

# 基于改进粒子群算法的 Web 服务组合推优方法

吴黎兵<sup>a,b</sup> 杨 科<sup>a</sup> 聂 雷<sup>a</sup> 彭红梅<sup>b</sup>

(武汉大学 a 计算机学院, b 软件工程国家重点实验室, 湖北 武汉 430072)

**摘要** 针对半自动的 Web 服务组合模型,为了尽可能多地发现服务质量处在 Pareto 前端的服务组合供用户参考使用,提出了一种基于改进粒子群算法(MPSO)的 Web 服务组合推优方法. 结合服务组合问题给出了粒子适应度评价函数以及群体多样性的计算模型. 为了改善粒子群算法存在的早熟问题并且发现更多服务质量处在 Pareto 前端的组合服务,给出了受群体多样性指导的速度更新方法和惯性权重模型. 针对指导粒子飞行的关键组合服务,给出了它们的寄存方法. 最后通过实验从有效率和精确度及平衡性方面验证了基于 MPSO 的 Web 服务组合推优方法的有效性.

**关键词** Web 服务组合; 服务质量; Pareto 支配; 改进的粒子群算法; 寄存集

**中图分类号** TP399 **文献标志码** A **文章编号** 1671-4512(2014)10-0069-05

## Web service composition recommendation based on modified particle swarm optimization

Wu Libing<sup>a,b</sup> Yang Ke<sup>a</sup> Nie Lei<sup>a</sup> Peng Hongmei<sup>b</sup>

(a School of Computer Science; b State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

**Abstract** Aimed at semi-automatic Web service composition model, in order to find composition service whose quality of services were on the Pareto front for reference and use as more as possible, a method about Web services composition pushing excellent was presented based on modified particle swarm optimization (MPSO). Particles fitness evaluation function model and diversity model of swarm were given considering service composition problem. In order to improve the existing precocity problem and find more Web services whose quality of services were on Pareto front for reference, speed and inertia weight update model guided by the diversity of swarm were given and the storage method of key services composition that could affect algorithm was given also. Finally, the effectiveness of Web services composition recommendation based on PSDO on pushing excellent services composition was verified through experiment.

**Key words** Web service composition; quality of services; Pareto dominate; modified particle swarm optimization; deposit set

近年来,随着 Web 服务标准的不断完善以及支持 Web 服务的企业级软件平台的成熟发展,尤其是蓬勃兴起和广泛应用的 SOC 和 SOA 概念<sup>[1]</sup>,很多商业组织和企业加入到软件服务化 SaaS 的行列中,纷纷将其业务功能以及组件进行

包装,并以 Web 服务的形式进行发布. 目前发布在网络上的服务由于结构简单并且功能单一,无法满足企业和用户的需要. 如何将网络上各种功能的服务组织起来,形成功能强大的企业级流程服务,从而达到企业的某种商业目标已经成为

收稿日期 2014-01-02.

作者简介 吴黎兵(1972-),男,教授, E-mail: wu@whu.edu.cn.

基金项目 国家自然科学基金资助项目(61272112);武汉市科技攻关应用基础项目.

Web 服务发展过程中的一个重要环节,也是 SOC 与 SOA 成功应用和实施的關鍵<sup>[2]</sup>. 文献[3]设计了一种面向动态 Web 服务组合的粒子群算法,为了解决粒子群算法容易陷入早熟的问题,定义了粒子无希望以及重希望准则来保证粒子群的多样性;文献[4]提出了服务质量全局最优服务动态选择方法,将服务的组合过程转化为一个带约束条件的多目标优化问题. 本研究针对半自动服务组合模型<sup>[4-5]</sup>,围绕基于服务质量的组合服务推优问题,提出了一种改进粒子群算法(MPSO)的 Web 服务组合推优方法,并建立了问题处理的相关模型以及处理流程,最后通过实验验证了该方法的有效性.

## 1 问题描述

Web 服务可以简要地描述为  $\omega = \{F, Q\}$ , 其中:  $F$  描述服务的功能特性;  $Q$  描述服务的服务质量,记  $Q = \{q_1, q_2, \dots, q_m\}$ ,  $q_m$  对应于服务质量第  $m$  个维度的属性. 记服务组合的抽象逻辑模型为  $W = \{ASN, IR, Map\}$ , 其中:  $ASN = \{Cans_i \mid 1 \leq i \leq n\}$  表示模型中业务活动节点的集合,  $Cans_i$  为第  $i$  个业务活动对应的服务实例集合,  $n$  为业务活动的个数;  $IR$  描述各业务活动之间的关联关系;  $Map$  描述业务活动与对应服务实例集合中实例的映射关系.

