

基于局部近似过滤的多约束服务选择方法

梁中军^{*①} 邹华^① 郭静^② 杨放春^① 林荣恒^①

^①(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

^②(北京邮电大学计算机学院 北京 100876)

摘要: 随着 Web 服务数量与用户需求的不断增长, 如何在功能约束和 QoS 约束下选择 Web 服务去构建高质量的组合服务已成为 Web 服务领域的一项重要研究内容。然而, 现有大多数 Web 服务选择方法仅仅解决了在多种 QoS 约束下选择服务的问题, 忽略了服务间的功能性约束。为解决这一问题, 该文提出一种基于局部近似过滤的多约束服务选择方法。该方法首先利用候选服务在多种约束下的被选关系, 滤除部分不满足约束的服务, 并估算每个剩余候选服务的局部适应度; 然后, 利用文中设计的有向粒子群算法, 以候选服务的局部适应度为引导信息搜索出最优方案。实验结果表明了方法的有效性。

关键词: 服务质量; 多约束服务选择; 粒子群; 局部近似过滤

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2013)10-2314-07

DOI: 10.3724/SP.J.1146.2013.00545

Multi-constraint Service Selection Based on Local Approximate Filter

Liang Zhong-jun^① Zou Hua^① Guo Jing^② Yang Fang-chun^① Lin Rong-heng^①

^①(State Key Laboratory of Networking and Switching Technology, Beijing University of

Posts and Telecommunications, Beijing 100876, China)

^②(School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Web service selection is a critical procedure for performance-enhancing in composite service. To find the best services from the candidate services, both the Quality of Service (QoS) requirements and functional requirements should be considered. However, most Web service selection methods are based on the assumption of the independence among candidate services, and ignore the dependency relationship and compatible relationship among candidate services. In practice, composite services emphasize the coordination among these component services. The function of a candidate service in a composite service usually depends on the other optional service. To solve this problem, a multi-constraint service selection method is proposed based on local approximate filter. This method filters out part of the unsatisfied constrain services based on local approximate filter, and estimates the local fitness of each of the rest candidate services, then defines a suitable particle swarm algorithm to search the optimal solutions in the light of the calculated local fitness. Experimental results demonstrate the effectiveness of this method.

Key words: Quality of Service (QoS); Multi-constraint service selection; Particle swarm; Local approximate filter

1 引言

随着 Web 服务数量与用户需求的不断增长, 如何选择 Web 服务构建高质量的组合服务已成为 Web 服务领域的一项重要研究内容^[1]。在实际的服务选择中, 被选服务需要满足多种约束, 不仅包括用户对组合服务的整体服务质量(QoS)约束, 还包括由于业务关联、技术兼容、地域性等多种因素所引

起的依赖、冲突约束(功能性约束)。

目前, 已有大量在 QoS 约束下选择 Web 服务的方法^[2-11]被提出。其中, 文献[3]考虑在多种 QoS 约束下, 以寻找全局最优的候选服务为目标, 采用混合整数规划为手段为用户选择满意的候选服务。文献[4]在加入了局部约束条件的基础上, 采用线性规划的方法对问题进行求解。在应用实时性要求的驱动下, 文献[7]同样在多种 QoS 约束下, 利用遗传算法对问题进行求解。文献[8]结合全局约束与局部约束提出了一种基于全局约束分解的服务选择算法。然而, 上述研究仅关注于满足组合服务的整体

2013-04-22 收到, 2013-07-27 改回

国家 973 规划项目(2009CB320406)和国家 863 计划项目(2011AA01A102)资助课题

*通信作者: 梁中军 kkliangjun@gmail.com

质量要求，忽略了服务间的功能性约束，无法直接应用于具有多种约束的现实场景中。目前，也有少量研究^[12-17]涉及了多约束下的服务选择问题。如文献[13]利用整数规划算法对服务之间存有冲突约束关系的问题进行求解；文献[15-17]针对服务间存在依赖和冲突的情况，提出3种遗传算法对问题进行求解。然而，上述方法均没有充分利用候选服务在约束下的被选关系，难以在满足实时性要求的前提下选择高质量的组合方案。

为此，本文提出一种基于局部近似过滤的多约束服务选择方法。该方法首先利用候选服务在多种约束下的被选关系，滤除部分不满足约束的服务，并估算每个剩余候选服务的局部适应度；然后，利用本文设计的有向粒子群算法，以候选服务的局部适应度为引导信息对最优方案进行搜索。

