软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(7):1601−1614 [doi: 10.13328/j.cnki.jos.004698]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向服务匹配问题的协同演化算法*

崔晓晖[1], 印桂生[2], 董红斌[2]

[1](北京林业大学 信息学院,北京 100083)

[2](哈尔滨工程大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

通讯作者: 印桂生, E-mail: yinguisheng@hrbeu.edu.cn

摘 要: 服务匹配是服务发现的主要环节.目前,原子服务匹配过程主要存在服务匹配概念狭窄、匹配算法的时间复杂度较高及匹配方案的表示难以被智能优化算法处理等问题.针对上述问题,在原子服务匹配的基础上引入复合服务匹配、抽象复合服务匹配过程的适应度函数及约束条件,设计适用于智能优化算法处理的匹配方案的表示方法.同时,结合协同演化算法设计思路,提出基于粒子群和模拟退火的协同演化算法(PSO-SA),用以求解复合服务匹配.实验结果表明:与现有智能优化算法相比,PSO-SA 可在有限迭代次数内获得精度较高的匹配结果,对不同维度的服务匹配问题具有较高的适应性,可用于提高服务发现结果的质量.

关键词: 服务匹配;粒子群优化;模拟退火;协同演化

中图法分类号: TP301

中文引用格式: 崔晓晖,印桂生,董红斌.面向服务匹配问题的协同演化算法.软件学报,2015,26(7):1601−1614. http://www.jos.org.cn/1000-9825/4698.htm

英文引用格式: Cui XH, Yin GS, Dong HB. Co-Evolutionary algorithm for Web service matching. Ruan Jian Xue Bao/Journal of Software, 2015,26(7):1601−1614 (in Chinese). http://www.jos.org.cn/1000-9825/4698.htm

Co-Evolutionary Algorithm for Web Service Matching

CUI Xiao-Hui[1], YIN Gui-Sheng[2], DONG Hong-Bin[2]

[1](School of Information Science and Technology, Beijing Forestry University, Beijng 100083, China)

[2](College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: Service matching is a principal process of Web services discovery. Nowadays, the narrow concept of the atomic Web service matching, the high time complexity of the current matching algorithm and the difficult expression of the Web service matching for the intelligent optimization algorithms become the main problems in Web service matching development. To solve the above problems, this article introduces the concept of the compound service matching by extending the concept of the atomic service matching, and abstracts the mathematical expression of the compound matching problem by the fitness function and restriction. The expression of the solution of the Web service matching for the intelligent optimization algorithm is also proposed. Based on the co-evolutionary idea of particle swarm optimization (PSO) and simulated annealing (SA), the study puts forward a co-evolutionary algorithm (PSO-SA) to the compound Web service matching problem. According to the experimental results, PSO-SA achieves better matching precision than other optimization algorithms within the limit iterations on various dimensional matching problems. Also, PSO-SA shows the adaptive ability to the compound service matching and improves the quality of result of Web services discovery.

Key words: Web services matching; PSO; SA; co-evolution

Web 服务是一种基于 Web 环境的自包含、自描述、模块化且具有良好互操作能力的分布式软件结构,可以通过网络发布、定位和调用[1].近年来,大量与 Web 服务相关的技术成果不断问世,如网格计算、服务计算、

* 基金项目: 中央高校基本科研业务费专项资金(BLX2014-27); 国家自然科学基金(60973075, 61272186); 黑龙江省自然科学基金(F200937, F201110)

收稿时间: 2013-01-09; 定稿时间: 2014-07-09

SOA、网构软件[2]等,这些成果一方面推动了 Web 服务的研究进程,一方面使得具有类似功能的服务数量呈爆炸式增长.如何获取满足用户请求的候选服务,成为 Web 服务发现的研究内容.