针对基本的业务活动关联关系,表 1 给出了服务组合模型部分服务质量的计算方法<sup>[6]</sup>. 其中:  $k$  为并行结构中关键分支 key 包含的业务活动个数;  $b$  为并行结构或选择结构中的分支数;  $u_i$  为选择结构中第  $i$  个分支被选择执行的概率,  $\sum_{i=1}^b u_i = 1$ ;  $l$  为循环结构的循环次数;  $T_i, c_i, A_i, R_i$  和  $S_i$  分别为第  $i$  个分支服务质量的时间、费用、可用性、可靠性及声誉度.

表 1 Web 组合逻辑模型目标服务质量的计算方法

服务质量	顺序	并行	选择	循环
时间	$\sum_{i=1}^n T_i$	$\sum_{i=1}^k T_i$	$\sum_{i=1}^b (u_i T_i)$	$l \sum_{i=1}^n T_i$
费用	$\sum_{i=1}^n c_i$	$\sum_{i=1}^k c_i$	$\sum_{i=1}^b (u_i c_i)$	$l \sum_{i=1}^n c_i$
可用性	$\prod_{i=1}^n A_i$	$\prod_{i=1}^k A_i$	$\sum_{i=1}^b (u_i A_i)$	$(\prod_{i=1}^n A_i)^l$
可靠性	$\prod_{i=1}^n R_i$	$\prod_{i=1}^k R_i$	$\sum_{i=1}^b (u_i R_i)$	$(\prod_{i=1}^n R_i)^l$
声誉度	$\sum_{i=1}^n S_i/n$	$\sum_{i=1}^k (S_i/b)$	$\sum_{i=1}^b (u_i S_i/b)$	$\sum_{i=1}^n S_i/n$

下面给出 Pareto 相关概念<sup>[7]</sup>, 记两个目标向

量分别为  $F(u) = (f_1(u), f_2(u), \dots, f_k(u))$ ,  $F(v) = (f_1(v), f_2(v), \dots, f_k(v))$ , 若  $\forall i \in \{1, 2, \dots, k\}$ , 都有  $f_i(u) \leq f_i(v)$ , 并且  $\exists i \in \{1, 2, \dots, k\}$ , 满足  $f_i(u) < f_i(v)$ , 则称目标  $v$  Pareto 占优于  $u$  或者称  $v$  支配  $u$ , 记为  $v \succ u$ . 假设对于集合  $S$ , 若有  $P \subseteq S$ , 并且  $\forall v \in S - P, \neg \exists u \succ v$ , 则称集合  $P$  处在集合  $S$  的 Pareto 前端; 当集合  $P$  取值最大时, 则称其为集合  $S$  的 Pareto 最优集.

围绕服务组合问题, 针对半自动的 Web 服务组合逻辑模型, 解决如何利用粒子群算法快速寻优特点, 从可能存在的所有组合服务中尽可能多地发现服务质量处在 Pareto 前端并且接近 Pareto 最优集的 Web 服务组合, 把这些组合服务提供给用户作为参考使用.

## 2 基于 MPSO 的服务组合推优

### 2.1 相关模型建立

将共享函数的概念引入面向 Web 服务组合推优的粒子群算法的适应度函数<sup>[8]</sup>中. 通过  $m$  维决策空间记录逻辑模型中各业务活动, 记粒子群中第  $i$  个粒子的位置为  $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ , 它描述一个具体的服务组合实例, 其中  $x_{ij}$  表示第  $i$  个粒子描述的服务组合实例在逻辑模型的第  $j$  个业务活动上对应的服务实例的编号. 记  $f(x_i) = (f_1(x_i), f_2(x_i), \dots, f_k(x_i))$  为第  $i$  个粒子在  $k$  维目标服务质量空间的计算值, 其中  $f_j(x_i)$  表示粒子  $i$  在第  $j$  维服务质量属性上的计算值.

记  $d_{ij} = |f_1(x_i) - f_1(x_j)| + |f_2(x_i) - f_2(x_j)| + \dots + |f_k(x_i) - f_k(x_j)|$  为粒子  $i$  与粒子  $j$  之间的表现型距离. 粒子间的共享函数记为

$$s_f(d_{ij}) = \begin{cases} 1 & (d_{ij} \leq \sigma_s); \\ 0 & (d_{ij} > \sigma_s), \end{cases}$$

式中  $\sigma_s$  为共享参数. 当粒子间的表现型距离在给定阈值内时, 则认为它们之间的临近程度很高. 记

粒子  $i$  在群体中的共享度值为  $s_d(i) = \sum_{j=1}^p s_f(d_{ij})$ , 其中  $p$  为群体中粒子个数, 粒子的共享度反映当前粒子所在区域的密集程度.

