

文章编号:1001-9081(2008)08-2170-03

基于位置矩阵 QoS 感知的 Web 服务组合研究

龚小勇^{1,2}, 朱庆生¹, 武春岭²

(1. 重庆大学 计算机学院, 重庆 400044; 2. 重庆电子工程职业学院 计算机应用系, 重庆 401331)

(g_x_y_h_m@yahoo.com.cn)

摘要:提出了一种基于位置矩阵 QoS 感知的 Web 服务组合方法。该方法使用遗传算法用位置矩阵对基因进行编码,使得该编码方式可以表示服务组合的所有组合路径和重计划信息,算法的一次执行就能完成所有路径 QoS 最优的全局搜索和动态重计划功能。实验结果证明了该方法的有效性和可行性。

关键词:服务质量; Web 服务组合; 位置矩阵; 遗传算法

中图分类号: TP311 **文献标志码:** A

Research on QoS-aware Web services composition based on position matrix

GONG Xiao-yong^{1,2}, ZHU Qing-sheng¹, WU Chun-lin²

(1. College of Computer Science, Chongqing University, Chongqing 400044, China;

2. Department of Computer Applications, Chongqing College of Electronic Engineering, Chongqing 401331, China)

Abstract: A method of QoS-aware Web services composition based on Position Matrix was presented. The method using genetic algorithm includes a position matrix coding scheme that can express simultaneously all of the composite paths and replanning information. Only one time running of the proposed genetic algorithm can construct the optimal executive plan and implement the dynamic replanning from a great deal of services compositions with different QoS. The experimental results indicate the feasibility and efficiency of this method.

Key words: QoS; Web services composition; position matrix; genetic algorithm

0 引言

Web 服务作为一种新型分布式计算模型,近年来得到了工业界和学术界的极大关注^[1]。随着 Web 服务数量的增多,出现了许多服务提供者提供的服务具有相同的功能不同的服务质量(QoS)。因此,在服务组合过程中,需要根据用户的 QoS 要求对 Web 服务进行选择。

支持 QoS 的服务组合研究方法主要有基于 QoS 语义和基于 QoS 属性计算两种方法。采用 QoS 属性计算可以很好地满足用户对服务组合的全局限制,穷尽计算法、线性规划法、遗传算法是几种主要的 QoS 属性计算方法^[2-4]。穷尽计算法存在扩展性差、计算量相对较大的弊端。线性规划法要求目标函数和约束条件是线性的,必须将非线性的 QoS 属性和约束条件的计算公式进行转换,这在一定程度上限制了算法的实用性^[5]。遗传算法作为一种智能优化方法,具有并行计算、群体寻优的特点,不需要与应用背景相关的启发式知识,只需要目标函数和相应适应值函数^[6,7],它已广泛应用于各种优化问题的求解。本文采用遗传算法来实现 QoS 感知的 Web 服务组合。

目前,Web 服务组合中的遗传算法大都采用简单染色体编码方式,即编码中只包含在服务组合时选取的 Web 服务在候选服务集中的编号,一次只能实现一条路径的服务组合选取,不能很好地满足具有多路径特征的组合需要,也无法采用简单的方式表示服务组合的多种类型以及重计划等信息。

本文针对这些问题,提出了一种基于位置矩阵的编码方

式,即用位置矩阵代替染色体参与遗传操作,解决了服务组合中多个路径同时表示的问题,可以简单、有效地表示服务组合的多种类型以及动态重计划信息,算法的一次执行就能完成所有路径 QoS 最优的全局搜索和动态重计划功能。

1 服务组合结构

Web 服务组合的功能可以分成多个子功能,本文称这些子功能为任务(task,简称 t,是构成服务组合的基本逻辑单位,仅包含功能描述和接口信息,不指向具体的 Web 服务)。为了能有效地说明服务组合中的问题,本文引入初始虚拟任务和终止虚拟任务(两个逻辑上存在、物理上不存在的任务)来标志一个服务组合逻辑上的开始和结束,这里分别用 t_0 和 t_{n+1} 来表示。根据任务间的执行逻辑关系,本文使用状态图^[8]将服务组合划分为顺序结构、并行结构、分支结构、循环结构,其他大部分服务组合都可以由这 4 种基本结构组合而成。

