

基于非均衡变异离散粒子群算法的 QoS 全局最优 Web 服务选择方法

王文彬¹, 孙其博¹, 赵新超², 杨放春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. 北京邮电大学理学院, 北京 100876)

摘 要: QoS 全局最优的 Web 服务选择是 NP 完全问题. 针对现有解决方案的不足, 提出了一种改进的离散粒子群算法. 该算法首先根据问题模型重新定义了粒子群算法中的位置、速度和算子操作, 然后对最优粒子进行非均衡变异, 并设计了非均衡变异概率函数, 同时在速度和位置更新中分别采用自适应权重调整机制和局部适应优先策略. 通过实验仿真, 与他人工作对比结果表明, 提出的算法在降低服务选择时间的同时, 提高了服务选择的质量.

关键词: Web 服务选择; QoS 全局最优; 非均衡变异; 自适应权重调整; 局部适应优先

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 12-2774-06

Web Services Selection Approach with QoS Global Optimal Based on Discrete Particle Swarm Optimization with Non-Uniform Mutation Algorithm

WANG Wen-bin¹, SUN Qi-bo¹, ZHAO Xin-chao², YANG Fang-chun¹

(1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China;

2. school of science, Beijing University of Posts and Telecommunications, Beijing, 100876, China)

Abstract: Global QoS optimal Web services selection is a NP complete problem. In order to overcome slow convergence of existing schemes, an improved discrete particle swarm optimization, called Discrete Particle Swarm Optimization with Non-Uniform Mutation Algorithm (short for DPSONUMA), is proposed. In DPSONUMA, we firstly redefine the particle position, velocity and update operations to make the algorithm more suitable for this problem. Then the best particle is introduced a mutation ability, and a mutation probability function is also designed. And weight factors of velocity will adaptively change according to the fitness values, which can improve convergence. In addition, a local fit first strategy is introduced, which lead to both quicker converge and better results. Experimental results show DPSONUMA costs less time but higher quality components are obtained for composite web services.

Key words: Web services selection; QoS global optimal; non-uniform mutation; adaptive weight adjustment; local fit first

1 引言

Web 服务计算^[1]近年来受到了工业界和学术界的广泛关注. Web 服务作为一种基本计算单元, 具有开放性、自治性、平台无关性等诸多优点, 并可以按照标准进行描述、发布、发现、绑定和执行. 但是, 单个 Web 服务通常提供比较单一的功能, 无法满足用户复杂的应用需求. 因此, 将多个 Web 服务组合起来, 提供新的、更复杂的、功能更强大的组合服务已经成为研究热点^[2,3].

随着 Web 服务应用范围的不断扩大, Web 服务数

量也不断增大, 不可避免地会出现大量的具有相同或相似功能的 Web 服务, 在需要组合出具有特定功能的组合服务时, 就会形成海量的组合方案; 同时, 用户在对组合服务提出功能要求之外, 也会对服务质量 (Quality of Service, QoS) 提出全局约束要求, 因此在这些海量组合方案中选择满足用户 QoS 限制的最优方案就成为组合服务中服务选择中的关键问题之一^[4].

QoS 全局最优 Web 服务选择问题是被证明了的 NP 完全问题^[5], 受到学者的普遍关注. 文献[6,7]根据每个任务中候选服务 QoS 的属性值进行加权和排序, 选择

加权和大的候选服务作为组合服务中的一部分,属于局部最优化的选择方法,无法选择出具有 QoS 全局最优的组合服务.文献[8,9]通过引入整数规划来求解,将所有的候选方案转换为线性的 QoS 属性和约束条件进行计算,采用穷尽计算的方法,当组合规模大时,算法的计算量巨大,且整数规划要求目标函数和约束条件是线性的,扩展性较差,因此限制了此类方法的实用性.文献[10,11]采用了遗传算法,将服务组合的方案编码为染色体,设计了适应度函数,通过群体适应度值寻优的方法进行服务选择,算法可以在一定时间内求得近似最优解,避免了穷尽计算的局限性,但是没有考虑算法的早熟问题;文献[12~14]基于文献[10,11],通过采取多样性保持策略,在一定程度上解决了遗传算法的早熟问题,算法得到的最优解质量和算法执行时间较文献[10,11]有所提高,但是文献[10~14]所采用的遗传算法中染色体信息共享、整个种群比较平均地向最优区域移动的性质,决定了算法的收敛速度慢,且随着服务选择问题规模的增加,算法执行时间呈指数增长,无法满足用户实时性的要求.

