_i 无全局约束, 取 τ_v 为 $+\infty$)。设 x_1, x_2, \dots, x_{m+1} 为满足 TCD 约束条件的一组解, 服务序列 $\{c_{1p_1}, c_{2p_2}, \dots, c_{m+1}, p_{m+1}\}$ 为 F 的一个组合实例, 且 $c_{ip_{m+1}}.t \leq x_i (i = 1, 2, \dots, m+1)$, 则该组合实例满足 $rt(v_i) \leq t(v_i) (i = 1, 2, \dots, m)$, 其中 $rt(v_i)$ 表示 v_i 所需的完成时间。

证明: 对 m 运用数据归纳法。

(1) 当 $m = 1$ 时, T 仅有一条约束 $rt(v_1) \leq t(v_1)$ 。

若 v_1 的模式为 SEQ, 按算法 1 有 $t(v_1) = x_1 + x_2$ 按 QoS 属性聚合规则,

$rt(v_1) = c_{1p_1}.t + c_{2p_2}.t \leq x_1 + x_2 = t(v_1)$

当 v_1 的模式为 AND 或 XOR, 类似可证结论正确。

(2) 假设当 $m < k (k \geq 2)$ 时结论成立。当 $m = k$ 时, v_m 的左、右子树的非叶子节点个数均小于 k , 对这两棵子树应用假设可知 $rt(v_i) \leq t(v_i) (i = 1, 2, \dots, m-1)$ 成立。因此, 对于 F 的组合实例, 只需验证 $rt(v_m) \leq t(v_m)$ 。

若 v_m 的模式为 SEQ, 按算法 1 有

$t(v_m) = t(v_{m.rc}) + t(v_{m.lc})$

按 QoS 属性聚合规则,

$$rt(v_m) = rt(v_{m.rc}) + rt(v_{m.lc}) \leq t(v_{m.rc}) + t(v_{m.lc}) = t(v_m)$$

当 v_m 的模式为 AND 或 XOR, 类似可证结论正确.

综上可知, 定理 1 正确. [证毕]

定理 1 表明, 在组合服务时, 只要所挑选的候选服务满足由 TCD 模型得到的局部时序约束, 所得组合实例就可满足全局时序约束.

4 TCD 算法

若约定 x_i 只能是单位时间的整数倍, TCD 模型可以看作一个整数规划问题. 本文提出一种贪心算法(记为 GTCD)求解该模型, 由算法 2 描述, 其中 StepNumber 是预设的一个常整数, 用来确定每次调整的步长. GTCD 算法的核心思想是: 给全部 x_i 赋予一个较合理的初值, 以一个时间刻度递增或递减全部 x_i , 直到其达到一种临界违约状态, 再通过不断调整部分 x_i 的方式优化结果. 前者的调整方法保障了公平性, 后者则是为了尽可能多的保留候选服务. 该算法包含两个顺序的循环操作, 若记 $r = \text{StepNumber}$, 第一个循环在最坏情况下执行次数不超过 r 次, 时间复杂度为 $O(r)$; 后一个循环最坏情况下执行次数不超过 $n * r$ 次, 时间复杂度为 $O(n * r)$; 整个算法的时间复杂度为 $O(n * r)$.

算法 2 GTCD

```

input: a TC_QSC problem and its temporal constraints decomposition model
output: a vector  $x$  representing the local temporal constraints for each activity
init  $x_i = \text{avg}(c_{ij}, t)$ ;
init  $\Delta x_i = (\max(c_{ij}, t) - \min(c_{ij}, t)) / \text{StepNumber}$ ;
if ( $x$  is not satisfied) then
  do
     $x_i = x_i - \Delta x_i$ ;
  while ( $x$  is not satisfied);
else
  do
     $x_i = x_i + \Delta x_i$ ;
  while ( $x$  is satisfied);
 $x_i = x_i - \Delta x_i$ ;
end if
while ( $x$  is satisfied) do
  find  $i$  when  $x_i$  adds  $\Delta x_i$ , the added value of object is the largest;
  if ( $i$  can't be found) then
    return  $x$ ;
   $x_i = x_i + \Delta x_i$ ;
end while
 $x_i = x_i - \Delta x_i$ ;
return  $x$ ;