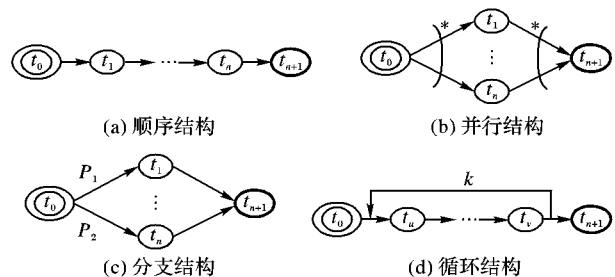


图 1 服务组合的 4 种基本结构

下面是一个 Web 服务组合的例子,该服务组合由任务集

收稿日期:2008-02-25; 修回日期:2008-03-27。 基金项目:国家科技支撑计划项目(2007BAH08B04); 国家 863 计划项目(2006AA102233); 重庆自然科学基金资助项目(CSTC2004BB2192)。

作者简介:龚小勇(1970-),男,重庆人,副教授,博士研究生,CCF 高级会员,主要研究方向:Web 服务、电子商务、分布式计算; 朱庆生(1958-),男,重庆人,教授,博士生导师,主要研究方向:软件工程、电子商务、图像处理; 武春岭(1975-),男,河南郑州人,讲师,硕士,主要研究方向:面向服务的计算、Web 服务组合。

合 $\{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ 组成,其结构如图 2 所示。

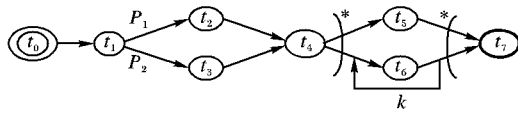


图 2 服务组合结构

由图 2 可知,完成从初始任务到终止任务的组合功能,会存在多条组合路径(Composite Path, CP, 一条从起始任务到终止任务的组合方案),这些路径在任务组成数量、组合顺序上是不同的。如: $t_0, t_1, t_2, t_4, t_5, t_6, t_7$ 和 $t_0, t_1, t_3, t_4, t_5, t_6, t_7$ 是两条不同的组合路径,如图 3 所示。

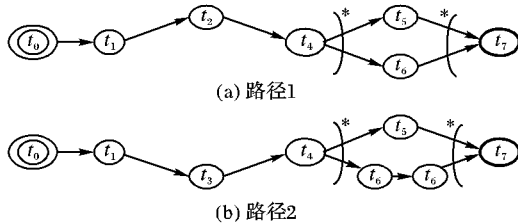


图 3 组合路径实例

在服务组合过程中,每个任务会有多个候选服务(Services, 以下简称 S, 用来完成逻辑任务各项操作功能的具体 Web 服务)与之对应,这些服务由不同提供者提供,具有相同调用接口、相同功能和不同 QoS 属性,因此,每条组合路径又包含多个执行计划(Executive Plan, EP, 从组合路径每个任务的候选服务中选取的,能完成组合路径功能的一组服务组合方案)组成,执行计划的 QoS 属性可以通过上述 4 种基本结构的 QoS 属性来获取。

2 位置矩阵及基因编码方式

2.1 位置矩阵的定义

为了能准确描述服务组合的基本结构,以便计算机快速处理,考虑到任务之间的组合特性和位置关系,本文采用图论中邻接矩阵的思想,设计用位置矩阵来表示任务之间的相互关系。首先讨论任务间的位置关系,对于任意两个任务 t_i 和 t_j ,它们在不同的组合结构中具有不同的位置关系。

1) 顺序结构,只有一种位置关系,图 4(a) 表示任务 t_i 是任务 t_j 的唯一直接前趋。

2) 并行结构,有两种位置关系,图 4(b) 表示任务 t_i 有一组并发直接后继任务,且任务 t_j 是其中的一个,图 4(c) 表示任务 t_j 有一组并发直接前趋任务,且任务 t_i 是其中的一个。

3) 分支结构,有两种位置关系,图 4(d) 表示任务 t_i 有多个可选择的直接后继,且选择任务 t_j 的概率为 p_j ,图 4(e) 表示任务 t_j 有多个可选择的直接前趋任务,且选择任务 t_i 的概率为 p_i 。在实际应用中,可通过任务 t_i 和 t_j 之间的运行日志找出其具体执行关系,通过求平均值来确定分支概率。

4) 循环结构,由于循环结构可通过日志确定循环时的平均执行次数 k ,可将循环结构转变成多个顺序结构。为了抓住服务组合的关键问题,本文默认服务组合中循环结构的平均执行次数为 1 次。