现有服务发现的过程以语义本体技术为基础,利用匹配程度表示服务请求与服务库中服务之间的紧密程度.在服务匹配过程中,由于服务与请求间存在多种匹配方案,如何在有限时间内找到匹配精度较高的方案,是提高服务发现质量的主要手段.但是,当前服务匹配存在的主要问题包括:

(1) 服务匹配概念的狭窄性.现有服务匹配假设单个服务满足服务请求,即,原子服务匹配过程.随着业务认知深度的加强,单一 Web 服务很难满足人们日益复杂的软件业务表达的需求,需拓展服务匹配的概念,实现一个请求由多个服务协同完成,即,复合服务匹配过程;

(2) 现有服务匹配求解算法的时间复杂度较高.目前,匹配算法多是基于经典的优化策略,如整数规划算法[3]、贪心算法[4]或二分图算法[5]等,这些算法在低维度的服务匹配问题下能够获得最优的结果;但是随着复合服务匹配概念的引入,匹配问题规模的扩大导致匹配结果呈爆炸型增长,现有匹配算法为获得最优的结果必须消耗大量时间,表现出较差的在线响应能力和匹配问题的适应能力,处理效率难以被用户接受;

(3) 在其他技术领域中,智能优化算法是处理复杂问题的主要技术手段,但是由于匹配方案的直观表示是一组满足约束的续偶对,而智能优化算法通常处理的求解方案是简单的序列形式,两者在表示上呈现出的差异,导致现有智能优化算法难以直接求解服务匹配问题.

基于服务匹配存在的问题,本文首先通过形式化手段对原子服务匹配进行命题,并以此为基础扩展服务匹配的概念,给出复合服务匹配命题以及服务匹配问题的数学表示,该数学表示可作为求解匹配问题的目标函数及约束条件;其次,设计匹配方案的表示方法,将具有续偶形态的匹配方案等价转换为智能优化算法所能够处理的一般序列形式;最后,采用协同演化的思路,利用 PSO(PSO particle swarm optimization)以及 SA(SA simulation annealing)变异策略的互补特性,设计能够在有限迭代次数内获得较优服务匹配方案的 PSO-SA 协同演化算法.

本文第 1 节介绍相关工作.第 2 节介绍服务匹配命题并抽象匹配问题的数学表达.第 3 节设计匹配方案的表示方法.第 4 节介绍 PSO-SA 算法.第 5 节通过实验选择合理参数配置,对比分析现有算法及 PSO-SA 的收敛精度、收敛情况以及在线响应时间.最后,总结并介绍今后研究方向.

1 相关工作分析

服务匹配是实现高质量服务发现的重要步骤,目前,国内外服务匹配方面的研究主要分为以下两类:服务匹配的语义相似度模型及服务匹配的求解算法等.

- 通过语义关系对服务匹配过程进行建模.

Paolucci 等人[6]提出了一种基于语义匹配的 Web 服务发现方法,通过语义描述 Web 服务的输入/输出参数,实现完全匹配、插入匹配、失败匹配和包含匹配这 4 种主要关系.较传统的基于关键字匹配的 UDDI(universal description discovery and integration)[7]方法,Paolucci 等人的方法能够以更加柔性的匹配过程发现更多满足用户请求的可用 Web 服务,但由于语义关系的复杂性,在实际中具有匹配不精细等问题.邝砾等人[1]针对语义本体相似度计算速度慢及服务发现问题单一性等问题,提出了基于倒排索引优化的语义服务发现机制,该机制对服务库中的候选服务参数进行标注建树,从而快速、准确、高效地发现目标服务.邱田等人[8]将 Web 服务匹配过程看作是描述服务不同属性间的细粒度匹配过程.欧伟杰等人[9]在前人方法的基础上,引入概念松弛方法,重新定义概念间的泛化和特化关系,降低服务查询过程中的匹配参数数量,提高服务查询效率.

- 针对服务匹配求解提出了不同的解决方案.

Zeng 等人[3]从 QoS(quality of service)角度重新对服务的输入/输出参数进行定义,提出了局部优化和全局优化算法.裘江南等人[4]在输入/输出本体的语义相似度矩阵上,利用贪心算法,选取匹配关系矩阵中每一行的最大匹配组成匹配方案,实现快速匹配,但其结果并不是最优的.邓水光等人[5]提出了基于二分图匹配的语义 Web 服务发现方法,该方法不受限于具体的 Web 服务模型和表达语言,通过对二分图最优匹配进行扩展,将求解服务

匹配问题转换成扩展二分图的最佳匹配.张佩云等人[10]采用遗传算法的方法将服务发现应用于已有任务流程不同阶段的服务选择上,其本质是一种与供应链问题类似的粗粒度匹配.

