

一种基于任务粒化的服务组合优化方法

张以文, 崔光明, 郭 星, 张燕平

(安徽大学计算智能与信号处理教育部重点实验室, 安徽合肥 230031)

摘 要: 在 big service 背景下, 越来越多的资源以服务的形式发布与使用, 用户需求越来越复杂, 导致服务组合计算规模呈指数级增长. 本文提出一种任务粒化算法 (TgA, Task-granular Algorithm), 用于快速有效地求解大规模服务组合优化问题. 首先, 构建任务粒化分层服务组合模型, 并分析了该模型的计算复杂性; 其次, 根据现有 QoS 属性计算方式, 从理论上分析其在任务粒化过程中的可行性; 最后, 大量仿真实验结果表明, 相比于经典的 PSO 算法, TgA 可以将服务组合优化时间性能提高约 4 至 7 倍, 且寻优精度提高 10% 以上.

关键词: 服务组合; 任务粒化; 分层模型; PSO

中图分类号: TP301

文献标识码: A

文章编号: 0372-2112 (2017) --

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2017.●.●

A Service Composition Optimization Method Based on Task-Granulating

ZHANG Yi-wen, CUI Guang-ming, GUO Xing, ZHANG Yan-ping

(Anhui University, Key Laboratory of Intelligent Computing & Signal Processing; Ministry of Education, Anhui, Hefei 230031, China)

Abstract:

Abstract: Under the background of big service, more and more resources are released and utilized in the form of services. Meanwhile, the users' requirements grow in complexity, which leads to the exponential growth of the service composition calculation scale. In this paper, a task-granulating algorithm called TgA is proposed to solve large-scale service composition optimization problem quickly and effectively. Firstly, we build a hierarchical service composition model based on task granulation, and analyze its computational complexity. Secondly, we analyze the feasibility theoretically during the task-granulating according to the calculation of existing QoS attributes. Finally, a large number of simulation experimental results show that, compared to the classical particle swarm optimization (PSO) algorithm, the proposed algorithm can improve the service composition optimization performance by 4 to 7 times and increase the optimization accuracy by more than 10%.

Key words: service composition (SC); task granulating; hierarchical model; PSO

1 引言

随着 big service 时代的到来^[1], “一切皆服务”的理念使得服务计算得到进一步普及, 以及用户个性化需求不断增加, 服务计算系统呈现出动态性、松耦合、大规模等显著特征. 服务组合问题也呈现出多极性、强约束性、动态性、大规模、高维度等特点, 其计算规模与问题复杂性呈指数级增长^[2,3], 服务组合优化问题变得日益复杂. 然而, 服务组合过程中往往涉及多个服务提供主体, 服务协同已经成为一种常态. 以图 1 所示的用户网上购物过程为例, 用户的一次购物体验不仅与其选择

的购物平台和支付方式有关, 而且还与卖方提供物品包装、配送方式, 以及物流运输方式等相关. 因此, 用户的购物体验不仅包含其自身参与的购物过程, 还包括卖方与物流方多个主体的协同合作过程.

图 1 所示的购物案例中, 卖方提供的服务会影响买方的购物体验, 但从何处仓储取货买方无需关心. 图 1 中, 服务组合系统需要进行 5 个任务的最优化处理, 但与用户直接相关的只有 3 个任务, 另外两个任务由卖方与物流方决定, 从而使用户达到整体购物体验最优. 现有的服务组合研究较少关注这一现象, 尤其在 big service 环境下, 这一现象更加普遍. 为解决这一问题, 本文

提出一种任务粒化服务组合模型,通过将图 1 的 5 个任务进行粒化处理,分解为多个任务粒,通过每个任务粒的局部最优处理从而实现全局最优.

2 服务组合粒化分层模型

定义 1 任务粒(Task-granular)

任务粒是指业务工作流程根据处理需要将原组合服务问题分解为多个简单子服务组合问题的粒化过程,用一个三元组来表示: $Tg = (id, sc', F)$. 其中, id 表示每个任务粒的唯一标识; sc' 表示原服务组合问题 sc 的子集,即 sc 中任务集的子集; F 表示每个任务粒的评估函数.