对于示任务之间的位置关系,可用位置矩阵 D 来表示,其元素 d_{ij} 的取值定义如下。

1) 当 $i = j$ 时,表示主对角线上的元素,对应组合结构图中的所有任务,取值含义为:

初值“0”表示本次执行路径中未包含该任务。

服务组合时更新“正整数”表示本次组合路径中用来

执行该任务的候选服务在候选服务集中的编号。

执行计划运行时更新“-1”表示该任务已执行完毕;“-2”表示发生执行异常时,该任务正处于执行状态;“-3”表示发生执行异常时,该任务处于待执行状态。

2) 当 $i \neq j$ 时,表示任务 t_i 与任务 t_j 在组合结构图中的位置关系,在组合结构确定后,取值不再发生改变。

“0”表示任务 t_i 在组合结构图中不是任务 t_j 的直接前趋;

“1”表示任务 t_i 与任务 t_j 具有图 4(a) 的位置关系;

“2”表示任务 t_i 与任务 t_j 具有图 4(b) 的位置关系;

“4”表示任务 t_i 与任务 t_j 具有图 4(c) 的位置关系;

“ $8 + p_j$ ”表示任务 t_i 与任务 t_j 具有图 4(d) 的位置关系;

“ $16 + p_i$ ”表示任务 t_i 与任务 t_j 具有图 4(e) 的位置关系。

对应图 2 的组合结构,其位置矩阵的初始状态如 D^0 所示(假设 $k = 1, p_1 = p_2 = 0.5$):

$$D^0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.5 & 8.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

位置矩阵既能够描述任务之间的静态位置关系,又能够描述服务组合、服务执行和服务重计划的动态情况,是支持 QoS 感知的 Web 服务组合的重要分析工具。

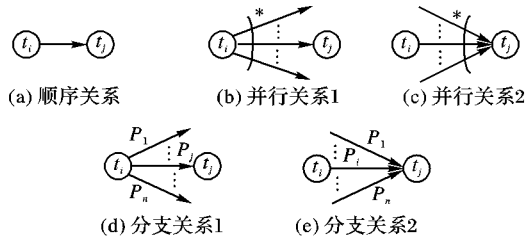


图 4 组合结构图中任务间的位置关系

2.2 基因编码

本文把位置矩阵 D 中的对角线元素 $d_{ij}(i = j)$ 定义为基因座,其个数与服务组合结构图中的任务数相等,所有基因座构成了染色体的基本结构,每个基因座包含一个和对应任务相匹配的候选服务集,候选服务集的每个服务再指向表示 QoS 属性的数组。对染色体采用整数定长编码的方式,染色体中第 1 个(最后)基因总是服务组合的起点(终点),染色体中间的每一个基因对应一个具体服务在候选服务集中的编号。本文采用位置矩阵参与遗传操作,位置矩阵的主对角线元素表示染色体,其他元素表示任务间的位置关系。

3 QoS 感知的 Web 服务组合过程

QoS 感知的 Web 服务组合过程包括两个阶段。第一阶段,从 QoS 全局最优的角度出发,搜索所有执行计划中满足约束条件的一组非劣解;第二阶段,在执行计划运行时,完成服务组合的动态重计划。

3.1 最优执行计划集的搜索

主要过程是:将每一个执行计划编码为染色体,通过染色体之间的交叉、变异等重组操作,产生具有更高目标函数值的新染色体,这一遗传过程不断进行,实现在解空间的并行全局搜索。搜索停止时,得到一个染色体集合,即满足约束条件的

优化或近似优化的执行计划集。

3.1.1 适应度函数设计

适应度评价是遗传操作的依据,其中目标函数的设计直接影响到遗传算法的性能。假设 Web 服务包括 4 种 QoS 属性,即执行时间 T (time)、执行费用 C (cost)、可靠性 R (reliability) 和信誉等级 Rep (reputation),由于 T 和 C 属性是减量型的,即其值越低,质量越高,而 R 和 Rep 是增量型的,即其值越高,质量越高,为使支持 QoS 的服务选择遗传算法适应度函数值与服务质量成正比,我们得出计算染色体目标函数 $f(x)$ 的公式为:

$$f(x) = \frac{w_R R_{ep} + w_{Rep} Rep_{ep}}{w_T T_{ep} + w_C C_{ep}} \quad (1)$$