本文的主要贡献为：(1)提出一种支持多种约束的服务选择框架。它利用候选服务在约束下的被选关系，由过滤和搜索两步完成服务选择。实验结果表明本框架能在满足实时性要求的前提下，有效提高选择方案的质量。(2)设计一种局部过滤处理算法。它以候选服务在约束下的被选关系为基础，采用近似策略快速滤除部分不满足约束要求的候选服务，并估计剩余候选服务的局部适应度，从而减少服务选择的搜索空间(提高可行方案的比例)，并为后续的搜索提供先验信息。(3)设计一种有向粒子群搜索算法。它根据多约束服务选择问题的特点，重新定义了粒子的更新算子，提出动态调整参数方法，设计新的适应度函数指导算法搜索最优方案，并根据候选服务的局部适应度，设计局部优先变异策略来提高算法对最优候选服务的探索能力。

2 问题形式化定义及预知识

本文将多约束服务选择问题看作一个满足多种约束条件的单目标优化问题，其表达式为

$$U(\mathbf{CS}) = \max \sum_{k=1}^r w_k \times q_k'(\mathbf{CS}) \quad (1)$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{|S_j|} x_j^i = 1, & 1 \leq j \leq n \\ x_a^e - x_b^c \leq 0, & a, b, c, e \text{ 满足 } \langle s_{ae}, s_{bc}, \subseteq \rangle \\ x_h^o + x_l^p \leq 1, & h, l, o, p \text{ 满足 } \langle s_{ho}, s_{lp}, \perp \rangle \\ q_k(\mathbf{CS}) \leq c_t, & c_t \in \mathbf{QC} \end{cases}$$

式(1)及其约束中参量定义说明如下：

(1) 设 $\mathbf{S} = \{S_1, \dots, S_j, \dots, S_n\}$ 是根据组合服务流程发现的 n 类候选服务，每个服务 s 拥有多个QoS属性(如：响应时间、价格、可靠性等)，其取值由 $\mathbf{Qs} = \{q_1(s), \dots, q_r(s)\}$ 表示。由于多种复杂因素的存

在，被选择的服务需要满足多种约束，包括整体QoS约束和候选服务之间的依赖冲突关系。其中前者由 m 个QoS约束向量 $\mathbf{QC} = \{c_1, c_2, \dots, c_m\}$ 来表示，其取值反映了QoS属性的上限或下限；后者分别由 $\mathbf{D} = \{d_1, d_2, \dots, d_k\}$ 和 $\mathbf{CL} = \{cl_1, cl_2, \dots, cl_u\}$ 来表示。其中，依赖关系 $d_i = \langle s_{ae}, s_{bc}, \subseteq \rangle$ 表示服务类 S_a 中的第 e 个服务 s_{ae} 的功能实现依赖于服务类 S_b 中第 c 个服务 s_{bc} ，冲突关系 $cl_i = \langle s_{ho}, s_{lp}, \perp \rangle$ 表示服务类 S_h 中的第 o 个服务 s_{ho} 不能与服务类 S_l 中第 p 个服务 s_{lp} 同时使用。

(2) w_k 为权重因子，反映了用户对第 k 个QoS属性的偏好程度； \mathbf{CS} 为本问题的选择方案，由每类服务的选择情况共同确定，其中 x_j^i 反映了服务类 S_j 中第 i 个服务被选择的情况， $x_j^i = 1$ 表示该服务被选择；公式 $\sum_{i=1}^{|S_j|} x_j^i = 1$ 保证了每个服务类中有一个候选服务被选择； $q_k(\mathbf{CS})$ 为 \mathbf{CS} 在第 k 个QoS属性上的取值， $q_k(\mathbf{CS}) \leq c_t$ 保证了选择方案能够满足用户的整体质量要求， $q_k'(\mathbf{CS})$ 为 $q_k(\mathbf{CS})$ 归一化后的取值； $x_a^e - x_b^c \leq 0$ 和 $x_h^o + x_l^p \leq 1$ 保证了被选成员服务之间的相容性。

3 基于局部近似过滤的多约束服务选择方法

3.1 候选服务的被选关系与方法的整体步骤

在多种约束的制约下候选服务的被选关系表现在以下3个方面：(1)整体QoS约束使得候选服务的可行性依赖于其它候选类中服务的质量分布。(2)依赖约束使得候选服务的可行性由被依赖的服务决定。(3)在同一候选类中，每个服务被选择的概率都是由服务的质量与其它服务的相容性所共同决定的，二者的取值越好，则该候选服务被选择的概率越高。

利用上述被选关系，本文提出一种基于局部近似过滤的多约束服务选择方法(简称为LDPSO)。方法首先过滤部分不满足约束要求的候选服务，并估计剩余服务的局部适应度；然后，以上述信息为基础，利用有向粒子群算法对候选服务进行选择。图1为该方法的整体框架图，具体实现步骤如下：