```

5 子 workflow 局部最优算法

当从各活动中删除不能满足局部时序约束的候选服务后, 原 TC_QSC 问题就转化成一般的 QSC 问题了. QSC 问题已经存在很多算法, 这不是本文的研究重点. 为了方便测试 TCD 效果, 本文采用了一种子 workflow 局部最优选择(SFLO)算法. 与常见的针对每个活动独立优选的局部优选算法不同, SFLO 算法对 workflow 中的每个子 workflow 按从小到大的顺序依次优选出 QoS 效用最优的组合方案. workflow F 的每个子 workflow 恰好对应二叉树 T 中的一个子树, 而 F 刚好对应 T 的根结点. 若将 T 中每个非叶子节点看成一个虚拟活动(VA), 其候选服务集可表示为 $CS(VA) = \{c_1 \odot c_2 \mid c_1 \in CS(VA.lc) \ \&\& \ c_2 \in CS(VA.rc)\}$, 其中 \odot 为该节点所代表的模式, 它决定了两个服务组合时各 QoS 属性的聚合方式. $CS(VA)$ 中的每个元素代表 VA 的一种组合方案, SFLO 算法就是从 $CS(VA)$ 中优选出 QoS 综合效用最高的组合方案(如算法 3). 设 F 中包含 n 个活动, 每个活动包含 m 个候选方案. 对于根节点 $root$, $CS(root)$ 的计算要求遍历整个二叉树, 其时间复杂度为 $O(n)$; 而访问每个节点的时间开销主要是在 $m * m$ (左右孩子均是叶子节点) 或 m (左右孩子恰有一个是叶子节点) 或 1 (左右孩子均是非叶子节点) 个候选组合方案中选择最大值, 其最大时间复杂度为 $O(m \log_2 m)$.

算法 3 SFLO (Node v)

```

input:  $T.root$ 
output: the local optimal composition service for  $v$ 
if ( $v$  is a leaf)
  return null;
else if ( $v.lc$  is a leaf &&  $v.rc$  is a leaf) then
  calculate  $CS(v) = \{c_1 \odot c_2 \mid c_1 \in CS(v.lc) \ \&\& \ c_2 \in CS(v.rc)\}$ ;
  return the element of  $CS(v)$  with maximum QoS utility;
else if ( $v.lc$  is not leaf) then
  return SFLO( $v.lc$ );
else if ( $v.rc$  is not leaf) then
  return SFLO( $v.rc$ );

```

6 实验分析

实验代码用 Java 7 编写, 运行环境为 Intel (R) Core (TM) i5-2430M 2.4 GHz Intel Xeon 双核处理器, 4 GB 内存及 Win7 操作系统. 测试时, workflow 根据活动总数随机生成, 其中 AND 与 XOR 模式各占总模式的 1/4, 其余为 SEQ 模式. 测试数据集来源于 QWS 数据库^[27,28], 该库中的每条记录包含了 9 个 QoS 属性, 实验时, 选取了响应时间、可靠性、吞吐率等 3 个属性, 每个候选服务的 QoS 属性由从该库中随机选择的一条记录

确定. QoS 属性值的标准化方法及聚合方式与文献[12]一致. 对于由活动个数及候选服务个数所确定的每种规模, 采用多次实验(20次)求平均值策略. 参数 $\alpha = 1$ 、StepNumber = 50, 时序约束条数为 T 中非叶子节点个数的 0.3 倍.

参数 base 用来衡量约束的强度. 以某活动的所有候选服务的平均完成时间为该活动的参考, 计算每个子工作流的参考时间; 若某个子工作流有时序约束, 则设定该约束值为参考时间的 $1 + \text{base}$ 倍. 可见, base 的值越大, 强度越小.