粒子群优化算法^[15] (Particle Swarm Optimization , PSO)作为一种功能强大的群体智能优化算法,由于其信息传递的单向性(最优粒子信息告诉其它粒子,但最优粒子并不能得到其它粒子的信息),使得粒子快速地收敛,可以大大降低算法的执行时间.但是目前,利用 PSO 解决 QoS 全局最优的 Web 服务选择问题的文献很少^[16],这主要是因为 PSO 算法快速收敛的同时也容易使粒子聚集,导致粒子群体多样性的下降,容易陷入局部最优^[17],从而限制了问题求解的质量.

基于上述分析,本文提出了一种非均衡变异离散粒子群优化算法 (Discrete Particle Swarm Optimization with Non-Uniform Mutation Algorithm, DPSONUMA) 用于解决 QoS 全局最优 Web 服务选择问题,该算法借鉴遗传算法中基因变异的思想,对全局最优粒子进行非均衡变异,增加粒子群在搜索过程中种群的多样性,便于搜索更优的解;同时,自适应地调整粒子速度更新因子,并在位置更新中采取局部适应优先策略,提高了算法的收敛性.实验表明,与已有成果相比,该算法在服务选择的执行时间和求解质量上都有明显提高.

2 QoS 全局最优的 Web 服务选择问题描述

2.1 QoS 定义

根据国际标准 ITU E.800^[18]对 QoS 的定义,QoS 由一些非功能属性组成,如服务价格 (Price P)、服务执行时间 (Response Time, T)、服务可用性 (Availability, A)和服务信誉 (Reputation, Rep)等,反映了用户对服务非功能属性的需求.在不影响主要结论的情况下,本文只考虑

上述 4 种 QoS 属性.

2.2 组合 Web 服务流程模型及其 QoS 计算方法

组合 Web 服务的功能可以分成多个子功能,本文称这些子功能为任务(task,是构成组合服务的基本功能单元,具有功能描述、输入接口与输出接口描述,不代表具体的 Web 服务).而组合 Web 服务实际上是一个基于服务的工作流,根据各个任务之间的逻辑关系,可以把组合服务流程分解成四种基本结构^[19]:顺序结构(图 1(a));循环结构(图 1(b));分支结构(图 1(c));并行结构(图 1(d)).

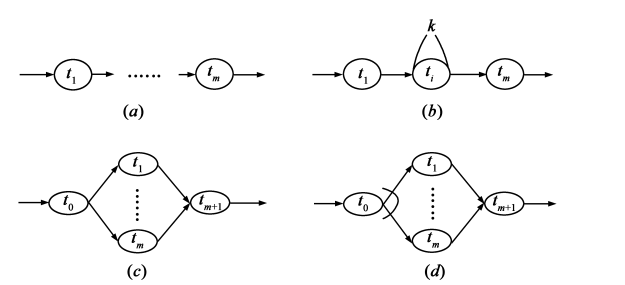


图1 组合服务的四种基本结构

组合服务是四种基本结构的组合和嵌套,因此组合 Web 服务的 QoS 可通过基本结构的 QoS 计算方法(如图 2 所示)获得.

结构	Price	Availability	Response time	Reputation
顺序	$\sum_{i=1}^m P_{usi}$	$\prod_{i=1}^m A_{usi}$	$\sum_{i=1}^m T_{usi}$	$\prod_{i=1}^m Rep_{usi}$
循环	kP_{us}	A_{us}^k	kT_{us}	Rep_{us}^k
并行	$\sum_{i=1}^m P_{usi}$	$\prod_{i=1}^m A_{usi}$	$\max \{ T_{usi} \}$	$\prod_{i=1}^m Rep_{usi}$
分支	P_{usi}	A_{usi}	T_{usi}	Rep_{usi}

图2 基本结构的 QoS 计算方法

2.3 QoS 全局最优的 Web 服务选择模型

组合 Web 服务的每个任务可以由不同的候选 Web 服务来完成,这些候选服务具有不同的 QoS 属性参数值,因此对于同一个组合 Web 服务,可形成海量组合方案.QoS 全局最优的 Web 服务选择问题,就是要在这些组合方案中,给每个任务选择合适的候选服务,使得组合服务的 QoS 在满足用户约束的前提下,达到全局最优的效果.本文采用多属性加权求和的方法对组合 Web 服务进行综合质量评价,如式(1).需要说明的是,QoS 各属性量纲的不同、对整体效果的贡献不同,须首先对其进行归一化处理^[8],由于非本文工作重点,在此不再赘述.

$$f(x) = \sum_{i=1}^4 w_i Q_i$$

(1)

其中 $w_i \in (0,1)$ 表示用户对相应 QoS 属性参数的关注程度,且 $\sum_{i=1}^4 w_i = 1$.