图 1 所示的网上购物案例,原问题粒表示为 $\{(t_1, t_2, t_3, \dots, t_5)\}$,按照任务执行主体,可以将 5 个任务分为三类,即:买方,卖方以及物流,任务粒表示为 $\{(t_1, t_2, t_3), (t_4), (t_5)\}$. 若假设每个任务的候选服务数为 n ,则原问题求解规模为 n^5 ,而基于任务粒的求解规模则为 $n^3 + n + n, n^5 \gg (n^3 + n + n)$.

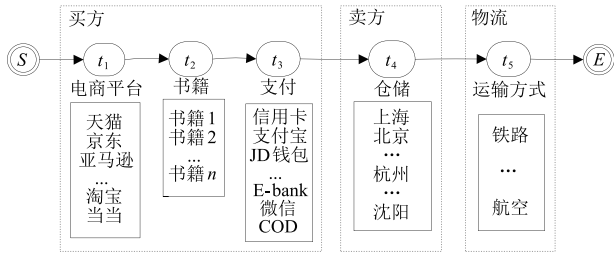


图1 网上购物案例

任务粒化过程将问题的不断细化,问题求解规模越来越小.按照任务粒论域中元素个数进行划分,可得服务组合粒化分层模型如图 2 所示.

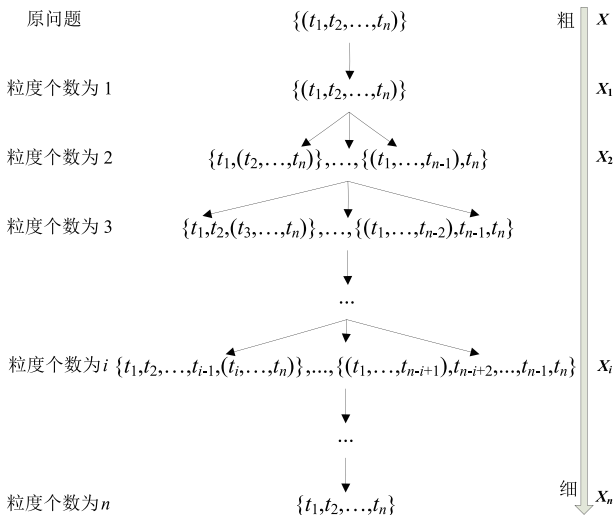


图2 服务组合粒化分层模型

图 2 所示的分层模型中,粒度个数为 i 与 $i + 1$ 层之

间,存在划分关系,表示粒度个数为 $i + 1$ 层的元素组成来自于 i 层的一次划分.例如,粒度个数为 i 时的问题描述为 $X_i = \{t_1, t_2, \dots, t_{i-1}, (t_i, \dots, t_n)\}$,粒度个数为 $i + 1$ 时问题描述为 $X_{i+1} = \{t_1, t_2, \dots, t_i, (t_{i+1}, \dots, t_n)\}$,其中, X_{i+1} 来自于 X_i ,并将其元素粒度更进一步细化,由问题 $\{(t_i, \dots, t_n)\}$ 分解为 $\{t_i, (t_{i+1}, \dots, t_n)\}$ 两个子问题,依此类推.

在实际任务粒化场景中,不同任务候选服务之间会存在耦合关系(或协同关系等).如图 1,当用户选择淘宝电商平台时,一般会选择淘宝物流,而不会选择京东物流.因此,淘宝电商平台和淘宝物流属于强耦合关系,具有较高的隶属度.在实际任务粒化求解过程中,应考虑该因素的影响.

3 可行性分析

3.1 组合服务 QoS 计算方式

不同的 QoS 属性拥有不同的组合计算方式,常用的可以归纳为表 1 所示的 5 类情况^[4-6].

表 1 常用的 QoS 组合计算方式

计算方式	符号	QoS 属性
和	Sum	Cost, Response time
积	Pro	Availability
均值	Avg	Reputation
最大值	Max	Response time(Parallel model)
最小值	Min	Throughput

3.2 QoS 组合计算粒化可行性

根据表 1 所列的组合服务 QoS 计算方式,其任务粒化可行性详细分析如下.