6.1 TCD 分解的效果

理想的 TCD 分解效果应该是 $E(P)$ 尽可能大, $D(P)$ 尽可能小. 图 3、5 显示了不同约束强度下 $E(P)/m$ 和 $D(P)/m$ 随候选服务个数变化的情况, 图 4、6 则显示了它们随活动个数变化的情况. 从图中可以观察到: (1) $E(P)$ 、 $D(P)$ 与服务个数或活动个数关联不大, 算法稳定性好. (2) $E(P)$ 随约束强度的增强而明显减小, 当约束松弛到一定程度后 ($\text{base} \geq 0.4$) 它将趋于稳定. 这是因为约束越强, 能满足约束条件的候选服务越少, $E(P)$ 也就越小; 当约束足够松弛后, 几乎所有候选服务都能满足约束条件, $E(P)$ 将接近最大值. (3) $D(P)$ 也明显受约束强度影响, 当 $\text{base} > 0$ 时, $D(P)/m$ 很小, 之后随约束强度的增强而增大, 但保持在小于 4 的水平, 可见算法具有较好的公平性.

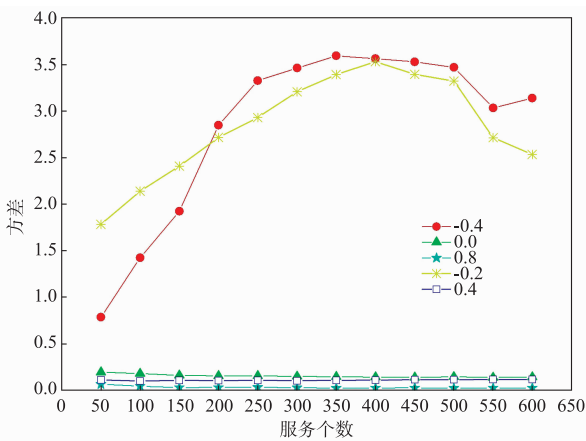


图3 方差随服务个数变化情况 (活动个数为100)

6.2 TCD 分解的必要性

现有的基于时序约束的服务组方法往往是在不(完全)考虑时序约束的前提下寻找组合方案, 然后验证该组合的时序兼容性. 当时序约束较强时, 这类方法找到的组合方案很可能不满足时序兼容性. 下面的实验将验证这一点. 实验时, 考虑两种 QSC 算法: 本文的算法 (SFLO) 及每个活动独立选择候选服务的局部优选法 (记为 ALO). QoS 的综合效用由服务完成所需时间、可靠性与吞吐率的标准化属性值的加权平均度量, 权值依次为

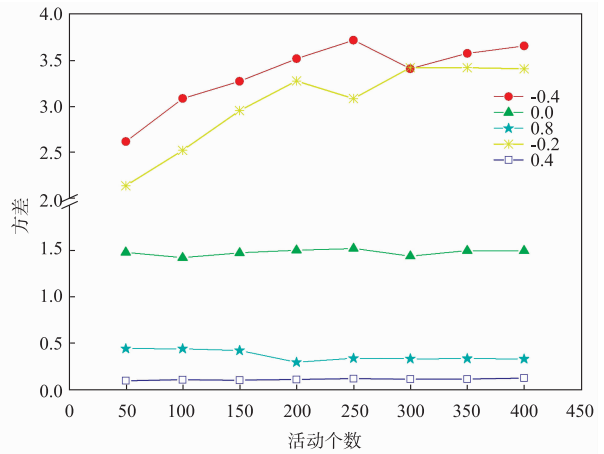


图4 方差随活动个数变化情况(服务个数300)