步骤1 局部近似过滤。该步骤为整个服务选择方法的预处理步骤，利用整体QoS约束信息和服务间的依赖关系对候选服务进行过滤，并计算每个剩余候选服务的局部适应度。其目的是提高可行方案的比例，减少服务的搜索空间，并为后续的服务选择提供先验信息。具体实现分为以下3步：

(1) 为每个服务 s_i 创建相应的依赖服务集合 $d(s_i)$ 和冲突服务集合 $cl(s_i)$ ；即利用依赖关系的传递特性，将间接依赖和间接冲突的服务分别纳入服务的依赖集合和冲突集合。

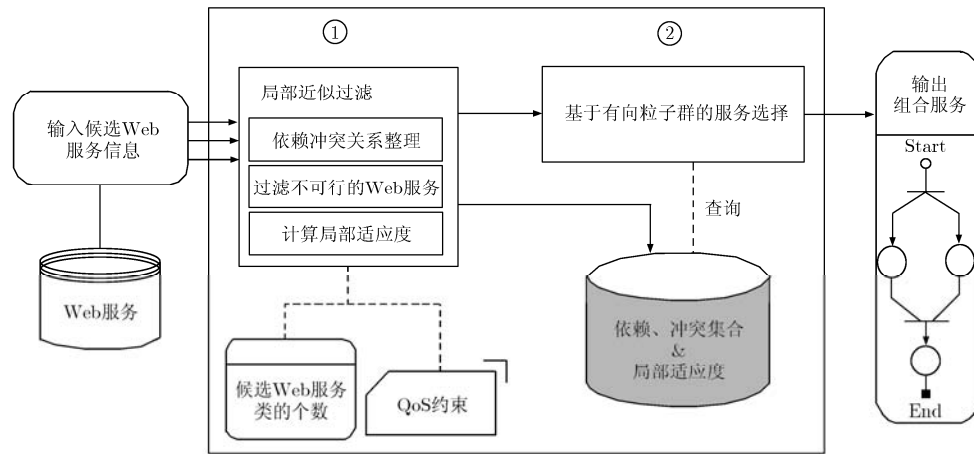


图1 基于局部近似过滤的多约束服务选择方法的整体框架

(2) 利用整体QoS约束信息和服务间的依赖关系, 过滤不满足QoS要求的服务和依赖于它们的服务。首先, 计算每类候选服务的局部质量过滤边界 $B(S_j) = \{b_1^j, \dots, b_m^j\}$, 并使用边界对该类候选服务进行过滤; 然后, 检查剩余候选服务的依赖集合, 将依赖于被过滤服务的候选服务去除, 更新候选服务的依赖集合与冲突集合。为保证选择方法的实时性, 算法采用近似过滤策略计算每类候选服务的质量过滤边界。不同的QoS聚合类型对应的计算方法为

$$b_i^j = \begin{cases} c_i - q_i^{\text{best}}(\mathbf{CS}) + q_i^{\text{best}}(S_j), & \text{QoS聚合类型为加法} \\ c_i \times q_i^{\text{best}}(S_j) / q_i^{\text{best}}(\mathbf{CS}), & \text{QoS聚合类型为乘法} \\ c_i, & \text{QoS聚合类型为最小值} \end{cases} \quad (2)$$

其中 c_i 为用户对组合服务在第 i 个QoS属性上的质量要求, $q_i^{\text{best}}(S_j)$ 为候选服务类 S_j 在第 i 个QoS属性上所能提供的最好质量, $q_i^{\text{best}}(\mathbf{CS})$ 为组合服务在第 i 个QoS属性上所能提供的最好质量, 该取值由每个候选服务类中的最好取值聚合计算获得。

(3) 估计剩余候选服务的局部适应度, 为搜索阶段服务; 即以服务的相容性与服务的质量来综合估计服务在候选服务类的综合情况。

定义1(局部相容度) 局部相容度 $\text{Con}(s_j)$ 反映了服务 s_j 与其它类服务的整体相容情况。假设服务 s_j 属于候选服务类 S_j , 则该服务的局部相容度 $\text{Con}(s_j)$ 可由式(3)计算得到。

$$\text{Con}(s_j) = \prod_{c=1, c \neq j}^n \frac{\text{com}_c(s_j)}{|S_j|} \quad (3)$$

$$\text{com}_c(s_j) = \begin{cases} d_c(s_j), & d_c(s_j) > 0 \\ |S_c| - cl_c(s_j), & \text{其它} \end{cases} \quad (4)$$