定义 1 粒子被支配次数. 粒子  $i$  被支配的次数  $D(i)$  是指当前种群中 Pareto 支配粒子  $i$  的粒子个数, 记

$$D(i) = \sum_{j=1}^p d(i, j), d(i, j) = \begin{cases} 1 & (i < j); \\ 0 & (i > j). \end{cases}$$

定义 2 粒子适应度评价函数. 记粒子的适应度评价函数值  $F(i) = -(1 + s_d(i))(1 + D(i))$ ,

并伴随着  $D(i)$  以及  $s_d(i)$  的增大,粒子的适应度值减小。

**定义 3** 多样性. 对于当前粒子群,记其多样性为  $V=1/(1+e^{-M})$ ,且  $M=\sum_{j=1}^m U(\bigcup_{i=1}^p x_{ij})/(pn)$ ,其中  $U(\cdot)$  描述集合中互异元素的个数。

**定义 4** 寄存集. 寄存集用来存储发现的组合服务,通过寄存集 GBCS 和 GWCS 以及 ND 分别存储服务质量全局最优和最差的服务组合实例,以及服务质量不被其他粒子支配的服务组合实例。

## 2.2 算法原理

MPSO 算法是一种基于粒子群多样性指导<sup>[9-10]</sup>的改进粒子群算法,当群体的多样性  $V$  高于阈值  $d_{\text{high}}$  时,粒子间相互吸引收缩,向服务质量处在 Pareto 前端的位置靠拢;当群体的多样性  $V$  低于阈值  $d_{\text{low}}$  时,粒子间执行排斥操作,避免粒子群集中收敛于局部最优,出现早熟现象,从而促进粒子群能够发现更多服务质量处在 Pareto 前端的服务组合。借鉴运动学速度的矢量分解方法,把粒子在各个业务活动节点的速度分解到各个服务质量维度上,具体更新方式如下。记

$$v_{i,j,q}^{t+1} = \begin{cases} \omega v_{i,j,q}^t + c_1 r_1 (p_{i,j,q}^{\text{best}} - x_{i,j,q}^t) + c_2 r_2 (g_{j,q}^{\text{best}} - x_{i,j,q}^t) & (V > d_{\text{high}} \text{ 更新方式同上次,} \\ d_{\text{low}} < V \leq d_{\text{high}}), \\ \omega v_{i,j,q}^t + c_1 r_1 (p_{i,j,q}^{\text{worst}} - x_{i,j,q}^t) + c_2 r_2 (g_{j,q}^{\text{worst}} - x_{i,j,q}^t) & (V \leq d_{\text{low}}); \end{cases}$$

$$v_{i,j,q}^{t+1} = \begin{cases} v_{\text{max}} & (v_{i,j,q}^{t+1} > v_{\text{max}}), \\ -v_{\text{max}} & (v_{i,j,q}^{t+1} < -v_{\text{max}}), \end{cases}$$

式中:  $v_{i,j,q}^t$  和  $x_{i,j,q}^t$  分别为第  $t$  次迭代后粒子  $i$  第  $j$  个业务活动服务质量属性  $q$  的速度和位置;  $p_{i,j,q}^{\text{best}}$  和  $p_{i,j,q}^{\text{worst}}$  分别为第  $i$  个粒子记录的服务质量历史最优和最差的组合服务的第  $j$  个业务活动对应的服务实例所描述的属性  $q$  的值;  $g_{j,q}^{\text{best}}$  和  $g_{j,q}^{\text{worst}}$  分别为粒子群所有粒子服务质量历史最优和最差的组合服务的第  $j$  个业务活动对应的服务实例所描述的属性  $q$  的值;  $c_1$  和  $c_2$  为学习因子,是非负常数,决定个体历史记录和全局历史记录对新速度的影响程度;  $r_1$  和  $r_2$  为区间  $[0,1]$  上均匀分布的随机变量;  $\omega$  为惯性权重;  $v_{\text{max}}$  为粒子飞行的最大速度。当粒子群的多样性处在最高阈值与最低阈值之间时,粒子速度的更新方式与上次迭代保持一致。