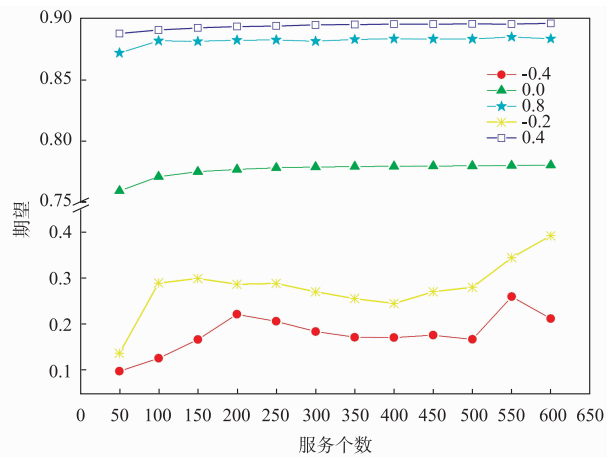


图5 期望随服务个数变化情况(活动个数为100)

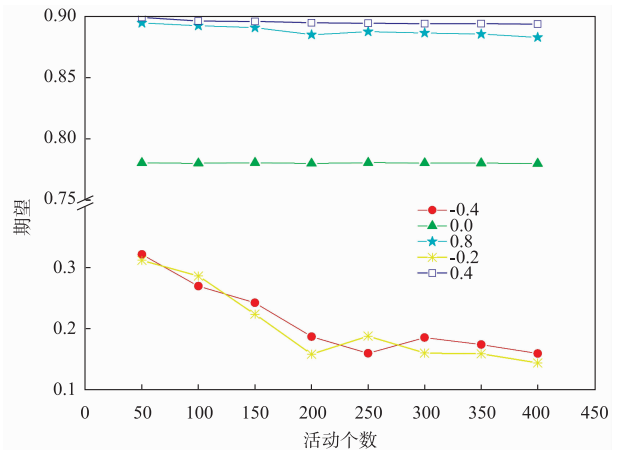


图6 期望随活动个数变化情况(服务个数300)

0.4、0.3、0.3. 如果先用 TCD 分解得到的局部时序约束筛选候选服务, 再用上述任何一种算法寻找的组合方案, 所得结果都是满足时序兼容性. 如果不经 TCD 分解并筛选候选服务, 而是直接寻找组合方案, 这些方案能够满足时序兼容性的概率如图 7 所示. 其中, G1 ~ G3 代表三种

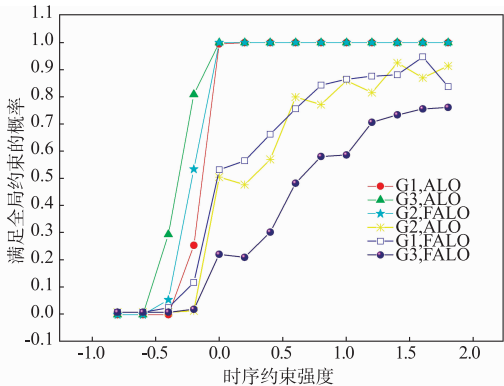


图7 约束满足概率

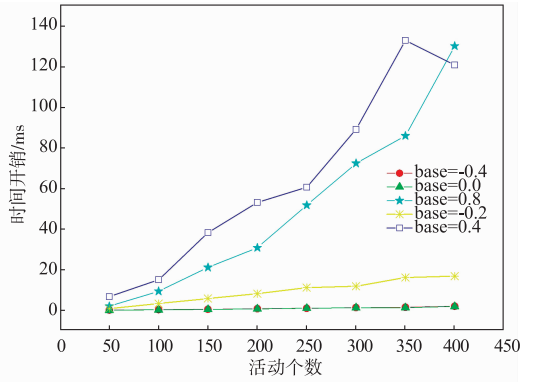


图9 时间开销随活动个数变化(服务个数300)

问题规模(活动个数分别为 100、100、200, 每个活动的候选服务个数分别为 200、400、400). 由图可知:(1) 约束满足概率与问题规模关系不大;(2) FALO 算法更能体现出约束强度对约束满足概率的影响,这是因为 FALO 考虑了活动之间的连接模式及相关的聚合方式,在一定程度上体现约束的全局性;(3) 无论是 ALO 或 FALO,当 $base < -0.4$ 后,可满足率均接近于 0,而此时 $E(P)/m$ 约为 0.2(见图 5、6),完全可以找到可行解. 可见,当约束强度较大,使用局部优选策略求解 TC_QSC 问题时,应先应用 TCD 分解策略.