其中 $\text{com}_c(s_j)$ 为服务 s_j 与服务类 S_c 中相容服务的个数, 其数量可由服务的依赖集合或冲突集合计算得

到。 $d_c(s_j)$ 为服务依赖集合 $d(s_j)$ 中属于服务类 S_c 的服务个数, $cl_c(s_j)$ 为服务依赖集合 $cl(s_j)$ 中属于服务类 S_c 的服务个数, $|S_c|$ 为候选服务类 S_c 的服务总数。

定义2(局部适应度) 局部适应度 $cq(s_j)$ 是服务 s_j 在该候选服务类中的一个综合评价。假设服务 s_j 属于候选服务类 S_j , 则该服务的局部适应度 $cq(s_j)$ 可由式(5)计算得到。

$$cq(s_j) = \text{Con}(s_j) \times \left(\sum_{k=1}^r w_k \times \frac{q_{\max}^k(S_j) - q_k(s_j)}{q_{\max}^k(S_j) - q_{\min}^k(S_j)} + \sum_{k=r+1}^m w_k \times \frac{q_k(s_j) - q_{\min}^k(S_j)}{q_{\max}^k(S_j) - q_{\min}^k(S_j)} \right) \quad (5)$$

其中 $q_{\max}^k(S_j)$ 和 $q_{\min}^k(S_j)$ 分别为候选服务类 S_j 在第 k 个QoS属性上的最大值和最小值, w_k 为用户对第 k 个QoS属性的偏好权重。

步骤2 基于有向粒子群的服务选择。 该步骤就是利用候选服务的局部适应度, 采用粒子群算法的思想在满足多种约束的条件下对高质量的组合方案进行搜索, 具体将在3.2节介绍。

3.2 基于有向粒子群的服务选择

为适应问题求解, 本文设计了一种有向粒子群算法来对最优方案进行搜索。根据多约束服务选择问题的特点, 本文对传统粒子群算法的执行操作进行了改进, 主要包括问题的粒子编码, 适应度函数建立, 粒子群更新算子的重新定义以及算子参数设置等。

3.2.1 问题的粒子编码 问题的合理表示是粒子群算法快速求解的基础, 根据本问题具有的组合特点, 本文以十进制整数对每个候选服务进行编码, 并将1个服务组合方案表示成一个粒子。具体如图2所示。

图2中, 每个粒子由 n 个位置信息 (x_1, x_2, \dots, x_n)

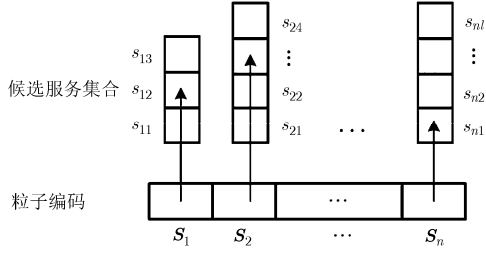


图2 服务选择问题的粒子编码

构成， n 是候选服务类的个数，粒子的每个位置 $x_i, i \in [1, n]$ 代表了服务类 S_i 中的一个具体服务。

3.2.2 适应度函数建立 适应度函数是评价问题解优劣的综合指标，用于指导粒子向当前最优方案进行迭代学习。为了更好地支持粒子对最优组合方案进行搜索，本文以乘法为算子采用式(6)评估所有方案的优劣。

$$F(X) = \begin{cases} 0.5 + 0.5 \times U(X), & VI(X) = 0 \\ 0.5 \times U(X) \times \frac{VI_{\max} - VI(X)}{VI_{\max}}, & VI(X) > 0 \end{cases} \quad (6)$$

其中 $U(X)$ 为服务组合方案 X 的效用函数， VI_{\max} 为所有方案违背约束个数的上限， $VI(X)$ 为方案 X 违背约束的个数。

3.2.3 粒子群更新算子的重新定义 为保证粒子群算法在候选服务类中充分搜索，本文为粒子赋予明确的搜索方向，并在候选服务的编码空间中对粒子的更新方式进行重新定义。

定义3(速度 V) 表示粒子在当前选择方案 X_j 上的搜索方向，其取值可由减法操作获得。由于选择方案来自 n 个候选服务类，因而被表示成 n 维向量 $V = \{v_1, v_2, \dots, v_n\}$ ，每维的取值均满足 $v_k \in \{1, 0, -1\}$ ， $k \in [1, n]$ 。当 $v_k = 1$ (或 $v_k = -1$) 时，粒子将在候选服务类 S_k 的当前选择服务的上方(或下方)进行搜索，而 $v_k = 0$ 时，粒子将保持当前选择。