现有研究成果表明,采用语义来描述 Web 服务已成为解决服务匹配问题的主要手段.但现有匹配算法均是针对单一服务发现而设计的,难以利用现有服务库中的服务形成具有协同关系的复合服务来满足日益复杂的用户需求,存在匹配概念狭窄的问题.同时,二分图和整数规划方法均是一种全局搜索策略,当匹配问题规模 n 上升时,获取最优匹配方案的计算复杂度为 $O(n^3)$,运行效率较低,匹配速度严重影响服务发现的在线响应能力.优化算法是解决复杂问题的主要手段,但是由于匹配方案的自然表示较难被智能优化算法所处理,制约了优化算法在服务匹配方面的应用.

2 服务匹配问题

2.1 原子服务匹配

原子服务匹配的目标是发现能够满足用户请求的单个服务,目前,大量的研究均是基于原子服务匹配展开的.本节在给出必要定义的基础上,对原子服务匹配方案进行命题.

定义 1(Web 服务). 一个 Web 服务 w 由一个二元组组成.$w:=\langle n^s,F\rangle$,其中,n^s 表示该服务的名称及服务的功能概述;F 表示该服务所实现的所有操作集合,f_i 为 F 的元素,表示具体的服务操作.

定义 2(服务操作). 一个服务操作 f 由一个三元组组成,$f:=\langle f^s,I,O\rangle$,其中,$f^s$ 表示操作名称及操作功能概述,$I=\{i_1,i_2,\ldots,i_n\}$描述 f 的输入本体参数集合,$O=\{o_1,o_2,\ldots,o_m\}$描述 f 的输出本体参数集合.

定义 3(用户请求). 用户请求 R 由一个三元组组成.$R:=\langle r^s,I^r,O^r\rangle$.其中,$r^s$ 用于标识本次请求,$I^r=\{i_1^r,i_2^r,...,i_k^r\}$描述 R 所能提供的输入本体参数集合,$O^r=\{o_1^r,o_2^r,...,o_l^r\}$ 描述 R 所期望得到的输出本体参数集合.

命题 1(原子服务匹配方案). 对于服务请求 R 和$\forall w$,如果$\exists f\in w.F$,使得建立在 I 到 I^r 的关系 I_f和 O^r 到 O 上的关系 O_f满足需求(1)和需求(2),则称$\langle I_f,O_f\rangle$为满足 R 的原子服务匹配方案.需求(1)和需求(2)为

(1) $\forall i_i,i_j\in I,i_i\neq i_j,\exists i_k^r,i_l^r\in I_r,\ i_k^r\neq i_l^r,$ s.t. $I_f(i_i)=i_k^r,I_f(i_j)=i_l^r$;

(2) $\forall o_i^r,o_j^r\in O^r,o_i^r\neq o_j^r,\exists o_k,o_l\in O,o_k\neq o_l,$ s.t. $O_f(o_i^r)=o_k,O_f(o_j^r)=o_l$.

命题 1 表明,原子服务匹配方案由建立在 $f.I$ 到 $R.I^r$ 上的本体参数映射关系 I_f和从 $R.O^r$ 到 $f.O$ 上的本体参数映射关系 O_f组成.在服务匹配过程中,存在多个满足命题 1 的匹配方案.如果采用本体相似度ε来表示不同本体参数间匹配程度,则可获得表示匹配方案的整体匹配程度 z.原子服务匹配问题的目标是从所有满足命题 1 匹配方案中,寻找能够最大化 z 的匹配方案作为最优匹配结果.

2.2 复合服务匹配

随着技术的发展,服务库中必然存在大量可用服务,现有原子服务匹配过程难以充分利用多个服务协同实现日益复杂的用户需求.为提高服务的利用率以及实现对复杂用户请求的匹配,本节在原子服务匹配定义的基础上,引入复合服务匹配的概念,对复合服务匹配方案进行命题.

定义 4(复合服务). 复合服务 cw 由一个二元组组成,$cw:=\langle n^{cw},F^c\rangle$,其中,$n^{cw}$ 为标识该复合服务的字符串.$F^c=\{f_1^c,f_2^c,...,f_n^c\}$ 为该 cw 所容纳的所有操作.对于 F^c 中任意操作 f_i^c,存在服务库中某一服务 w,使得 $f_1^c\in w.F$.

命题 2(复合服务匹配方案). 对于服务请求 R 和$\forall cw$,如果 $\exists F^x\subseteq 2^{F^c}$,使得建立在 $F^x.I^c$ 到 $R.I^r$ 的关系 I_{fc}和建立在 $R.O^r$ 到 $F^x.O^c$ 的关系 O_{fc} 为本体参数映射关系,则称$\langle I_{fc},O_{fc}\rangle$为满足 R 的复合服务匹配方案.