6.3 TCD 分解的时间开销

我们测试了活动个数、服务个数、约束强度、约束个数等因素对 TCD 分解的时间开销的影响. 图 8 表明服务个数对算法的时间开销影响较弱;图 9 表明时间开销随活动个数的增加呈线性增加方式,增加的速度受约束强度的影响较为明显;图 10 表明约束个数对时间开销无明显影响. 图 8、9 均说明约束强度对时间开销有显著影响:当 $base \leq 0$ 时,时间开销极小;当 $base \geq 0.8$ 时,时间开销较大,且趋于稳定. 这是因为 GTCD 算法以步长 Δx_i 增大 x_i ,当约束较强时,可调整幅度较小,所以时间开销较小.

6.4 性能评估

本节比较评估了 TCD 模型(采用 GTCD 求解)、

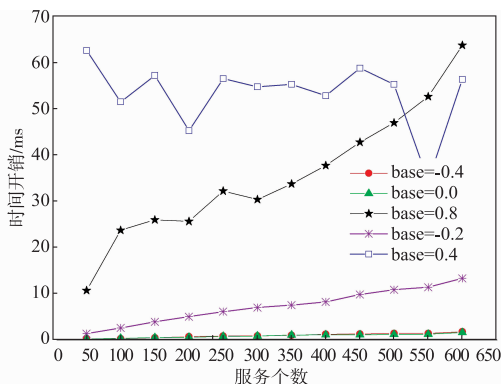


图8 时间开销随服务个数变化(活动个数为100)

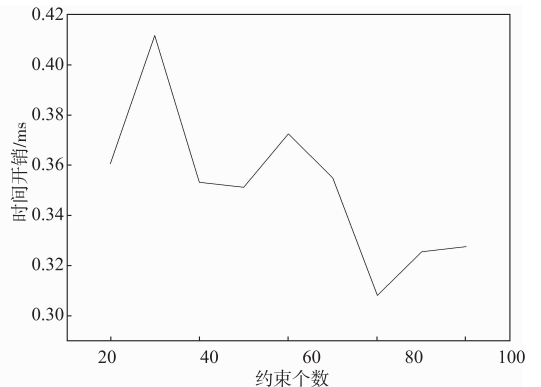


图10 时间开销随约束条数变化

MMKP^[6](采用 CGA^[18]求解)及 MILP^[3](采用 CPLEX 求解)的优化效果与时间开销. 由于 MILP 要求目标函数可线性化, TCD 模型仅分解时序约束, MMKP 及 MILP 均未讨论覆盖部分活动的时序约束,服务组合时以执行价格最大化为目标,且满足全局时序约束. 候选服务的执行价格在 [50, 200] 间随机取整数,响应时间在 [10, 50] 间随机取整数;问题的规模与 6.2 节一致; $base = 0.2$; 其它参数与相关文献一致. 表 1 对比了三种模型求取组合服务的时间开销及求得的组合服务的执行价格(以 MILP 为参照). 实验结果表明:本文的方法在时间开销方面优势显著,在执行价格方面略逊于 MILP.

表 1 算法的时间开销与执行价格对比

问题规模	时间开销 (ms)			执行价格		
	MILP	MMKP	GTCD	MILP	MMKP	GTCD
G1	334	1151	24	1	1.024	1.002
G2	422	1161	50	1	1.033	1.002
G3	982	2432	212	1	1.03	1.002

7 结论

时序约束是 Web 服务组合时需要考虑的一个重要因素,现有的大多数方法仅关注时序约束的兼容性验证. 本文提出了一种时序约束服务组合方法,可将全局

约束分解为针对每个活动的局部约束,从而在进行服务选择时,只要能保证这些局部约束,就可以保证全局约束.建立的 TCD 模型以最大化 $E(P)$ 、最小化 $D(P)$ 为目标,以保证全局约束为条件,其规模与活动个数、约束个数有关,与服务个数无关,其复杂性远低于 MMKP、MILP 等模型.实验结果表明,用于求解 TCD 模型的 GTCD 算法具有较好的算法效果、较低的时间复杂度以及较高的稳定性.