定义4(减法 \ominus) 表示在候选服务编码空间中选择方案 X_i 与选择方案 X_j 在各维度上的位置关系。如果 x_i^k, x_j^k 分别为 X_i 与 X_j 在第 k 维的编码取值， v_k 为 x_i^k 和 x_j^k 在候选组合服务编码空间的位置关系，则 v_k 的取值由 $\{1, 0, -1\}$ 构成，分别代表 x_i^k 在 x_j^k 的下方， x_i^k 与 x_j^k 位置相同和 x_i^k 在 x_j^k 的上方。方案的减法操作满足式(7)：

$$X_i \ominus X_j = \{v_1, v_2, \dots, v_n\} \quad (7)$$

定义5(系数) 表示粒子速度(即：搜索方向)在更新中保持当前速度惯性，向自身最优位置方向和全局最优位置方向运动的概率，取值分别由 ω, c_1, c_2 来表示，其中 $\omega + c_1 + c_2 = 1$ 。

定义6(乘法 \otimes) 表示粒子受系数影响改变速度的操作。由于粒子受当前速度惯性，自身最优位置和全局最优位置的影响，其改变速度的情况可由 $\omega V^w \otimes c_1 V^{c_1} \otimes c_2 V^{c_2}$ 来表示。其中 ωV^w 表示粒子以 ω 的概率按照惯性方向搜索， $c_1 V^{c_1}$ 和 $c_2 V^{c_2}$ 分别表示粒子以 c_1 的概率向自身最优位置方向搜索，以 c_2 的概率向全局最优位置方向搜索。

定义7(加法 \oplus) 表示粒子在当前位置 X_i 上按照速度 V_i 进行随机搜索的操作，通过该步骤将获得新的选择方案。其操作满足式(8)和式(9)：

$$x_i^k \oplus v_{ik} = x_i^k + r_i \times v_{ik} \quad (8)$$

$$r_i = \begin{cases} e, & v_{ik} = 1 \text{ 且 } e \in [1, |S_k| - x_i^k] \\ 0, & v_{ik} = 0 \\ z, & v_{ik} = -1 \text{ 且 } z \in [1, x_i^k - 1] \end{cases} \quad (9)$$

其中 x_i^k 为 X_i 在第 k 维的位置取值， v_{ik} 为 V_i 在第 k 维的速度。 r_i 为随机选取的整数，其取值范围与 v_{ik} 相关，反映了粒子的搜索范围。

定义8(越界阻止算子 \rightarrow) 表示粒子在当前位置 X_i 上由更新速度 V_i 获得惯性速度 V_i^w 的操作，它通过判定 X_i 是否在候选服务编码空间的边界来改变惯性速度的方向，目的是防止粒子搜索区域超出候选服务的编码空间。其操作满足式(10)和式(11)：

$$x_i^k \rightarrow v_{ik} = v_{ik}^w \quad (10)$$

$$v_{ik}^w = \begin{cases} v_{ik}, & x_i^k \neq 1 \text{ 或 } x_i^k \neq |S_k| \\ v_{\text{new}} \in \{1, 0, -1\} - v_{ik}, & \text{其它} \end{cases} \quad (11)$$

其中 x_i^k 为 X_i 在第 k 维更新后的位置取值， v_{ik} 为 V_i 在第 k 维的速度取值， v_{new} 为速度取值集合中除去 v_{ik} 的随机取值， v_{ik}^w 为 V_i^w 在第 k 维的取值。

在上述定义的基础上，有向粒子群算法的更新过程可表示为

$$V_{t+1} = \omega V_t^w \otimes c_1(X_{pb} \ominus X_t) \otimes c_2(X_{gb} \ominus X_t) \quad (12)$$

$$X_{t+1} = X_{t+1} \oplus V_{t+1} \quad (13)$$

$$V_{t+1}^w = X_{t+1} \rightarrow V_{t+1} \quad (14)$$

3.2.4 速度系数的计算 速度决定了粒子搜索的方向和快慢(太快可能错过最优解区域，太慢则影响求解效率)。为了平衡二者的关系，本文设计了一种动态的系数调整算法，其系数计算公式为

$$\omega = \frac{F^a(X_t)}{F^a(X_t) + F^a(X_{pb}) + F^a(X_{gb})} \quad (15)$$

$$c_1 = \frac{F^a(X_{pb})}{F^a(X_t) + F^a(X_{pb}) + F^a(X_{gb})} \quad (16)$$

$$c_2 = \frac{F^a(X_{gb})}{F^a(X_t) + F^a(X_{pb}) + F^a(X_{gb})} \quad (17)$$

$$a = a_{\min} + (a_{\max} - a_{\min}) \times \frac{t}{T} \quad (18)$$

其中 $F(\mathbf{X}_t)$ 和 $F(\mathbf{X}_{pb})$ 分别为粒子在当前位置和自身最优位置的适应度, $F(\mathbf{X}_{gb})$ 为全局最优位置的适应度; a 为选择压力参数, a_{\max} 和 a_{\min} 分别是该压力参数的最大值和最小值, 其取值满足 $a_{\max} > 1$ 和 $a_{\min} < 1$ 。