(1) 和: $\text{Sum}(n + m) = \text{Sum}(n) + \text{Sum}(m)$, 其中 $\text{Sum}(n + m)$ 表示 $n + m$ 个数的和,即 $n_1 + n_2 + \dots + n_n + m_1 + m_2 + \dots + m_m = (n_1 + n_2 + \dots + n_n) + (m_1 + m_2 + \dots + m_m)$. 易知,显然成立.

(2) 积: $\text{Pro}(n + m) = \text{Pro}(n) * \text{Pro}(m)$, 其中 $\text{Pro}(n + m)$ 表示 $n + m$ 个数的积,即 $n_1 * n_2 * \dots * n_n * m_1 * m_2 * \dots * m_m = (n_1 * n_2 * \dots * n_n) * (m_1 * m_2 * \dots * m_m)$. 易知,显然成立.

(3) 均值: $\text{Avg}(n + m) = \text{Sum}(n + m) / (n + m)$, 其中 $\text{Sum}(n + m)$ 表示 $n + m$ 个数的和, $\text{Avg}(n + m)$ 表示 $n + m$ 个数的均值,对于同一服务组合优化问题而言,任务数唯一,则 Avg 转换为 Sum 问题, case1 已证.

(4) 最大值: $\text{Max}(n + m) = \text{Max}(\text{Max}(n), \text{Max}(m))$, 其中 $\text{Max}(n + m)$ 表示 $n + m$ 个数中最大值,即 $\text{Max}(n_1, n_2, \dots, n_n, m_1, m_2, \dots, m_m) = \text{Max}(\text{Max}(n_1, n_2, \dots, n_n), \text{Max}(m_1, m_2, \dots, m_m))$, 易知,显然成立.

(5) 最小值: $\text{Min}(n + m) = \text{Min}(\text{Min}(n), \text{Min}(m))$, 类似最大值问题,易知,显然成立.

上述分析表明,组合服务 QoS 计算方式均可以进行粒化计算,并且在任务粒化处理过程中 QoS 运算结果与原问题均保持一致。

3.3 问题粒化计算复杂性分析

本节在文献[7]的基础上对问题粒化过程的计算复杂性进行分析,为方便推导,将图 2 各个层次关系按照从上至下标记为 X, X_1, X_2, \dots, X_n . 因为每个任务的候选服务数是有限集合,则其组合方案也属于有限集合,即 X 是有限集,其个数定义为所有子问题组合方案数之和. 为简化说明,采用 $|X_1|$ 表示 X_1 中元素个数. 随着子问题个数增加,产生论域 X_2 ,若 X_2 包含两个子问题,且分解的子问题规模分别为 n 与 m ,则 $|X_1| = n * m$, $|X_2| = n + m$,由于各任务的候选服务数量大而且非负,因而 $|X_1| > |X_2|$.

虽然,论域 X, X_1, \dots, X_n 是建立在拓扑上的子问题分解,但分解的子问题影响论域元素个数,而服务组合问题求解是建立在服务组合实例搜索之上,即论域 X . 并且,问题的计算量只与 X 的元素个数有关, X 的元素个数越多,表明服务组合方案数越大,搜索空间就越大,最优解的计算复杂性越高. 基于此,计算量函数用 $f(|X|)$ 表示, $f_i(|X|)$ 表示对原问题在第 i 层商空间上的计算量. 为分析 $f(|X|)$ 与 $f_i(|X|)$ 之间的关系,作如下假设.

假设:设问题求解的计算量 $f(\cdot)$ 只与论域元素个数有关,与论域的结构和性质无关. 于是, $f(\cdot)$ 的定义域为 $[0, \infty)$ 半直线,其值域也为非负实数,即:

- (1) $f: \mathbb{R}^+ \rightarrow \mathbb{R}^+$, 是 $\mathbb{R}^+ = [0, \infty)$ 上的单调增函数;
- (2) $x \in \mathbb{R}^+$, 有 $f(x) \leq x$.