参考文献

- [1] Zou Guobing, Lu Qiang, Chen Yixin, et al. QoS-aware dynamic composition of web services using numerical temporal planning [J]. *IEEE Transactions on Services Computing*, 2014, 7(1): 18 – 31.
- [2] Wang Pengwei, Ding Zhijun, Jiang Changjun, et al. Constraint-aware approach to web service composition [J]. *IEEE Transactions on Systems, Man, and Cybernetics: System*, 2014, 44(6): 770 – 784.
- [3] Virginie G, Maude M, Cécile M. Web services composition: complexity and models [J]. *Discrete Applied Mathematics*, 2015, 196(2): 67 – 82.
- [4] Mohammad A, Thomas R. Combining global optimization with local selection for efficient QoS-aware service composition [A]. *World Wide Web Conference Series [C]*. New York: ACM Press, 2009. 881 – 890.
- [5] Yu Tao, Zhang Yue, Lin Kwei-Jay. Efficient algorithms for web services selection with end-to-end QoS constraints [J]. *ACM Transactions on the Web*, 2007, 1(1): 1 – 26.
- [6] Luo Yuansheng, Qi Yong, Hou Di. A novel heuristic algorithm for QoS-aware end-to-end service composition [J]. *Computer Communications*, 2011, 34(9): 1137 – 1144.
- [7] Zeng Liangzhao, Boualem B, Anne H H N, et al. QoS-aware middleware for web services composition [J]. *IEEE Transactions on Software Engineering*, 2004, 30(5): 311 – 327.
- [8] Palanikkumar D, Kousalya G. An evolutionary algorithmic approach based optimal web service selection for composition with quality of service [J]. *Journal of Computer Science*, 2012, 8(4): 573 – 578.
- [9] Erdinc Y A, Pinar K. Improved genetic algorithm based approach for QoS aware web service composition [A]. *IEEE International Conference on Web Services [C]*. New York: IEEE Press, 2014. 463 – 470.
- [10] Wang Dandan, Yang Yang, Mi Zhenqiang. A genetic-based approach to web service composition in geo-distributed cloud environment [J]. *Computers and Electrical Engineering*, 2015, 43(C): 129 – 141.
- [11] Liao Jianxin, Liu Yang, Wang Jingyu, et al. Service composition based on niching particle swarm optimization in service overlay networks [J]. *KSI Transactions on Internet and Information Systems*, 2012, 6(4): 1106 – 1127.
- [12] Wang Shangguang, Sun Qibo, Zou Hua, et al. Particle swarm optimization with skyline operator for fast cloud-based web service composition [J]. *Mobile Networks and Applications*, 2013, 18(1): 116 – 121.
- [13] 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 Shangguang, Sun Qibo, Yang Fangchun. Web service dynamic selection by the decomposition of global QoS constraints [J]. *Journal of Software*, 2011, 22(7): 1426 – 1439. (in Chinese)
- [14] Farhad M, Naser N B, Kamran Z, et al. QoS decomposition for service composition using genetic algorithm [J]. *Applied Soft Computing*, 2013, 6(5): 3409 – 3421.
- [15] 尹浩, 张长胜, 张斌等. 一种求解 SLA 等级感知服务组合问题的多目标离散粒子群优化算法 [J]. *电子学报*, 2014, 42(10): 1983 – 1990.
Yin Hao, Zhang Changsheng, Zhang Bin, et al. A multi-objective discrete particle swarm optimization algorithm for SLA-aware service composition problem [J]. *Acta Electronica Sinica*, 2014, 42(10): 1983 – 1990. (in Chinese)
- [16] 徐猛, 崔立真, 李庆忠. 基于扩展图规划的 Top-k 服务组合方法研究 [J]. *电子学报*, 2012, 40(7): 1404 – 1409.
Xu Meng, Cui Lizhen, Li Qingzhong. An extended Graph-planning based Top-k service composition method [J]. *Acta Electronica Sinica*, 2012, 40(7): 1404 – 1409. (in Chinese)
- [17] Sherry X S, Jing Zhao. A decomposition-based approach for service composition with global QoS guarantees [J]. *Information Sciences*, 2012, 199(15): 138 – 153.
- [18] Liu Zhizhong, Xue Xiao, Shen Jiquan, et al. Web service dynamic composition based on decomposition of global QoS constraints [J]. *Int J Adv Manuf Technol*, 2013, 69(9): 2247 – 2260.
- [19] Nawal G, Olivier P, Christophe R. Timed specification for web services compatibility analysis [J]. *Electronic Notes in Theoretical Computer Science*, 2008, 200(3): 155 – 170.
- [20] Du Yanhua, Tan Wei, Zhou Mengchu. Timed compatibility analysis of Web service composition: a modular approach based on Petri nets [J]. *IEEE Transactions on Automation Science and Engineering*, 2014, 11(2): 594 – 606.
- [21] Jerry R H, Feng Pan. An ontology of time for the semantic web [J]. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2004, 3(1): 66 – 85.
- [22] Chen Jinjun, Yang Yun. Localizing temporal constraints in scientific workflows [J]. *Journal of Computer and System*