其中, T_{ep} , C_{ep} , R_{ep} , Rep_{ep} 都是经归一化后的 QoS 属性值, w_T , w_C , w_R , w_{Rep} 分别是对应的权值,表示用户对 QoS 属性的关注程度,一般来说,可以通过观察目标函数的应用前景和分析不同因子对性能的影响从而校准加权重值,并且 $w_T + w_C + w_R + w_{Rep} = 1$ 。

3.1.2 交叉和变异的合法性检查

在位置矩阵中,主对角线包含了服务组合结构图中的所有任务,可以产生代表任何路径的染色体(这也是用位置矩阵表示服务组合的最大优势所在),算法获得的是所有组合路径中的最优组合,因此,种群中会存在代表不同路径的染色体。由于服务组合结构图中可能存在分支结构,实际存在的组合路径数量只占主对角线元素可组合数量的很少一部分,再加上遗传算法固有的随机性,造成交叉或变异操作生成的新染色体可能不代表任何现有的组合路径。因此,在进行交叉与变异操作时,需要根据位置矩阵中任务间的位置关系检查新生成染色体的路径合法性。

3.2 动态重计划的实现

主要过程是:通过位置矩阵记录服务组合的执行状态,当执行计划运行失败或者染色体中某个未执行的 Web 服务 QoS 值发生重大变化时,根据位置矩阵中的状态信息进行动态重计划或者从备选的执行计划集中选择另一个执行计划来完成服务组合中未执行的部分。具体步骤如下,其中输入参数包括位置矩阵 D^0 ,由 4.1 得到的执行计划集 P^* ,用户总时间限制 T_0 (从服务选择开始到所有服务执行完毕的时间):

- 1) $P_k^* \leftarrow \text{OptimalSelect}(P^*)$
- 2) $D_k^* \leftarrow \text{Revise}(D^0, P_k^*)$
- 3) For each $d_{ij} \in D_k^* \wedge i = j \wedge i < = n + 1$ Do
- 4) $s \leftarrow \text{Execute}(d_{ij})$
- 5) If $s < 0$ then
- 6) Update(D_k^*, s)
- 7) $D_k^* \leftarrow \text{ReduceOrder}(D_k^*)$
- 8) If $T^{wa} < T^0 - T$ then
- 9) GeneticSearch()
- 10) Else
- 11) $P^* \leftarrow \text{ReduceLenth}(P^* - P_k^*)$
- 12) $P_k^* \leftarrow \text{OptimalSelect}(P^*)$
- 13) $D_k^* \leftarrow \text{Revise}(D_k^*, P_k^*)$
- 14) Endif
- 15) Else
- 16) Update(D_k^*, s)
- 17) Endif
- 18) Endfor

步骤 1) 在执行计划集中选择一个最优解 P_k^* ,并用他来修改初始位置矩阵 D^0 ,得到一个表示最优执行计划的位置矩阵 D_k^* 。步骤 4) 中的 s 表示服务运行状态,当 $s < 0$ 时,表示执行计划在运行到 d_{ij} 时发生了异常。步骤 6) 根据执行计划的运

行状态和位置矩阵编码规则对位置矩阵对角线元素值进行修订。步骤 7) 对位置矩阵 D_k^* 进行降阶处理。步骤 8) - 14) 判断,如果重计划的算法执行时间 T^{wa} 小于用户限定的总执行时间 T_0 和已用执行时间 T 的差,则再次进行最优执行计划集的遗传搜索,否则,对优良解集 P^* 中的所有染色体(除去 P_k^*) 移去已执行的任务,再从中选择一个最优解,继续执行服务组合中的剩余部分。

4 实验及分析

为了更深入地阐述服务组合方法的有效性和可行性,本文通过实验对图 2 的 Web 服务组合情况进行了验证。实验环境为 100 M 局域网,算法运行微机配置为 PentiumIV 2 GHz 处理器,1 GB 内存,操作系统为 Windows 2003 Server,用 C++ 实现。服务的 QoS 属性采用随机方法在一定范围内生成,服务及其 QoS 属性信息在集中式 UDDI 注册中心进行注册。

首先分析服务组合的执行时间,该实验通过随机改变服务的 QoS 属性值和运行状态(如:由可执行变成无效)来触发服务组合的重计划,图 5 分别考虑了候选服务规模为 5、10 和 15,进化代数取 100、200、300 和 400 的情况下,服务组合算法的时间开销。图 6 是在候选服务规模为 10,进化代数取 300 的情况下,重计划的不同次数对算法执行时间的影响。对于每一种情况,算法分别运行 15 次取平均值。