因此 QoS 全局最优的 Web 服务选择问题实际上是带约束条件的最优化问题,是被证明了的 NP 完全问题.粒子群算法作为一种智能优化算法,具有并行计算、群体寻优的特点;同时粒子群算法不需要与背景知识相关的启发式信息,只需确定其优化目标函数和约束条件即可.而 QoS 全局最优的 Web 服务选择问题的目标函数与约束条件如式(2).

$$\begin{cases} \text{Max}\{f(x)\} \text{ (或者 } \min\{f(x)\}) \\ Q_i \leq Q_i^0 \text{ (或者 } Q_i \geq Q_i^0) \end{cases} \quad (2)$$

其中 Q_i^0 表示 QoS 的全局限制值.

3 基于 DPSONUMA 的 QoS 全局最优 Web 服务选择方法

3.1 PSO 算法原理

PSO 是由 Kennedy 和 Eberhart 提出的一种新的全局优化进化算法^[15].算法首先生成一群粒子,粒子本身具有两个参数:位置和速度.粒子的位置表示优化问题的一个解,粒子的速度表示位置变化的方向,指导粒子在迭代过程中向最优解的运动.粒子同时具有记忆功能,每次迭代时,粒子根据本身达到过的最优位置 X_{pbest} 和整个粒子群到达过的最优位置 X_{gbest} 来调整当前的位置.其速度和位置更新公式如下:

$$V_{i+1} = wV_i + c_1 r_1 (X_{\text{pbest}} - X_i) + c_2 r_2 (X_{\text{gbest}} - X_i) \quad (3)$$

$$X_{i+1} = X_i + V_{i+1} \quad (4)$$

式中 X_i 表示第 i 个粒子当前的位置, V_i 表示其当前速度.式(3)中 w 表示粒子保持惯性的权重; c_1 、 c_2 是加速度系数,分别表示粒子向本身最优位置和全局最优位置移动的趋势, r_1 、 r_2 是(0,1)之间的随机数.

PSO 算法在解决带约束条件的数值优化问题时取得了很好的效果^[20],但是在解决离散问题时需要根据具体问题重新定义算法的参数与运算操作,此时称之为离散 PSO (Discrete PSO, DPSONUMA).同时,PSO 算法中粒子群在迭代过程中,迅速向最优粒子位置聚集,限制了粒子群的多样性,容易导致算法陷入早熟,因而降低问题求解的质量.为此,本文提出了一种非均衡变异离散粒子群算法来解决 QoS 全局最优的 Web 服务选择问题.

3.2 DPSONUMA 算法描述

3.2.1 算子定义

设 (x_1, x_2, \dots, x_m) 为粒子在 m 维离散空间中的位置, x_i (i 为整数,且 $1 \leq i \leq m$) 为 $0 \sim N_i$ 之间的整数;设 F 表示粒子适应度函数,则 $F(X_i)$ 为第 i 个粒子在位置 X_i 的适应度值;设 (v_1, v_2, \dots, v_m) 为粒子的速度 V_i , v_i 的取值为 0 或者 1.

定义 1 减法 \odot : $X_i \odot X_j$, 表示离散空间位置 X_i 与 X_j 之间的差,若对应维度的值相同则操作结果取 1, 否

则取 0.

如: $(1, 2, 3, 4, 3, 5) \odot (1, 3, 2, 4, 5, 5) = (1, 0, 0, 1, 0, 1)$.

定义 2 系数: 系数表示粒子速度的更新中,保持当前速度惯性、向自身最优位置和全局最优位置运动的概率, $P_i V_i$ 表示粒子以 P_i 的概率保持速度 V_i .

定义 3 加法 \oplus : \oplus 表示粒子的速度更新过程中,保持自身速度惯性,受本身最优位置和全局最优位置影响改变速度的操作,代表粒子位置寻优的方向. $P_i V_i \oplus \dots \oplus P_j V_j$ 表示在对应维度上以 P_i 的概率按照 V_i, \dots, P_j 的概率按照 V_j 的速度更新,且操作项的概率和为 1,

即 $\sum_{i=1}^n P_i = 1$.