则称 f 为 X 上的计算复杂性函数,简称为计算量函数.

假设(1)说明的是随着论域元素个数的增加,计算量呈现出非减趋势;假设(2)说明对于问题论域 X , $f(|X|)$ 的计算量不超过论域 X 元素的个数,即在服务组合问题中,最大的计算量是将每个组合实例都评估一遍.

在对服务组合问题进行分层求解时,在层与层之间都存在信息损失. 例如,图 2 中的 X 与 X_1 ,将原问题分解成两个子问题,则子问题与子问题之间任务层拓扑信息都不存在,只保留了子问题层的拓扑信息,所以,在 X_1 层上每个子任务的求解最优解组合可能不会原问题最优解,则需要获取子问题中 top- k 方案,最优解的命中概率就会显著增加,为描述这一问题,引入目标估计函数 $g(\cdot)$.

定义 2 设 X_1 是 X 的一个商空间, f 为 X 的计算量函数,若对 X_1 进行不超过 $f(|X_1|)$ 的计算量,能估算出在 X_1 中至多有 g 个元素可能是最优解,则定义目标估计函数 $g(\cdot)$ 为:

$$g: \mathbb{R}^+ \rightarrow \mathbb{R}^+, g(|X_1|) = g \quad (1)$$

设 $g(\cdot)$ 是单调增加函数,且 $x \in \mathbb{R}^+, g(x) \leq x$.

其中, $g(x)$ 是单调增函数,即随着服务组合搜索空间的增大,每个子问题最优可行解的组合方案数呈非减趋势,同理, $g(x) \leq x$ 表明每个子问题最优可行解的方案数不超过子问题规模.

在定义 2 的基础上,将可能包含目标的元素看成总体进行分层考虑,可进一步对基于服务组合分层粒化模型的计算量分析如下.

设 $g(x) = bx, 0 < b < 1$,即通过一次分层后,可以判定目标包含在 bx 个元素中,设原问题论域 X 与其商空间 X_1, \dots, X_i ,令 $|X| = x, |X_1| = x_1, |X_2| = x_2, \dots, |X_i| = x_i$. 则 X_1 层的计算量为:

$$f_1(X) = f(x_1) + f(bx_1 * x/x_1) = f(x_1) + f(bx) \quad (2)$$

这里设 X_1 中每个元素包含的 X 中的元素个数相同,即分解后的子问题均分原问题的元素个数.

对 X_1 进行第二次分层,即 X_2 ,于是有:

$$\begin{aligned} f_2(X) &= f(x_1) + f(x_2) + f(bx_2 * bx/x_2) \\ &= f(x_1) + f(x_2) + f(b^2x) \end{aligned} \quad (3)$$

由归纳法可得:

$$f_i(X) = f(x_1) + f(x_2) + \dots + f(x_i) + f(b^i x) \quad (4)$$

当 $b^i x < 1$ 时,表明每个子任务的最优可行解不大于 1 个,则分层停止.

现设 $x = e^n$,取 $t = n$,得 $b \leq 1/e$,即有 $b^n x \leq 1$. 一般地,令 $b^t x = b^t e^n = 1$,得:

$$t \ln b + n = 0 \quad (5)$$

因为 $b < 1$,进一步可得:

$$t = n / |\ln b| \quad (6)$$

当取 $x_i = m$ (一元变量)时,

$$f_i(x) = t f(m) = mn \sim O(mn) \quad (7)$$

当取 $x_i = c$ (常数)时,

$$f_i(x) = t f(c) = c_i n \sim O(n) \quad (8)$$

综上所述,在服务组合问题的粒化分层求解过程中,问题的计算量函数逐步等价于二元乘积型时间复杂度,简化处理后,乃至线性级时间复杂度.

4 算法设计

本文基于任务粒化的服务组合优化算法,记为 TgA (Task-granulating Algorithm). 下面分别对单属性任务粒化验证算法和多属性混合任务粒化验证算法进行阐述.