$t+1$  代粒子所描述的组合服务在各个业务活动节点对应服务实例的各个服务质量属性维度

上的值可以通过公式  $y_{i,j,q}^{t+1} = x_{i,j,q}^t + v_{i,j,q}^{t+1}$  计算得到,并在第  $j$  个业务活动服务实例集合中寻找与  $y_{i,j,q}^{t+1}$  最为相似的服务作为  $x_{i,j,q}^{t+1}$ ,记

$$x_{i,j,q}^{t+1} = \left\{ \omega \mid \min \sqrt{\sum_{k=1}^m \omega_{q_k} - y_{i,j,q_k}^{t+1}}^2, \omega \in \text{Cans}_j \right\},$$

式中  $\text{Cans}_j$  为第  $j$  个业务活动节点所有候选服务的集合。

结合文献[11],给出了 MPSO 算法惯性权重自适应的二次震荡模型。当群体相互吸引靠拢时,惯性权重

$$\omega = (t_{\text{max}} - t/t_{\text{max}} - t_{\text{ccurrent}})^2 (\omega_{\text{initial}} - \omega_{\text{final}}) + \omega_{\text{final}},$$

式中:  $t_{\text{max}}$ ,  $t$  和  $t_{\text{ccurrent}}$  分别为算法迭代总次数、当前迭代次数和群体到目前连续收缩运动的迭代次数;  $\omega_{\text{initial}}$  和  $\omega_{\text{final}}$  分别为惯性权重的初始值和上界。

若群体粒子间相互排斥运动,则惯性权重设定为一个常量  $C$ ,使其以一个固定的步伐相互排斥运动,增大群体的多样性。

## 2.3 算法流程

算法主流程的步骤为: a. 初始化粒子群,并初始化标志位 Flag 置为 true; b. 计算每个粒子的  $F(i)$  及  $D(i)$ ; c. 更新每个粒子个体最优以及最差位置; d. 对寄存集 ND 以及 GBCS 和 GWCS 进行处理; e. 计算当前群体的多样性值  $V$  以及惯性权重值  $\omega$ ; f. 若  $V \leq d_{\text{low}}$ ,则转至 g; 若  $V > d_{\text{high}}$ ,则转至 i; 否则转至 k; g. 排斥方式更新粒子每个业务活动各维服务质量速度; h. 将  $t_{\text{ccurrent}}$  置 0, Flag 置为 false,转至 l; i. 以收缩方式更新每个粒子各维速度; j.  $t_{\text{ccurrent}} + 1$  并且 Flag 置为 true,转至 l; k. 若 Flag 为 true,则转到 i, 否则转至 h; l. 更新所有粒子每个业务活动各维服务质量属性值; m. 为粒子的每个业务活动计算最相似的服务实例; n. 判断是否迭代完毕,若没有则转至 b, 否则结束。根据前面定义,则算法的复杂度为  $O(t_{\text{max}} \otimes p \otimes n \otimes k)$ 。

下面给出对寄存集 ND 的处理过程,把每次迭代后服务质量不被其他粒子支配同时不被寄存集中记录支配的组合服务实例添加到寄存集 ND 中,并且移除寄存集中被待加入的实例支配的记录。服务质量全局最优及最差粒子寄存集的处理流程类似 ND 的处理过程,同时增加了适应度函数值  $F(i)$  的比较过程。主要步骤如下: a. 从群体中选取粒子  $i$ , 若  $D(i)=0$  则转至 b, 否则转至 g; b. 若寄存集 ND 为空,转至 f, 否则转至 c; c. 从 ND 中选取记录 cs, 若 cs Pareto 支配粒子  $i$ , 则转

至 g, 若粒子  $i$  pareto 支配 cs, 则转至 d; 否则转至 e; d. 将 cs 从 ND 中移除; e. 判断 ND 中的记录是否比较完毕, 若没有则转至 c, 否则转 f; f. 将粒子  $i$  加入 ND; g. 若群体中的粒子选取完毕则结束, 否则转至 a.

### 3 实验设计与结果分析

实验环境为 Intel(R) G620 CPU 2.7 GHz 处理器, 3.4 GB 内存, 在 Windows XP 平台下用 Matlab 进行仿真实验, 采用顺序结构的半自动 Web 服务组合抽象逻辑模型. 最大迭代次数为 2 500, 粒子群大小为 20,  $c_1=c_2=0.5$ ,  $d_{\text{high}}=0.4$ ,  $d_{\text{low}}=0.3$ ,  $v_{\text{max}}=0.2$ ,  $t_{\text{initial}}=1$ ,  $t_{\text{final}}=0.9$ ,  $C=0.95$ ,  $\sigma_s=1$ .