如: $0.1(1, 0, 0, 1, 1, 1) \oplus 0.9(1, 0, 1, 0, 1, 0) = (1, 0, \times, \times, 1, \times)$, 其中第 3 维以 0.1 的概率为 0, 0.9 的概率为 1, 依次类推.

定义 4 乘法 \otimes : 乘法表示在离散空间中粒子在当前位置 X_i 上,按照速度 V_i 前行的操作.位置与位置 ($X_i \otimes X_j$),速度与速度 ($V_i \otimes V_j$) 均没有意义.算子运算规则为:按照对应位实行 0 变, 1 不变的操作.

如: $(1, 2, 4, 3, 2, 5) \otimes (1, 0, 1, 0, 0, 1) = (1, \times, 4, \times, \times, 5)$, 其中 \times 表示位置更新后的数值,根据算法采取的策略随机选取.

因此得出离散粒子群优化算法的位置和速度更新公式如下:

$$V_{i+1} = P_1 V_i \oplus P_2 (X_{\text{pbest}} \odot X_i) \oplus P_3 (X_{\text{gbest}} \odot X_i) \quad (5)$$

$$X_{i+1} = X_i \otimes V_{i+1} \quad (6)$$

其中 P_1, P_2, P_3 的计算方法见 3.2.2.

3.2.2 自适应权重调整机制

速度决定了粒子运动的方向和快慢(太快可能错过最优解区域,太慢则影响求解效率),而惯性与加速度权重直接影响粒子的速度更新,因此,本算法设计了一种动态的权重调整机制,即根据粒子在运动过程中的适应度值来确定其系数,计算公式如下.

$$P_1 = \frac{F(X_i)}{F(X_i) + F(X_{\text{pbest}}) + F(X_{\text{gbest}})} \quad (7)$$

$$P_2 = \frac{F(X_{\text{pbest}})}{F(X_i) + F(X_{\text{pbest}}) + F(X_{\text{gbest}})} \quad (8)$$

$$P_3 = \frac{F(X_{\text{gbest}})}{F(X_i) + F(X_{\text{pbest}}) + F(X_{\text{gbest}})} \quad (9)$$

一般来讲,粒子在迭代的过程中,有 $F(X_i) \leq F(X_{\text{pbest}}) \leq F(X_{\text{gbest}})$ 成立.在迭代过程中,粒子以一定的速度向最优粒子靠拢.该速度由表征粒子当前位置、自身最优位置和粒子群全局最优位置的优劣的 $F(X_i)$ 、 $F(X_{\text{pbest}})$ 和 $F(X_{\text{gbest}})$ 共同决定,反映了速度更新过程中,粒子自适应地保持当前速度惯性、向自身最优位置

和全局最优位置运动。

3.2.3 非均衡变异策略设计

在粒子群迭代的过程中,最优粒子的运动方向是盲目的,因此算法容易陷入局部优化,且不易跳出。文献[21,22]借鉴基因变异的思想,对最优粒子进行变异操作,使最优粒子达到最优位置后,去探索新的解空间。较高的变异率会扩大算法的搜索空间,更容易搜索到全局最优解,但是却降低了算法的收敛速度;变异率低,虽然收敛较快,但容易陷入局部优化。因此,本文在之前工作^[20]的基础上,提出了一种总体趋势递减,并且在递减中概率随机浮动的非均衡变异策略,使得最优粒子在算法初期全方位的搜索解空间,并随着迭代次数的增加,变异率降低,便于算法快速收敛。在此过程中,随机的高变异率使得最优粒子仍然存在变异和搜索新的解空间的可能。概率函数设计如式(10)。

$$f(k, r) = \sigma * (1 - r^{(1 - \frac{k}{K})^b}) \quad (10)$$

图3显示了随着迭代次数的增加非均衡变异函数的变化($\sigma = 0.5, b = 5, K = 1000$)。

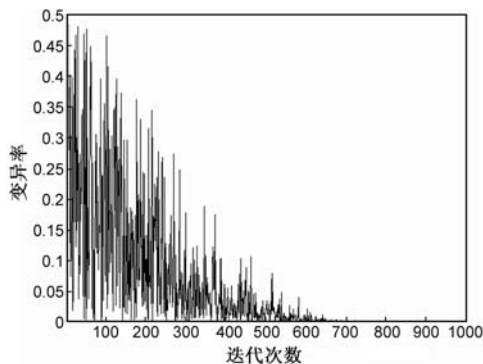


图3 非均衡变异概率函数曲线图

3.2.4 局部适应优先策略

由定义4可知,位置在更新过程中,当相应维度上的值更新时,可以采用不同策略进行,随机选择是较为常见的方式。但是,随机选择会影响算法整体的收敛性。因此,提出了一种局部适应优先的策略。