4.1 单属性任务粒化验证算法

依据表 1 中提供的五种不同的组合计算方式,本部分算法分为两个部分:(1) 获得每个任务粒的最优解;(2) 根据任务粒的最优解获得组合服务最优解. 算法 1 为每个任务粒最优解获取算法.

输入:每个任务粒
 输出:每个任务粒的最优解
 Initialize $min = +\infty, mini = -1$
 Generate an array $best[n]$ to keep the best services
 FOR each $s_i \in s$
 FOR each $s_{i,j} \in s_i$
 IF $min > s_{i,j}$ THEN
 $min = s_{i,j}$
 $mini = j$
 END IF
 END FOR
 Add $s_{i,mini}$ into $best[i]$
 END FOR
 RETURN $best[n]$

其中, s 表示原问题, s_i 表示原问题的第 i 个任务粒, n 表示任务粒数量, min 表示每个任务粒中最优解, $mini$ 表示每个任务粒的最优解位置, $best[n]$ 表示每个任务粒最优解集合。

根据算法 1, 可进一步获取原问题的最优解, 即组合服务最优解, 如算法 2 所示。

算法 2 原问题最优解构造算法

输入:每个任务粒最优解集合 $best[n]$
 输出:原问题最优解
 Initialize $bestfit = 0$
 Get the *type* of composition computing method
 FOR each $s_{i,j} \in best[n]$
 IF *type* = Sum THEN
 Calculate $bestfit = \text{sum}(s_{i,j})$
 END IF
 ...
 IF *type* = Pro THEN
 Calculate $bestfit = \text{pro}(s_{i,j})$
 END IF
 END FOR
 RETURN $bestfit$

其中, $\text{sum}(s_{i,j})$ 表示获得 $s_{i,j}$ 的和, 其余类似。 *type* 表示组合计算方式的类别。通过算法 2, 可得原问题的最优解为 $bestfit$ 。

4.2 多属性混合任务粒化验证算法

多属性服务组合在所有维度上均最优的候选服务不一定存在, 本文取每个任务粒的 top- k 最优解, 将各个任务粒的 top- k 最优解进行枚举, 取其最优作为原问题的解。其详细过程如算法 3 所示。

算法 3 多属性混合任务粒化验证算法

输入:每个任务粒
 输出:原问题最优解

Initialize $bestfit = 0$
 Get the k value of the top- k
 Generate an array $best[n, k]$ to keep the best services
 FOR each $s_i \in s$
 FOR each $s_{i,j} \in s_i$
 Keep the top- k best services $s_{i,j}$
 END FOR
 Add $s_{i,j}$ into $best[i, k]$
 END FOR
 FOR each $s_{i,j} \in best[n, k]$
 Enumerate the $bestfit = \text{Enume}(best[n, k])$
 END FOR
 RETURN $bestfit$

其中, $best[n, k]$ 表示保存每个任务粒中 top- k 最优解, $\text{Enume}(best[n, k])$ 表示获得 $best[n, k]$ 中的枚举最优解。

5 实验分析

本文以数据集 QWS 为测试数据^[8]。验证算法的寻优性能。由于表 1 中 Max 与 Min 类型具有对称性, 可通过数据的简单变化相互转换, 所以, 本文仅测试 Sum、Pro、Avg 以及 Min 四种情形。

5.1 实验环境

实验环境为 VS2010, 编程语言为 C#, 16G、i7、3.6GHZ、Win7。任务数为 5, 每个任务的候选服务数从 100 递变到 500, 每次实验进行 100 次重复实验, 取其均值作为最终衡量指标, 对比算法 PSO 中参数设置如表 2 所示。对于混合多属性计算方式而言, TgA- k 表示每个任务粒中取 top- k 个最优解。

表 2 PSO 算法实验参数表

参数	含义	数值
ω_{\max}	惯性权重最大值	0.9
ω_{\min}	惯性权重最小值	0.1
$C_{1, \max}$	局部学习因子最大值	2
$C_{1, \min}$	局部学习因子最小值	0.5
$C_{2, \max}$	全局学习因子最大值	2
$C_{2, \min}$	全局学习因子最小值	0.5
Sizepop	种群规模	50
Iteration	进化代数	300