3.2.5 局部优先变异策略 为保证有向粒子群算法能够搜索到最优的候选服务, 本文对每次迭代的全局最优粒子实行变异操作。具体为随机选择当前全局最优方案的某个候选服务类 S_j , 以该类服务的局部适应度为基础, 采用概率选择的方式进行服务替换, 其中每个候选服务的被选概率将用式(19)计算。

$$p(x_i) = \frac{cq(x_i)}{\sum_{k=1}^{|S_j|} cq(x_k)} \quad (19)$$

其中 $p(x_i)$ 为服务 x_i 被选择的概率, $|S_j|$ 为服务类 S_j 中候选服务的个数。

3.3 时间复杂度分析

假设有 n 类候选服务集合, 每类集合包含 w 个服务, 每个服务有 r 个 QoS 属性, 候选服务间的依赖冲突关系的总数分别为 k 和 u , 则局部近似过滤步骤的时间复杂度为 $O(n \times w \times r + k + u)$; 假设种群数为 N , 最大迭代次数为 T , 则利用有向粒子群算法进行服务选择的时间复杂度为 $O(N \times T \times n \times w \times r)$; 因此, 本方法的时间复杂度最多为 $O(N \times T \times n \times w \times r + k + u)$ 。

4 实验与分析

为验证LDPSO的合理性和有效性, 实验对LDPSO与文献[15]的Repair Genetic Algorithm (RGA), 文献[16]的Penalty-based Genetic Algorithm (PGA)和文献[17]的Hybrid Genetic Algorithm (HGA)算法进行比较, 并以方法的执行时间和选择方案的质量(适应度)为评价指标, 详细分析方法在不同候选服务数量, 不同任务规模和不同约束规模下的选择性能。为获得真实可靠的分析结果, 实验

以真实的数据集QWS^[8]为选择对象, 采用多次实验的平均结果来分析方法的性能。

实验 1 不同候选服务数量下的选择性能

本次实验将候选服务类的个数 n 和候选服务间的约束关系 C_0 设置为常量, 通过改变候选服务的数量 w 来分析方法的执行结果, 其中 $n = 5$, $C_0 = 300$, w 从40变化至400。

图3和图4展示了4种方法在不同候选服务数量下的执行时间和被选方案的适应度。从图中可以看出, LDPSO不仅在执行时间和选择方案的适应度上优于其它3种算法, 而且执行时间并不随服务数量的增加而剧烈增长。LDPSO选择方案的适应度相比于HGA选择方案的适应度平均提高了7.9%, 相比于RGA选择方案的适应度提高了7.3%, 相比于PGA选择方案的适应度提高了7%。

实验 2 不同任务规模下的选择性能

本次实验将每类候选服务的数量 w 和候选服务间的约束关系 C_0 设置为常量, 通过改变候选服务类的个数 n 来分析4种方法的执行结果, 其中 $w = 80$, $C_0 = 300$, n 从4变化至28。

图5和图6展示了4种方法在不同任务规模下的执行时间和被选方案的整体质量。从图中可以看出, LDPSO不仅能在不同的任务规模下选择质量较好的方案, 而且执行时间并不随任务数量的增加而剧烈增长。LDPSO选择方案的适应度相比于HGA选择方案的适应度平均提高了5.3%, 相比于RGA和PGA选择方案的适应度提高了5.2%。

实验 3 不同约束规模下的选择性能

本次实验将每类候选服务的数量 w 和候选服务类的个数 n 设为常量, 通过改变候选服务间的约束关系 C_0 来分析4种方法的执行结果, 其中 $n = 8$, $w = 80$, C_0 从200变化至2000。

图7和图8展示了4种方法在不同约束规模下的执行时间和选择方案的整体质量。从图中可以看出, LDPSO不仅在执行时间和选择方案的质量上优于其它3种算法, 而且执行时间并不随约束数量的增大