在命题 2 中,I^c 和 O^c 分别是 $cw.F^c$ 中输入和输出本体参数的并集,如公式(1)和公式(2)所示.

$$I^c=\{f_i^c.I\mid f_i^c\in F^c,i\in 1,2,..,n'\}=\bigcup_{i=1}^{n'}F^c.f_i^c.I \tag{1}$$

$$O^c=\{f_j^c.O\mid f_j^c\in F^c,j\in 1,2,..,m'\}=\bigcup_{j=1}^{m'}F^c.f_j^c.O \tag{2}$$

命题 2 表明:复合服务匹配并不要求在服务库中存在单个满足请求的服务,而是认为现有服务库中大量的遗留服务,可组合成一个逻辑上的复合服务 cw 来协同实现复杂的服务请求任务.复合服务的引入,能够在现有服务的基础上最大化地满足复杂请求的多参数需求,增加匹配成功的概率;但与原子服务匹配的方案数量相比,复合服务匹配方案数量将存在更多满足命题 2 的匹配方案.如何在众多匹配方案中高效地获取能够最大化匹配程度 z 的匹配方案,是求解复合服务匹配的关键.

在某些实际服务发现过程中,存在 $R'\subset R.O^r$ 依赖于 $I'\subset R.I^r$ 的情况,即,用户提供的请求本体集合的子集 R' 部分依赖于用户所需的本体参数集合的子集 I'.这类情况的处理方法是:首先将具有依赖关系的子集 R' 和 I' 从 $R.O^r$ 和 $R.I^r$ 中取出,采取命题 1 的原子服务匹配过程获取所需的服务;对于集合中非依赖关系的本体 $\{R.O^r-R'\}$ 和 $\{R.I^r-I'\}$,仍然采取命题 2,通过复合服务匹配过程获取所需的服务.最终结果为原子服务匹配和复合服务匹配的并集.

2.3 服务匹配问题的数学表达

在命题 2 的基础上,将复合服务匹配问题转化为优化问题,获得问题的适应度函数以及约束条件.如果令 $I^*=\{\varepsilon_{ij}\}_{n'\times n}$ 表示 I^c 与 I^r 中输入本体参数的语义相似度矩阵,$O^*=\{\pi_{ij}\}_{m\times m'}$ 表示 O^r 与 O^c 的输出本体参数的语义相似度矩阵,则满足命题 2 的服务匹配问题可表达为公式(4)约束下,最大化公式(3)所示的目标函数.

$$\max z=\sum_{i=1}^{n}\sum_{j=1}^{n'}\alpha_{ij}\varepsilon_{ij}+\sum_{i=1}^{m'}\sum_{j=1}^{m}\beta_{ij}\pi_{ij} \tag{3}$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{n}\alpha_{ij}=1, \quad j=\{j\mid j\in Z^+, j\in[1,n']\}\\ \sum_{j=1}^{n'}\alpha_{ij}=1, \quad i=\{i\mid i\in Z^+, i\in[1,n]\}\\ \sum_{i=1}^{m'}\beta_{ij}=1, \quad j=\{j\mid j\in Z^+, j\in[1,m]\}\\ \sum_{j=1}^{m}\beta_{ij}=1, \quad i=\{i\mid i\in Z^+, i\in[1,m']\}\\ \sum_{i=1}^{n'}\sum_{j=1}^{n}\alpha_{ij}+\sum_{i=1}^{m'}\sum_{j=1}^{m}\beta_{ij}=n'+m'\\ \alpha_{ij},\beta_{ij}\in\{0,1\}\end{cases} \tag{4}$$

公式(3)和公式(4)表明:求解服务匹配旨在输入和输出相似度矩阵中找到满足命题 2 的映射关系 I_{fc} 和 O_{fc},且能够最大化目标函数 z 的匹配方案.由公式(3)及公式(4),I_{fc} 和 O_{fc} 均可表示为在 I^* 和 O^* 上寻找从原象集 $(F^x.I^c,R.O^r)$ 到目标象集 $(R.I^r,F^x.O^c)$ 的映射关系,且在目标函数的表示上及对应参数的约束上具有等价的特性.因此,为避免后续内容描述的冗余,后续章节以 I_{fc} 为例,对于 O_{fc} 的求解过程可对照 I_{fc} 的求解过程.