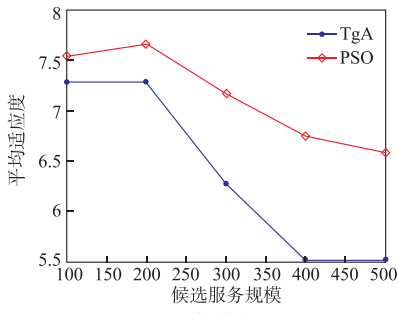
5.2 实验结果与分析

5.2.1 单属性任务粒化验证实验

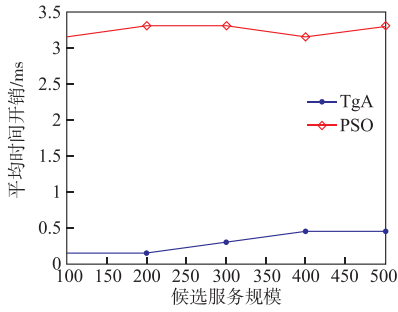
(1) Sum 类型实验结果如图 3 所示。

由图 3(a) 可知, 相比 PSO 算法, TgA 可以获得更优的适应度值, 其寻优效果提升约 10%, 且 TgA 寻优精度不会随候选服务数量的增加而变化。图 3(b) 表明相比于 PSO 算法平均大于 3 秒的寻优时间, TgA 的时间性能提升 6 倍。图 3(c) 可以看出, PSO 算法均方差随着候选服务数量的增加而呈现线性增加, TgA 均趋于 0。

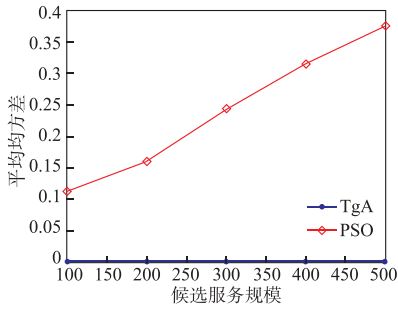
(2) Pro 类型: 为表明 Max 类型与 Min 类型的一致



(a) 寻优效果



(b) 时间开销



(c) 稳定性

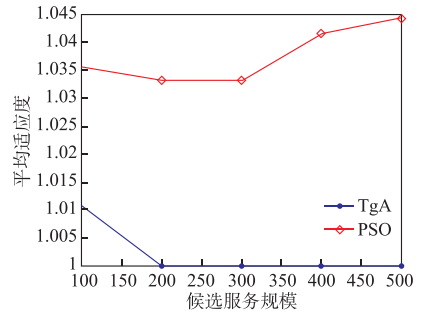
图3 Sum类型寻优性能对比实验

性,对数据采用归一化处理,而归一化所得结果满足 $[0,1]$,最小值为0,若采用乘积 Pro 进行组合计算,无法获得最优组合,所以本文针对 Pro 类型实验,均将 QoS 属性数据归一到 $[1,2]$ 之间.实验结果如图 4 所示.

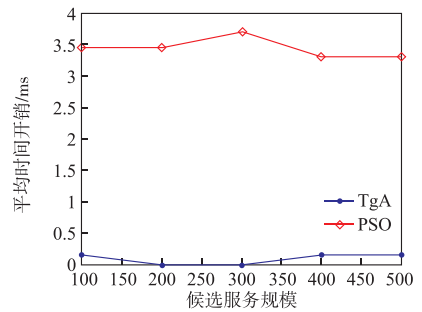
从图 4(a)可以看出,随着候选服务数量的增加,PSO 的平均适应度值呈现略微上升趋势,这是由于候选服务数量的增加,PSO 算法的粒子变化域也呈现指数增长趋势,实验精度有所降低.而 TgA 能维持稳定不增趋势,且实验结果能达到全局最优解.从图 4(b)可知,TgA 的时间性能约为 PSO 的 1/7,表明 TgA 可利用更少的时间获得更优的实验效果,即本文算法具有很强的全局寻优能力和执行效率.