Sciences,2010,76(2):464-474.

- [23] Zhou Zhangbin, Laurence T Y, Sami B, et al. Verifying mediated service interactions considering expected behaviors[J]. Journal of Network and Computer Applications, 2011,34(4):1043-1053.
- [24] 杜彦华, 范玉顺. 基于生成图的工作流多过程动态时序一致性验证方法[J]. 电子学报, 2009, 37(10):2181-2186.
Du Yanhua, Fan Yushun. A sprouting graph based approach to dynamic check the temporal consistency of workflow multi-process[J]. Acta Electronica Sinica, 2009,37(10):2181-2186. (in Chinese)
- [25] Du Yanhua, Wang Xiaofei, Yao Jianshi. Probability based timed compatibility of Web service composition[J]. Practical Applications of Intelligent Systems, 2012, 124(1):31-39.
- [26] 徐红霞, 杜彦华, 董绍华. 时序约束下 Web 服务组合的兼容性修正研究[J]. 计算机集成制造系统, 2012, 18(11):2562-2572.
Xu Hongxia, Du Yanhua, Dong Shaohua. Study on the compatibility and modification of web service composition with temporal constraints[J]. Computer Integrated Manufacturing System(CIMS), 2012, 18(11):2562-2572. (in Chinese)
- [27] Eyhab A M, Qusay H M. Discovering the best web service [A]. International Conference on World Wide Web (WWW)[C]. New York:ACM Press,2007. 257-1258.

- [28] Eyhab A M, Qusay H M. QoS-based discovery and ranking of Web services [A]. International Conference on Computer Communications and Networks (ICCCN)[C]. New York:IEEE Press,2007. 529-534.

作者简介



叶恒舟 男,1980年出生于湖北汉川,广西大学电气工程学院博士研究生,桂林理工大学信息科学与工程学院副教授,研究方向:Web服务组合、优化设计.

E-mail:18123408@qq.com



李陶深 男,1957年出生于广西南宁,广西大学计算机与电子信息学院教授,博士生导师,研究方向:CAD理论与应用、分布式工程数据库、遗传优化设计、网络计算与信息安全、无线 Mesh 网络.

E-mail:tshli@gxu.edu.cn



关云慧 女,1990年出生于安徽安庆,桂林理工大学硕士研究生,研究方向:Web服务技术、组合优化.