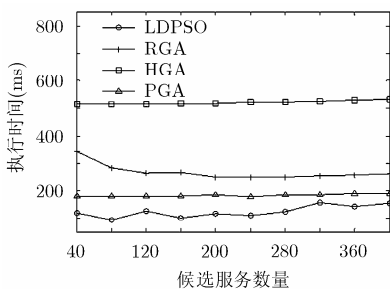


图3 执行时间随候选服务增长的变化

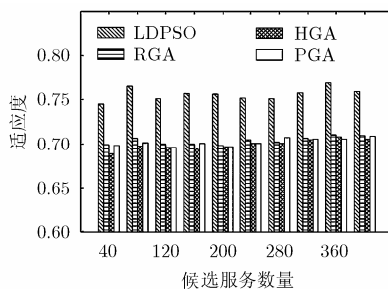


图4 不同候选服务数量下取得的适应度

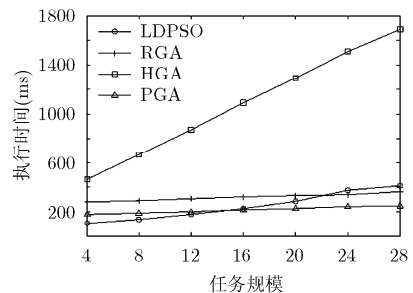


图5 执行时间随任务规模增长的变化

而明显增长。LDPSO选择方案的适应度相比于HGA选择方案的适应度平均提高了5.9%，相比于RGA选择方案的适应度提高了5.7%，相比于PGA选择方案的适应度提高了5.6%。

实验 4 局部近似过滤步骤的作用和影响

为分析局部近似过滤步骤对提高被选方案质量的作用和对选择效率的影响，实验将在不同的候选服务数量 w ，不同候选服务类的个数 n 和不同的约束关系 C_0 下进行，具体取值如下：选择实例1: $n = 10, w = 100, C_0 = 100$ ；选择实例2: $n = 10, w = 200, C_0 = 100$ ；选择实例3: $n = 5, w = 200, C_0 = 100$ ；选择实例4: $n = 5, w = 200, C_0 = 1000$ 。其中，DPSO表示去除局部近似过滤步骤的本文方法。

图9和图10展示了本文方法使用局部近似过滤

步骤和不使用该步骤时，在不同选择实例下的执行时间与选择方案的适应度。从图中可以看出，使用局部近似过滤步骤能够在较短的时间内有效提高被选方案的适应度。另外，由于该步骤计算复杂度较低，因此不会对选择的效率造成较大影响。

5 结论

本文针对多约束下的服务选择问题，提出一种基于局部近似过滤的多约束服务选择方法。该方法首先利用候选服务在多种约束下的被选关系，滤除部分不满足约束的服务，并估算每个剩余候选服务的局部适应度；然后，利用本文设计的有向粒子群算法，以候选服务的局部适应度为引导搜索出最优方案。实验结果表明本方法能在满足实时性要求的前提下，为用户提供更好的服务选择方案。

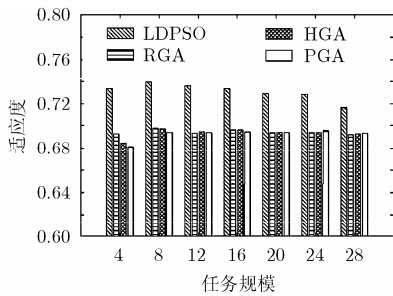


图 6 不同任务规模下取得的适应度

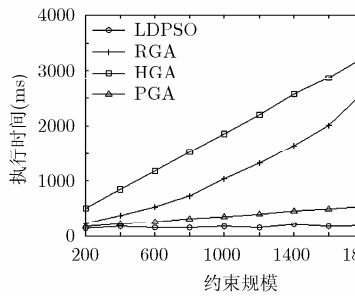


图 7 执行时间随约束规模增长的变化

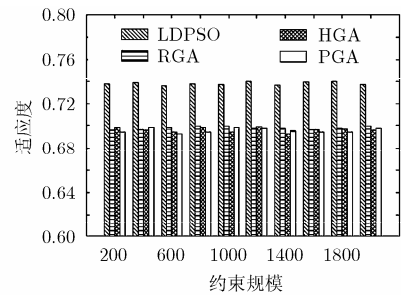


图 8 不同约束规模下取得的适应度

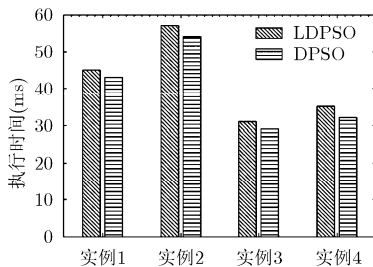


图9 不同选择实例下的执行时间

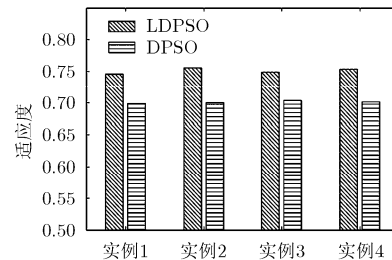


图10 不同选择实例下取得的适应度

参 考 文 献

[1] Yu Tao and Lin Kwei-jay. Service selection algorithms for composing complex services with multiple QoS constraints [C]. Proceedings of the 3th International Conference Service-Oriented Computing, Amsterdam, Netherlands, 2005: 130-143.