(3) Avg 类型实验结果如图 5 所示.

由图 5(a)可知,相比于 PSO 而言,TgA 寻优效果提高约为 10%,并且随着候选服务数量的增加,其差距越大.图 5(b)表明 TgA 的算法效率约为 PSO 的 6 倍.同样表明 TgA 是一种全局寻优能力更强、效率更高的组合寻优算法.

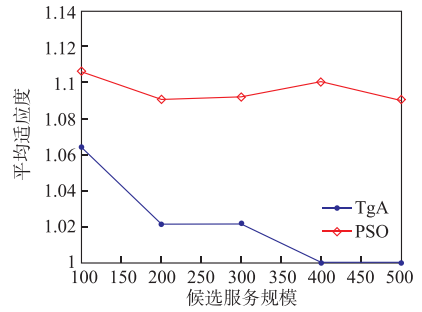


(a) 寻优效果

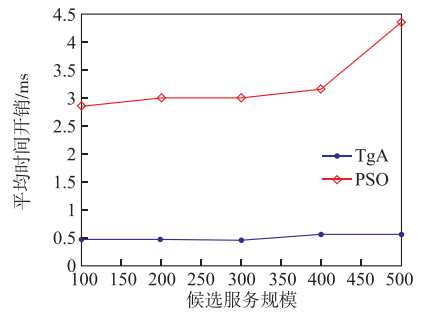


(b) 时间开销

图4 Pro类型寻优性能对比实验



(a) 寻优效果



(b) 时间开销

图5 Avg类型寻优性能对比实验

(4) Min 类型:由于 TgA 与 PSO 实验结果相差较小,采用数据表形式展示实验结果如表 3 所示.

由表 3 可知,在 Min 类型中,TgA 可以获得与 PSO 同样的寻优结果,但从时间开销方面,TgA 所用时间均小于 0.5ms,而 PSO 的时间开销约为 3s,即 TgA 的时间性能约是 PSO 的 6 倍.从 RMSE 来看,TgA 与 PSO 具有相同的稳定性,均趋于 0.

表 3 Min 类型实验结果

指标	算法	100	200	300	400	500
Fitness	TgA	1.0011	1.0006	1.0006	1	1
	PSO	1.0011	1.0006	1.0006	1	1
Time	TgA	0.15	0.45	0.45	0.45	0.45
	PSO	4.05	3.9	3	3.15	3.45
RMSE	TgA	1.11E-15	1.55E-15	1.55E-15	0	0
	PSO	1.11E-15	1.55E-15	1.55E-15	0	0

5.2.2 多属性混合验证实验

在更多情况下, QoS 计算方式以混合多属性形式出现, 如 Cost 属性往往采用 Sum 计算方式, 而可用性 Availability 则采用 Pro 计算方式, 若同时出现 Cost 与 Availability 属性, 则需要对 Sum 与 Pro 计算方式进行组合. 实验结果如图 6 所示.

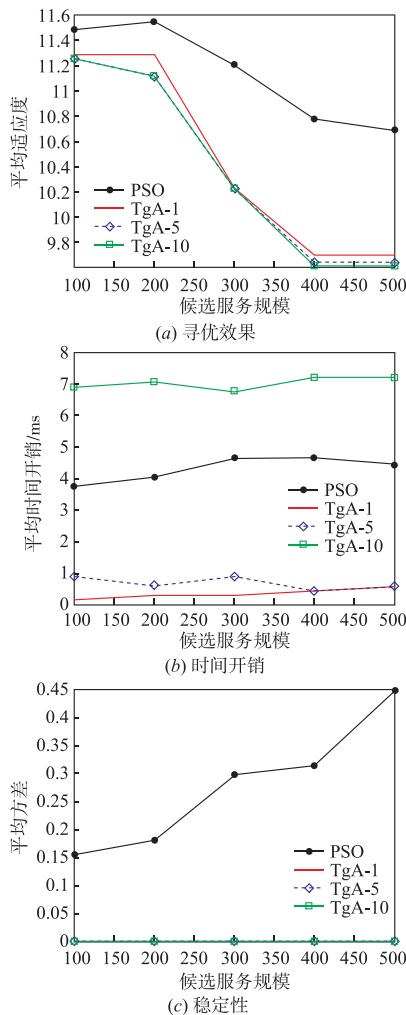


图6 多属性混合验证实验

图 6(a) 可知, 相比于 PSO 算法, TgA 可以获得更优的寻优结果, 尤其随着 TgA-k 中 k 值的增加, TgA 算法的实验结果越优. 总体上, TgA 算法的平均适应度相比