3 服务匹配方案的转换方法

由命题 2,匹配方案 I_{fc} 可理解为使矩阵 $\{\alpha_{ij}\}$ 中某一行某一列为 1 的所有下标 (i,j) 的序偶集合.每个序偶的第 1 个元素来源于匹配关系原象集元素下标,第 2 个元素来源于匹配关系的目标象集的元素下标.虽然序偶表示方法直观地体现了匹配过程,但其形式过于复杂,若要应用优化算法进行求解,必须进行转化.因此,通过 3 种等价转换——方阵转化、排列转化和顺序转换,将 I_{fc} 从序偶形式转换成智能算法所能处理的简单序列形式 I_{fcr}.转换过程如图 1 所示.

$$I_{fc}\xleftrightarrow{\text{方阵转换}}I_{fc'}\xleftrightarrow{\text{排列转换}}I_{fca}\xleftrightarrow{\text{顺序转换}}I_{fcr}$$

Fig.1 Transformations of the services matching solutions for the optimization algorithm

图 1 适用于优化算法的匹配方案的转换过程

3.1 方阵转换

在矩阵 $I^*_{n'\times n}$ 中,可能存在用户给定参数冗余的情况,即,$n\geqslant n'$.为便于匹配算法处理,在 $I^*_{n'\times n}$ 矩阵中添加$|n-n'|$行以$\varepsilon=0$为元素的行向量,使矩阵 $I^*_{n'\times n}$ 转换为方阵 $I^*_{n\times n}$.定理 1 保证了方阵转换的等价性.

定理 1. 在服务匹配问题中, $I^*_{n\times n}$ 为 $I^*_{n'\times n}$ 经过方阵转换的语义相似度矩阵,如果令 z'和 z 分别为 $I^*_{n\times n}$ 和 $I^*_{n'\times n}$ 上的适应度,则该转换过程不改变匹配方案的适应度,即 $z'=z$.

证明:在方阵 $I^*_{n\times n}$ 上的适应度函数 z'如公式(5)所示.

$$z'=\sum_{i=1}^{n}\sum_{j}^{n}\alpha_{ij}\varepsilon_{ij}=\sum_{i=1}^{n'}\sum_{j}^{n'}\alpha_{ij}\varepsilon_{ij}+\sum_{i=n'+1}^{n'}\sum_{j}^{n}\alpha_{ij}\varepsilon_{ij} \tag{5}$$

根据转换过程, $\sum_{i=n'+1}^{n'}\sum_{j}^{n}\alpha_{ij}\varepsilon_{ij}=0$,即 $z'=\sum_{i=1}^{n'}\sum_{j}^{n'}\alpha_{ij}\varepsilon_{ij}=z$.证毕. □

3.2 排列转换

根据公式(3)和公式(4),$I_{fc'}$中每一个ε_{ij},都存在唯一的序偶(i,j)与之对应,如果将该等价变化过程定义为 R_A,则由 I_{fc} 生成序偶集 $I_{fcs}=\{(i,j)\}$的过程如公式(6)所示.

$$I_{fc'}\xleftrightarrow{R_1}I_{fcs}:\varepsilon_{ij}=R_A((i,j)) \tag{6}$$

R_A 过程是将 I_{fcs} 中各序偶第 1 项按照自然数大小进行排序.排序后,所有序偶的后项构成全排列 I_{fca},该全排列即为匹配方案 I_{fcs} 的等价转换.该转换过程的等价性由定理 2 保证.

定理 2. $\forall(i,j)\in I_{fcs}$,使得 R_A:$(i,j)\leftrightarrow j$,则 $R_A(I_{fc'},I_{fca})$为原有匹配方案的等价转换.

证明:易证$|I_{fca}|=|I_{fc}|$,显然,R_A 为$(i,j)\leftrightarrow j$ 是一种等价转换.证毕. □

3.3 顺序转换

虽然 I_{fca} 已经具备了一般优化算法所能处理的序列结构,但根据约束,序列中某一个数字仅允许出现 1 次.该约束导致优化算法必须消耗额外的资源确保 I_{fca} 的有效性.在维度较高的问题上,该约束将制约算法的求解效率.因此还需进一步对 I_{fca} 转换,使得匹配方案的每一维取值独立于其他维度的取值.基于此,本文利用顺序表示方法,将各维度具有约束关系的 I_{fca} 等价转化为各维度间无约束的 I_{fcr}.