为便于表述,将粒子位置中各维度上的值称为局部位置,其适应度称为局部适应度。所谓局部适应优先策略就是在局部位置的选择上,以较大的概率选择局部适应度值大的局部位置。如果局部位置有 m 个候选位置,那么每个候选位置被选择的概率为 $\frac{f'_i}{\sum_{j=1}^m f'_j}$ 。其中 f'_i 表示第 i 个局部位置的局部适应度。局部适应度与全局适应度函数相同。

3.3 基于 DPSONUMA 的服务选择方法

根据组合 Web 服务的结构关系,首先将其任务按照顺序进行编号,同时对每个任务的候选服务进行编

号,这样,就可用一维定长的整数编码方式表示组合 Web 服务的组合方案。如(1,3,4,9,0,4)可表示该组合方案第一个任务选择编号为1的候选服务,第二个任务选择编号为3的候选服务,依此类推。0则表示分支结构中不被选择的分支上的任务。

因此,如果将服务组合方案映射为 DPSONUMA 算法中粒子的位置,候选服务对应为粒子的局部位置,组合 Web 服务的质量评价函数当作适应度函数,那么在海量组合方案中搜索最优方案过程即为 DPSONUMA 算法过程。

算法1 基于 DPSONUMA 的 Web 服务选择算法

Step 1 初始化:设定粒子群的规模数 N ,算法执行的最大迭代次数 K 。根据组合 Web 服务的流程逻辑为粒子随机生成位置参数 X_i ,随机生成初始速度 V_i 。检查粒子位置的可行性(粒子位置须正确反映组合服务逻辑关系,且须满足用户的全局 QoS 限制);根据式(1)计算每个粒子的适应度 F_i ,并排序得到全局最优初始位置 X_{gbest} ,同时设定粒子个体最优初始位置 X_{pbest} 。

Step 2 算法迭代控制: If ($k = K$), goto Step 4; else continue;

Step 3 算法迭代:

(1) 对于粒子群中的每个粒子,计算 $f(X_i)$, if ($f(X_i) > f(X_{pbest})$) 则 $X_{pbest} = X_i$; if ($f(X_{pbest}) > f(X_{gbest})$), 则 $X_{gbest} = X_{pbest}$;

(2) 按照式(5)进行速度更新,式(6)进行位置更新;位置更新中候选服务的选择按照局部适应优先原则进行;

(3) 根据式(10)计算当前最优粒子的变异概率,粒子按照变异概率进行变异(变异粒子位置的生成按照局部适应优先原则,且变异后的粒子需要进行可行性检查);

$k + KG - * 3 +$; goto Step 2.

Step 4 结果输出:输出求得的组合方案(粒子位置)及其 QoS 综合评价值(适应度值)。

4 仿真实验结果与分析

为了验证本文算法的有效性和合理性,分别从适应度函数的收敛性、算法的执行时间、求解质量三个方面对算法的性能进行了仿真实验,并与文献[12]算法进行了比较。比较工作采用相同的硬件环境,即 CPU 为 Intel 酷睿双核(T6570)2.1GHz,内存 1.99GB,操作系统为 Windows XP,开发语言为 C++,开发环境为 VS2005。两种算法采用相同的随机产生的初始数据:随机产生 20~40 个候选服务并在一定范围内随机产生其 QoS 属性参数值。实验结果数据为算法运行 50 次所得数据平均值。

实验 1 算法收敛性对比

图 4 展示了当组合服务任务数为 30 情况下两种算法执行过程中适应度值随时间的变化关系.