PSO 而言均提高约 10%. 图 6(b) 可看出, TgA-1 与 TgA-5 的时间开销远小于 PSO, 相比于 PSO 的 4ms, 时间性能提高约 4 倍, 而 TgA-10 的时间开销大于 PSO, 这是由于每个子问题取的解越多, 其组合方式就越多, 计算时间越长. 从图 6(c) 可知, 随着候选服务数量的增加, PSO 的均方差呈上升趋势, 表明其稳定性较差, 而本文算法在不同的 top-k 上, 均方差同样趋于 0.

6 总结

本文基于粒计算理论, 提出服务组合问题粒化分层模型, 根据不同的应用场景将原问题分解到不同的粒层, 进而将问题的计算复杂性从指数级降为任务线性级, 从计算方式上改变服务组合问题的求解过程, 易于实现. 并从理论与实验两方面验证本文所提模型与算法的可行性.

参考文献

- [1] He Q, Han J, Chen, et al. QoS-aware service selection for customisable multi-tenant service-based systems; maturity and approaches [A]. Proceedings of 2015 IEEE 8th International Conference on Cloud Computing [C]. New York, USA; IEEE, 2015. 237 - 244.
- [2] Zhang Y, Cui G, et al. IFOA4WSC: a quick and effective algorithm for QoS-aware service composition [J]. International Journal of Web and Grid Services, 2016, 12(1): 81 - 108.
- [3] 尹浩, 张长胜, 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 Chang-sheng, 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)
- [4] 黄龙涛, 邓水光, 等. 基于 MapReduce 的并行 Web 服务自动组合 [J]. 电子学报, 2012, 40(7): 1397 - 1403.
HUANG Long-tao, DENG Shui-guang, et al. Automatic service composition in parallel with MapReduce [J]. Acta Electronica Sinica, 2012, 40(7): 1397 - 1403. (in Chinese)
- [5] Zhang Y, Cui G, Wang Y, et al. An optimization algorithm for service composition based on an improved FOA [J]. Tsinghua Science & Technology, 2015, 20(1): 90 - 99.
- [6] Jatoth C, Gangadharan G, Buyya R. Computational intelligence based qos-aware web service composition; a systematic literature review [J]. IEEE Transactions on Services Computing, 2015, DOI 10.1109/TSC.2015.2473840.
- [7] Zhang L, Zhang B. The quotient space theory of problem solving [J]. Fundamenta Informaticae, 2004, 59(2-3): 287 - 298.

[8] Al-Masri E, Mahmoud Q H. Discovering the best web service[A]. Proceedings of the 16th international conference on

World Wide Web [C]. Banff, Alberta, Canada: ACM, 2007. 1257 – 1258.

作者简介



张以文 男,1976 年出生. 安徽大学计算机科学与技术学院副教授, 博士, 硕士生导师. 研究方向为服务计算、大数据.

E-mail: zhangyiwen@ahu.edu.cn



崔光明(通信作者) 男,1991 年出生. 安徽大学计算机科学与技术学院硕士生. 研究方向为服务计算、机器学习.

E-mail: cgm133123@163.com



郭星 男,1983 年出生. 安徽大学计算机科学与技术学院讲师, 博士. 研究方向: 机器学习、服务计算、大数据.

E-mail: guoxing@ahu.edu.cn



张燕平 女,1962 年出生. 安徽大学计算机科学与技术学院教授, 博士生导师. 研究方向为商空间、智能计算.

E-mail: zhangyp2@gmail.com