根据算法对比的角度, 随机生成多组原始数据集, 并对数据进行归一化处理<sup>[5]</sup>. 对于归一化处理后的数据集 data  $M\_N\_P$ , 它描述 Web 服务组合逻辑模型包含  $M$  个业务活动节点, 每个业务活动节点对应的服务实例集合包含  $N$  个 Web 服务, 每个服务实例包含  $P$  个维度的服务质量属

性.

利用有效率  $E_r$ 、精确度  $A_r$  以及平衡度  $E_r - A_r$  三类指标对算法的性能进行评价. 记

$$E_r = \text{size}(N_D) / \text{size}(V_{\text{Set}});$$

$$A_r = \text{size}(A_{\text{Set}}) / \text{size}(V_{\text{Set}}),$$

式中:  $\text{size}(\cdot)$  为选取集合大小;  $A_{\text{Set}} = \{\omega | \omega \in N_D \wedge \omega \in V_{\text{Set}}\}$ ;  $V_{\text{Set}}$  为对特定数据集通过穷举法计算得到的服务质量处在 Pareto 前端的所有组合服务的集合. 通过  $E_r$  描述算法发现可供使用的组合服务的有效率, 用  $A_r$  描述算法发现组合服务的精确度,  $E_r$  和  $A_r$  反映算法平衡全局搜索与局部搜索能力的强弱, 其值越小表示性能越高.

图 1 给出了 data6\_10\_5, data5\_12\_5 和 data4\_20\_5 通过 MPSO 以及 SPSO 算法<sup>[12-13]</sup> 发现的服务质量不被支配的组合服务的有效率( $E_r$ )和精确度( $A_r$ ), 每 50 代记录一次数据. 图中  $t$  为代数. 可以发现: 面向 Web 服务组合推优的 MPSO 算法在发现服务质量处于 Pareto 前端的组合服务问题上, 在有效性及精确度方面明显优于 SPSO 算法的性能, 并且随着迭代次数的增大, 这种优势将更加明显.

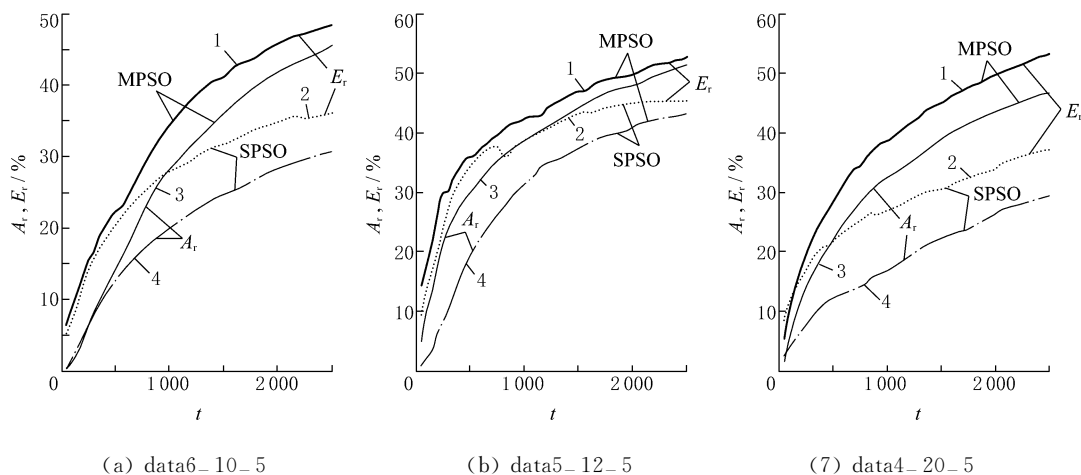


图 1 MPSO 与 SPSO 有效率及精确度对比

比较图 1 中 MPSO 与 SPSO 得到的  $E_r$  与  $A_r$  的差值, 可以明显发现 MPSO 算法在平衡度方面比 SPSO 算法表现出更加优良的性能.

为了研究 Web 服务组合模型中业务活动节点个数、每个业务活动对应的服务实例集合大小以及服务质量的维度对 MPSO 算法的影响, 控制其他参数不变的情况下变换待考察参数的值进行了三组实验, 如图 2 所示.