顺序表示过程是将 I_{fca} 中所有元素依次构成一个顺序表,从 I_{fca} 的最左端开始,依次从顺序表中选取元素.选取后,从顺序表中移除该元素,并将记录该元素在顺序表中的下标.当 I_{fca} 中所有元素处理完成后,由下标组成的序列 I_{fcr} 为原排列 I_{fca} 的顺序转换.转换后,I_{fcr} 在第 i 维上的取值范围为$[1,n-i]$,独立于其他维度的取值.容易证明,从 I_{fcr} 到 I_{fca} 的转换也是等价转换.

在匹配问题的求解算法中,将利用 I_{fcr} 进行演化.每一代演化结束后,采取顺序转换、排列转换和方阵转换的逆过程,将 I_{fcr} 转换为 I_{fc}.使用公式(3)计算匹配方案 I_{fc} 的匹配精度.

4 求解服务匹配问题的协同演化算法

4.1 算法设计思路

根据 NFL(no free lunch)[11]理论,不存在适用于所有问题的优化算法;但对于一些特定的优化问题,则可通过协同演化进行优化算法设计,提高优化算法获取最优解的期望.从结构上看,协同演化算法分为串行协同[12−14]和并行协同[15−17].其中,串行协同是指在每一代中顺序执行多个演化策略;并行协同通过策略选择概率,在多个策略中依历史信息,选择适用于当前演化过程的演化策略.与串行协同相比,并行协同对不同优化问题的适应性更强、时间复杂度更低.

本文基于 PSO(particle swarm optimization)和 SA(simulated annealing)演化策略设计适用于服务匹配问题的并行协同演化算法(PSO-SA).PSO 易于实现,且具有更快的收敛速度,但容易陷入到局部最优解,表现为极强的趋同性和较低的种群多样性.SA 采取概率的方式保留一些具有潜在寻优能力的候选解,提高了种群的多样

性,但其收敛速度较慢.SA 与 PSO 的演化特征具有互补性,结合 PSO 和 SA 进行算法设计,能够提高收敛速度并实现深度和广度的优化过程.

4.2 PSO-SA协同演化算法实现

PSO-SA 的基本流程如图 2 所示.

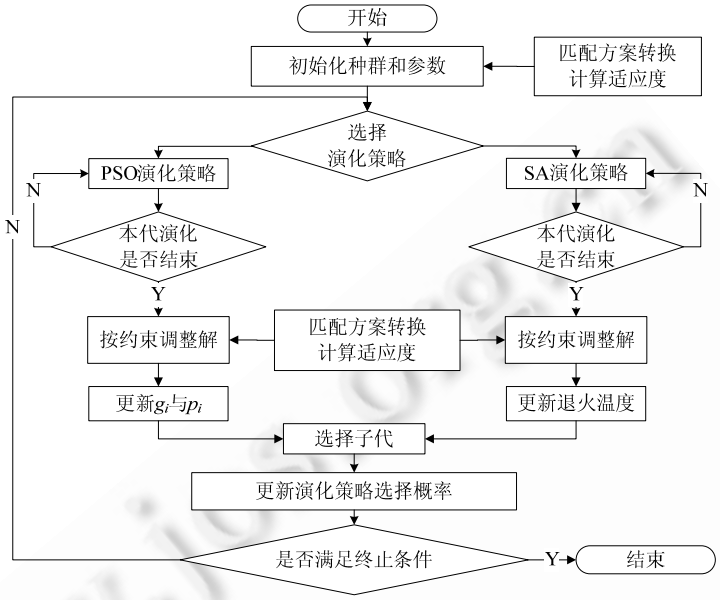

Fig.2 Flow of PSO-SA

图 2 PSO-SA 算法流程

(1) 初始化种群及参数

设 n 为问题规模,S 为种群规模,演化策略的选择概率$\mu^{(1)}=(\mu^{(1)}(\text{PSO}),\mu^{(1)}(\text{SA})$,SA 初始温度 $t^{(1)}$,SA 的温度调节因子α,PSO 中调和因子λ_1和λ_2,扰动因子 r_1和 r_2,迭代次数 G,锦标赛轮数等于种群大小 S,适应度函数为 z.