[2] 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, and Yang Fang-chun. Web service dynamic selection by the decomposition of global QoS constraints[J]. *Journal of Software*, 2011, 22(7): 1426-1439.

[3] Zeng Liang-zhao, Benatallah B, Ngu A H H, et al.

QoS-aware middleware for Web services composition[J]. *IEEE Transactions on Software Engineering*, 2004, 30(5): 311-327.

[4] Ardagna D and Pernici B. Adaptive service composition in flexible processes[J]. *IEEE Transactions on Software Engineering*, 2007, 33(6): 369-384.

[5] Watanabe A, Ishikawa F, Fukazawa Y, et al. Web service selection algorithm using Vickrey auction[C]. Proceedings of the 2012 IEEE 19th International Conference on Web Services, Honolulu, Hawaii, United States, 2012: 336-342.

[6] Hiratsuka N, Ishikawa F, and Honiden S. Service selection with combinational use of functionally-equivalent services[C]. Proceedings of the 2011 IEEE 18th International Conference

- on Web Services, Washington, United States, 2011: 97–104.
- [7] Canfora G, Di Penta M, Esposito R, *et al.* An approach for QoS-aware service composition based on genetic algorithms [C]. Proceedings of the 2005 conference on Genetic and Evolutionary Computation, Washington, United States, 2005: 1069–1075.
- [8] Alrifai Mohammad and Risse Thomas. Combining global optimization with local selection for efficient QoS-aware service composition[C]. Proceedings of the 18th International Conference on the World Wide Web, Madrid, Spain, 2009: 881–890.
- [9] Ramacher Rene and Mönch Lars. Cost-minimizing service selection in the presence of end-to-end QoS constraints and complex charging models[C]. Proceedings of the 2012 IEEE International Conference on Services Computing, Honolulu, Hawaii, United States, 2012: 154–161.
- [10] Geetha T and Sathya M. Modified Particle Swarm Optimization (MPSO) algorithm for Web Service Selection (WSS) problem[C]. Proceedings of the 2012 IEEE International Conference on Data Science & Engineering, Xiamen, China, 2012: 113–116.
- [11] Kang G, Liu J, Tang M, *et al.* An effective dynamic Web service selection strategy with global optimal QoS based on particle swarm optimization algorithm[C]. Proceedings of 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, Shanghai, China, 2012: 2280–2285.
- [12] Channa Nizamuddin, Li Shan-ping, Shaikh Abdul Wasim, *et al.* Constraint satisfaction in dynamic Web service composition[C]. Proceedings of the 16th International Workshop on Database and Expert Systems Applications, Copenhagen, 2005: 658–664.
- [13] Gao Ai-qiang, Yang Dong-qing, Tang Shi-wei, *et al.* QoS-driven Web service composition with inter service conflicts [C]. Proceedings of the 8th Asia-Pacific Web Conference, Harbin, China, 2006: 121–132.
- [14] Li Lei, Wei Jun, and Huang Tao. High performance approach for multi-QoS constrained Web services selection[C]. Proceedings of the 5th International Conference on Service-Oriented Computing, Vienna, Austria, 2007: 283–294.
- [15] Ai Li-feng and Tang Mao-lin. QoS-based Web service composition accommodating inter-service dependencies using minimal-conflict hill-climbing repair genetic algorithm[C]. Proceedings of the 4th IEEE International Conference on e-Science, Indianapolis, Indiana, United States, 2008: 119–126.
- [16] Ai Li-feng and Tang Mao-lin. A penalty-based genetic algorithm for QoS-aware Web service composition with inter-service dependencies and conflicts[C]. Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control and Automation, Vienna, Austria, 2008: 738–743.
- [17] Tang Mao-lin and Ai Li-feng. A hybrid genetic algorithm for the optimal constrained Web service selection problem in Web service composition[C]. Proceedings of the 2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain, 2010: 1–8.
- 梁中军: 男, 1983 年生, 博士生, 研究方向为服务计算及 Web 服务挖掘.
- 邹 华: 女, 1969 年生, 博士, 教授, 主要研究方向为软件体系结构、服务计算、网络智能.
- 杨放春: 男, 1957 年生, 博士, 教授, 博士生导师, 主要研究方向为下一代网络、业务生成环境及业务工程理论、网络安全.