图 2(a) 表明随着业务活动对应的服务实例集合的增大, MPSO 算法发现服务质量处于 Pareto 前端的组合服务的  $E_r$  及  $A_r$  都会相应增大;

图 2(b) 表明增加业务节点个数, 降低了算法发现服务质量处于 Pareto 前端组合服务的  $E_r$  及  $A_r$ ; 图 2(c) 反映出增大服务质量的维度, 提高了 MPSO 算法发现服务质量处于 Pareto 前端组合服务的  $E_r$  和  $A_r$ .

通过图 2 中  $E_r$  与  $A_r$  的差值可以发现随着业务活动对应的服务实例集合的增大以及服务质量的属性维度的增大, 面向 Web 服务组合的 MPSO 算法平衡全局搜索与局部搜索的能力增大; 增加业务活动节点个数对算法平衡能力影响比较小.

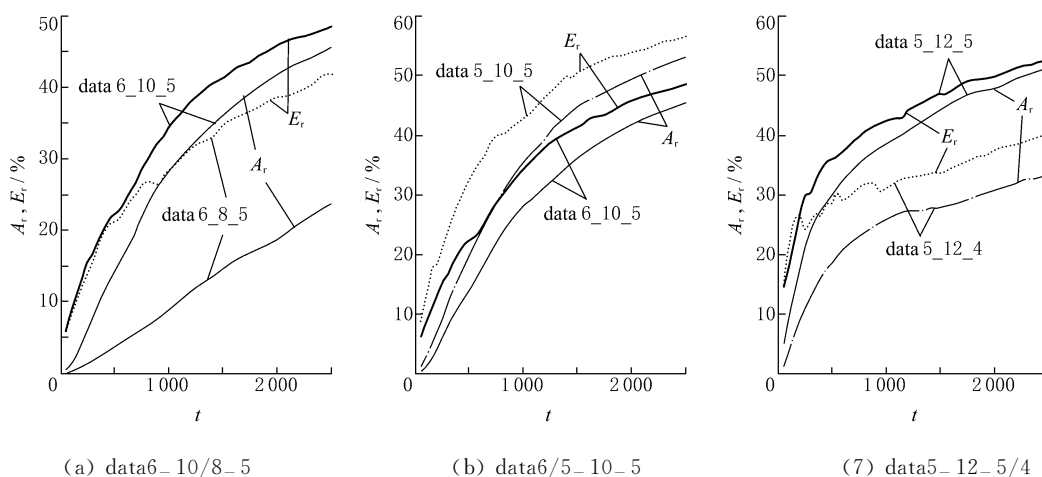


图 2 组合服务相关参数对 MPSO 算法影响

## 参 考 文 献

- [1] Georgakopoulos D, Papazoglou M P. Service-oriented computing[M]. Boston: MIT Press, 2008.
- [2] 岳昆,王晓玲,周傲英. Web 服务核心支撑技术: 研究综述[J]. 软件学报, 2004, 15(3): 428-442.
- [3] 范小芹,蒋昌俊,方贤文,等. 基于离散微粒群算法的动态 Web 服务选择[J]. 计算机研究与发展, 2010, 47(1): 147-156.
- [4] 刘书雷,刘云翔,张帆,等. 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.
- [5] Majithia S, Walker D W, Gray W A. A framework for automated service composition in service-oriented architectures in the semantic web: research and applications[M]. Berlin: Springer, 2004.
- [6] 温涛,盛国军,郭权,等. 基于改进粒子群算法的 Web 服务组合[J]. 计算机学报, 2013, 36(5): 1031-1046.
- [7] 郑金华,蒋浩,邝达,等. 用擂台赛法则构造多目标 Pareto 最优解集的方法[J]. 软件学报, 2007, 18(6): 1287-1297.
- [8] 郭文忠,陈国龙. 离散粒子群优化算法及其应用[M]. 北京: 清华大学出版社, 2012.
- [9] Pant M, Radha T, Singh V P. A simple diversity guided particle swarm optimization[C]// IEEE Congress on Evolutionary Computation. Washington: IEEE, 2007: 3294-3299.
- [10] 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.
- [11] Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers and Operations Research, 2006, 33(3): 859-871.
- [12] 夏虹,李增智. 粒子群算法求解 Web 服务组合中基于 QoS 的服务选择[J]. 北京邮电大学学报, 2009, 32(4): 63-67.
- [13] 刘莉平,陈志刚,刘爱心. 基于粒子群算法的 Web 服务组合研究[J]. 计算机工程, 2008, 34(5): 104-106.