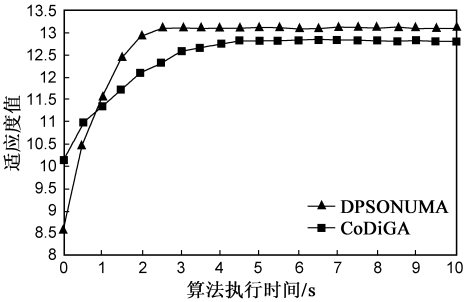


图4 适应度函数的收敛

如图 4 所示,CoDiGA 算法初始值较高,但收敛速度比较缓慢,这是由于其随机性地交叉和变异操作所致,同时其采用的初始种群控制机制使得算法种群初始适应度较高;而 DPSONUMA 因为其粒子位置的更新是有方向的,粒子迅速向最优粒子靠拢,所以收敛速度较快;同时由于 DPSONUMA 采用的非均衡变异、局部适应先策略使得算法早起收敛速度很快,后期非常稳定地收敛于一个较好的适应度值.

实验 2 算法执行时间与求解质量对比

对于进化算法,执行时间和求解质量也是衡量其性能的主要指标.图 5 描述了在不同任务数下的两种算法执行时间的对比关系.算法执行时间为算法“第一次命中时间”,即在迭代过程,第一次达到最优解的时间.

如图 5 所示,DPSONUMA 的执行时间优于 CoDiGA,特别是随着任务数的增多,CoDiGA 算法执行时间增长的幅度远远大于 DPSONUMA.

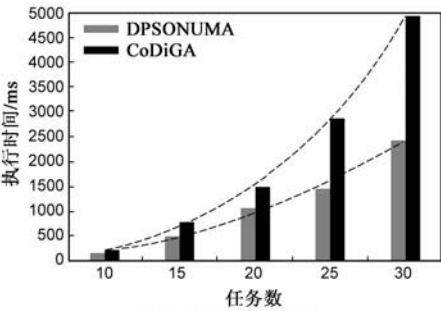


图5 执行时间比较

表 1 给出了收敛时算法得到的最优解.

表 1 算法适应度值对比

任务数		10	15	20	25	30
适应度值	DPSONUMA	4.362	6.538	8.867	10.95	13.10
	CoDiGA	4.10	6.383	8.706	10.70	12.81
		45	1	4	38	93

由表 1 的统计数据可知,由于 DPSONUMA 算法中

最优粒子非均衡变异扩大了粒子群搜索最优解空间的范围,粒子在位置更新中的局部适应优先策略也使得粒子群在各维度上选择最优解可能性增强,在缩短算法执行时间的同时,增加了算法的求解质量.而 CoDiGA 尽管采取了种群多样性保持策略,但是其对初始种群的依赖性和较小的变异率也使得其易陷入局部优化,导致求解质量相对较低.

5 结论

QoS 全局最优的 Web 服务选择问题已经成为服务计算领域比较活跃的研究方向.本文提出的 DPSONUMA 主要贡献在于:(1)根据问题模型,重新定义了粒子群算法中的算子操作,使算法更适合解决该问题;(2)对粒子群中最优粒子实施变异操作,并设计了非均衡的变异概率函数,便于在迭代的初期大范围地搜索,后期迅速收敛;(3)提出了一种自适应的权重调整机制,动态地确定粒子保持当前速度、根据个体经验和向全局最优粒子学习的权重,有利于提高收敛速度;(4)在粒子的位置更新中采用局部适应优先策略,采用概率的方法便于粒子群多样性的保持,大大提高了算法的求解质量.实验结果表明,DPSONUMA 算法在降低了服务选择时间的同时,提高了组合 Web 服务的质量.

参考文献:

[1] M P Papazoglou, D Georgeakopoulos. Service oriented computing: Introduction [J]. Communications of the ACM, 2003, 46 (10): 24 – 28.

[2] N Milanovic, M Malek. Current olutions for Web service composition [J]. IEEE Internet Computing, 2004, 8(6): 51 – 59.

[3] B Orriëns, J Yang, M P Papazoglou. Model driven service composition [A]. The First International Conference on Service Oriented Computing [C]. Trento: Springer, 2003. 75 – 90.

[4] L Z Zeng, B Benatallah, M Dumas, et al. Quality driven Web services composition [A]. Proceedings of the 12th International Conference on World Wide Web [C]. Budapest: ACM, 2003. 411 – 421.

[5] T Yu, K J Lin. Service selection algorithms for composing complex services with multiple QoS constraints [A]. Proceedings of the 3rd International Conference on Service Oriented Computing [C]. Amsterdam: Springer, 2005. 130 – 143.