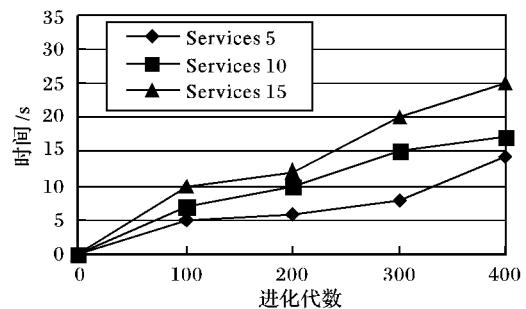


图 5 不同进化代数执行时间

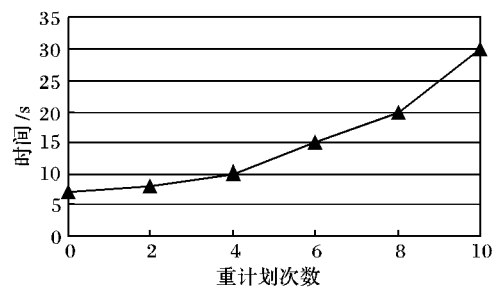


图 6 不同重计划次数执行时间

本文考虑的服务组合包含 8 个任务,这一求解规模一般可以满足大部分服务组合求解的需要。由图 5 可以看出:随着候选服务数目的增加,在不同的进化代数下,CPU 运行时间没有大量增加,在候选服务规模为 10 的情况下,算法迭代 400 次总的执行时间在 17 s 左右。图 6 显示,在服务组合执行期间,重计划达到 10 次时,算法总执行时间在 30 s 左右。

下面讨论基于位置矩阵编码和基于简单染色体编码遗传算法的实验对比。前者定义为 DC,后者定义 SC,图 7 为在候选服务规模为 10 的情况下,DC 与 SC 分别迭代 100、200、300、400 代的运行时间比较,这里的时间为 DC 与 SC 分别运行 15 次取平均。从图 7 可以看出:因为位置矩阵的加入,DC 的时间开销要大于 SC。图 8 为在候选服务规模为 10、在不同进化代数下,DC 与 SC 所得的最优解的目标函数值的分布情况。可以看出,DC 所求得的最优解性能要优于 SC。

(下转第 2176 页)

交易,信任扩散机制都是可信的和逼真的。

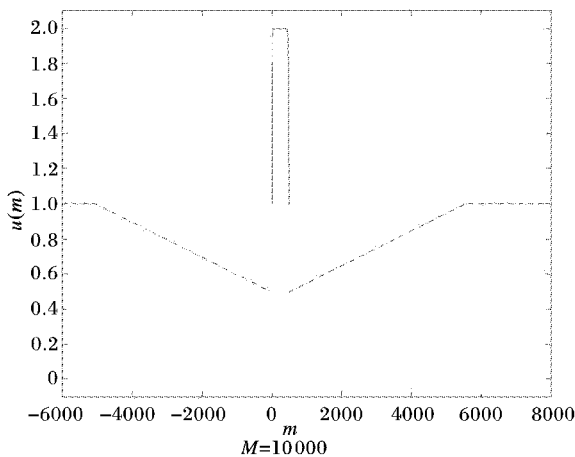


图 7 交易者群体的规模对信任扩散的影响

4 结语

本文建立一种基于 ESS 均衡的电子商务信任动态博弈模型,并据此提出了相应的信任扩散机制与算法、理论推导与仿真实验,显示出新模型能够逼真地分析交易者之间相互信任的建立与形成过程,消除了静态研究的局限性。与此同时,基于仿真实验结果提出了建立或恢复电子商务信任机制的若干对策建议,包括增强信任激励以增加选择信任策略的交易者数量和提高交易者选择信任策略的收益比较优势,增大电子商务交易网站的信息交换技术投入以增强交易信息透明

度等。

参考文献:

- [1] 侯玉凤. 我国 C2C 电子商务诚信问题及其对策思考[J]. 中国管理信息化, 2007, 10(5): 73-76.
- [2] 蒋佳利. 关于我国 C2C 电子商务支付方式及信用建设[J]. 北方经济, 2007(4): 99-100.
- [3] 陆弘彦, 韦欣. C2C 网络市场的信用体系比较研究[EB/OL]. [2006-12-08]. <http://www.paper.edu.cn>.
- [4] 黄黎清, 李彬彬, 李明珠. C2C 电子商务诚信问题与平台设计探讨[J]. 商场现代化, 2007(6): 101-102.
- [5] 吴亮, 文静. C2C 电子商务中的诚信问题研究[J]. 商场现代化, 2007(4): 139.
- [6] 王晓燕. CtoC 电子商务中的信任问题: 一个进化博弈分析模型[J]. 商业研究, 2005(6): 179-181.
- [7] PENG LIFANG, ZHONG CHENG, QI LI. Model and method for evaluating creditability of C2C electronic trade[C]// ICEC'06. New York, NY, USA: ACM Press, 2006: 244-249.
- [8] 蔡涵, 姜锦虎. 我国 C2C 市场中有效第三方机制建立问题的博弈分析[J]. 科技进步与对策, 2007(2): 172-175.
- [9] 刘树成. 现代经济辞典[M]. 南京: 江苏古籍出版社, 2005.
- [10] 毛捷, 金雪军, 袁佳. 证券市场新旧投资理念更替过程研究: 一个基于演化视角的分析[J]. 经济学, 2005, 4(3): 707-738.
- [11] SMITH J M, PRICE G R. The logic of animal conflict[J]. Nature, 1973, 246(5427): 15-18.
- [12] 王平, 邱劲, 邱玉辉. 电子商务环境中一种基于多主体系统的信任模型[J]. 计算机应用, 2007, 27(3): 601-604.

(上接第 2172 页)

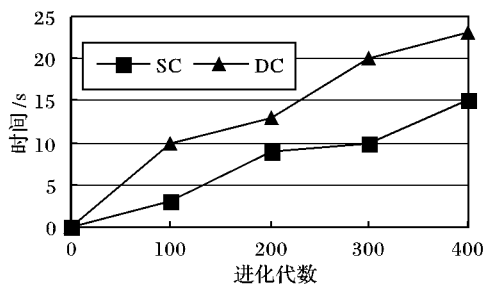


图 7 DC 与 SC 执行时间比较

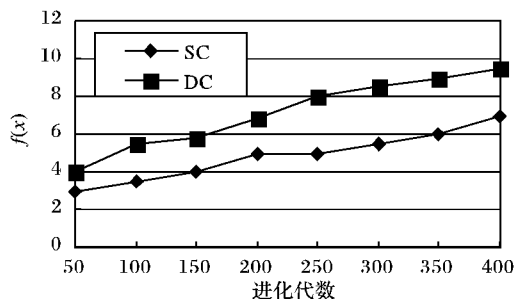


图 8 DC 与 SC 目标函数值比较

5 结语

Web 服务以其特有优势使人们看到其广泛的应用前景,而支持 QoS 的 Web 服务组合的有效解决将为 Web 服务的普及起到有力的推动作用,使 Web 服务由一种技术转化为真正可以为人们使用的工具。本文提出的基于位置矩阵 QoS 感知的 Web 服务组合,采用位置矩阵对基因进行编码,使得该编码方式可以表示服务组合的多种类型和所有组合路径,以

及动态重计划的初始状态和更新信息,有效提高了服务组合的效率。实验结果证明了该服务组合方法的可行性和有效性。

参考文献:

- [1] 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.
- [2] BENATALLAH B, DUMAS M, SHENG Q Z, *et al*. Declarative composition and peer-to-peer provisioning of dynamic Web services [C]// Proceeding of the 18th International Conference on Data Engineering. Washington: IEEE Computer Society, 2002: 297-308.
- [3] LIU YUTU, ANNE H H, ZENG L Z. QoS computation and policing in dynamic Web service selection[C]// Proceeding of the WWW 2004. New York: ACM Press, 2004: 66-73.
- [4] TIAN M, GRAMM A, RITTER H, *et al*. Efficient selection and monitoring of QoS-aware Web services with the WS-QoS framework [C]// Proceeding of the IEEE International Conference on Web Intelligence (WI 2004). Washington: IEEE Computer Society, 2004: 152-158.
- [5] ZENG L Z, 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.
- [6] SAFE M, CARBALLIDO J, PONZONI I, *et al*. On stopping genetic criteria for algorithms[C]// SBIA 2004, LNAI3171. Berlin: Springer, 2004: 405-413.
- [7] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 2001: 128-139.
- [8] 代钰, 杨雷, 张斌, 等. 支持组合服务选取的 QoS 模型及优化求解[J]. 计算机学报, 2006, 29(7): 1167-1178.