[6] B Benatallah, M Dumas, Q Z Sheng, et al. Declarative composition and peer-to-peer provisioning of dynamic Web services [A]. Proceedings of the 18th International Conference on Data Engineering [C]. San Jose: IEEE, 2002. 297 – 308.

[7] Y Liu, A Ngu, L Z Zeng. QoS computation and policing in dynamic Web service selection [A]. Proceedings of the 13th International Conference on World Wide Web [C]. New York:

- ACM, 2004. 66 – 73.
- [8] L Z Zeng, B Benatallah, A Ngu, et al. QoS-aware middleware for Web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311 – 327.
- [9] D Ardagna, B Pernici. Adaptive service composition in flexible processes[J]. IEEE Transactions on Software Engineering, 2007, 33(6): 369 – 384.
- [10] L J Zhang, B Li, T Chao, et al. On demand Web services-based business process composition[A]. International Conference on System, Man, and Cybernetics[C]. Washington: IEEE, 2003. 4057 – 4064.
- [11] G Canfora, M D Penta, R Esposito, et al. A lightweight approach for QOS-aware service composition[A]. Proceedings of the 2nd International Conference on Service oriented Computing[C]. New York: ACM, 2004. 36 – 47.
- [12] Y Ma, C W Zhang. Quick convergence of genetic algorithm for QoS-driven Web service selection[J]. Computer Networks, 2008, 52(5): 1093 – 1104.
- [13] C W Zhang, S Su, J L Chen. GAMi: Genetic algorithm-based middleware for QoS-Aware Web services selection[J]. Chinese Journal of Electronics, 2007, 16(3): 537 – 543.
- [14] 刘书雷, 刘云翔, 张帆, 唐桂芬, 景宁. 一种服务聚合中 QoS 全局最优服务动态选择算法[J]. 软件学报, 2007, 18(3): 646 – 656.
- Liu Shulei, Liu Yunxiang, Zhang Fan, Tang Guifen, Jing Ning. A dynamic Web services selection algorithm with QoS Global optimal in Web services composition[J]. Journal of Software, 2007, 18(3): 646 – 656. (in Chinese)
- [15] J Kennedy, R Eberhart. Particle swarm optimization[A]. International Conference on Neural Networks[C]. Washington: IEEE, 1995. 1942 – 1948.
- [16] C H Hu, X H Chen, X M Liang. Dynamic services selection algorithm in Web services composition supporting cross-enterprises collaboration[J]. Journal of Central South University of Technology, 2009, 16(2): 269 – 274.
- [17] 张长胜, 孙吉贵, 欧阳丹彤. 一种自适应离散粒子群算法及其应用研究[J]. 电子学报, 2009, 37(2): 299 – 304.
- Zhang Changsheng, Sun Jigui, Ouyang Dantong. A self-adaptive discrete particle swarm optimization algorithm[J]. Acta Electronica Sinica, 2009, 37(2): 299 – 304. (in Chinese)
- [18] ITU-T Recommendation E. 800. Terms and Definitions Related to Quality of Service and Network Performance Including Dependability[S].
- [19] WFM. Workflow management coalition terminology & glossary[R]. Brussels: Workflow Management Coalition, 1999.
- [20] X C Zhao, X S Gao, Z C Hu. Evolutionary programming based on non-uniform mutation[J]. Applied Mathematics and Computation, 2007, 192(1): 1 – 11.
- [21] N Higashi, H Iba. Particle swarm optimization with gaussian mutation[A]. Proceedings of the Swarm Intelligence Symposium[C]. Indianapolis: IEEE, 2003. 72 – 79.
- [22] T Krink, M Løvbjerg. The LifeCycle model: Combining particle swarm optimisation, genetic algorithms and hillClimbers[A]. Proceedings of Parallel Problem Solving from Nature[C]. Granada: Springer, 2002. 621 – 630.

作者简介:



王文彬 男, 1985 年生于河南安阳市. 现为北京邮电大学博士研究生, 主要研究方向为移动服务计算技术.

E-mail: buptwwb@gmail.com

孙其博 男, 1975 年生于河南郑州市. 现为北京邮电大学副教授, 主要研究方向为网络智能化和服务计算.

赵新超 男, 1976 年生于河南商丘市. 现为北京邮电大学副教授, 主要研究方向为智能优化算法及其应用.

杨放春 男, 1957 年 3 月生于北京, 现为北京邮电大学教授, 博士生导师, 主要研究方向为网络智能化与服务、通信软件和网络安全.