在第 1 代,种群 $X^{(1)}$中每一个个体可表示为一个三元组向量 $((I_{fcr}^i)^{(1)},\xi,v_i^{(1)}),\forall i\in\{1,2,...,S\}$.每个向量都是 n–1 维,如公式(7)所示.

$$\begin{cases}(I_{fcr}^i)^{(1)}=((I_{fcr}^i)^{(1)}(1),(I_{fcr}^i)^{(1)}(2),...,(I_{fcr}^i)^{(1)}(n-1))\\ \xi=(\xi(1),\xi(2),...,\xi(n-1)) \\ v_i^{(1)}=(v_i^{(1)}(1),v_i^{(1)}(2),...,v_i^{(1)}(n-1))\end{cases},i\in\{i\mid i\in Z^+,i\in[1,S]\} \tag{7}$$

其中,$(I_{fcr}^i)^{(1)}$ 是决策变量,即匹配方案,随机显示为长度为 n–1 的自然序列;ξ为匹配方案的约束变量,控制匹配方案各维度的取值范围,$\xi(d)=n-d,d\in\{1,2,\ldots,n-1\}$; $v_i^{(1)}$ 为辅助变量,表示匹配方案不同维的速度,初值为$[v_{\min},v_{\max}]$内的随机数.

在第 k 代开始时,PSO-SA 根据策略选择概率$\mu^{(k)}$选择本代的演化策略.

(2) PSO 演化策略

PSO-SA 所使用的 PSO 演化策略与标准 PSO 策略类似,均采用群体的最优位置和个体的历史最优位置来调控粒子当前速度.不同之处在于,PSO-SA 综合考虑调和因子λ_1和λ_2对惯性因子的影响,通过公式(8)计算ω.

$$\begin{cases}\phi=\lambda_1+\lambda_2,\lambda_1,\lambda_2\geqslant 2\\ \omega=2/|2-\phi-\sqrt{\phi(\phi-4)}|\end{cases} \tag{8}$$

对于决策变量 $(I_{fcr}^i)^{(k)}$, 采用 inter-weight 的方法,将ω不仅作用于前一时刻的速度,也作用于本代寻优过程中

趋向个体的历史最优位置 p_i 和群体的历史最优位置 g_i.同时,根据匹配问题的离散特征,对演化后的 $(I^i_{fcr})^{(k)}$ 进行离散化操作.综合上述考虑,按照公式(9)产生 k+1 代候选解 $X'^{(k+1)}$.

$$\begin{cases}(v'_i)^{(k+1)}=\omega(v_i^{(k+1)}+\lambda_1 r_1(p_i^k-(I^i_{fcr})^{(k)})+\lambda_2 r_2(g_i^k-(I^i_{fcr})^{(k)}))\\(I^i_{fcr})'^{(k+1)}=\lceil(I^i_{fcr})^{(k)}+(v'_i)^{(k+1)}\rceil\end{cases}\tag{9}$$

(3) SA 演化策略

SA 演化策略随机选择决策变量 $(I^i_{fcr})^{(k)}$ 的某一维 d,以离散高斯分布 $DN_d(-\xi(d),\xi(d))$在约束范围内获取该维度的变异步长,按照公式(10)产生子代的候选解 $X'^{(k+1)}$.

$$\begin{cases}(I^i_{fcr})'^{(k+1)}(d)=(I^i_{fcr})^{(k)}(d)+DN_d(-\xi(d),\xi(d))\\(v'_i)^{(k+1)}=v_i^{(k)}\end{cases}\tag{10}$$

按照公式(11)确定候选子代的决策变量,其中,$I^{(k+1)}_{fcr}\in X^{(k+1)}$,$(I^*_{fcr})^{(k+1)}\in X'^{(k+1)}$.

$$\begin{cases}I^{(k+1)}_{fcr}\leftarrow(I^*_{fcr})^{(k+1)}, & z((I^*_{fcr})^{(k+1)})\geqslant z(I^{(k+1)}_{fcr})\\I^{(k+1)}_{fcr}\xleftarrow{P}(I^*_{fcr})^{(k+1)}, & z((I^*_{fcr})^{(k+1)})<z(I^{(k+1)}_{fcr})\end{cases}\tag{11}$$

在公式(11)中,概率 P 的计算方法如公式(12)所示.

$$P(I^{(k)}_{fcr},(I^*_{fcr})^{(k)},t(k))=\min\left\{1,\exp\left(-\frac{z((I^*_{fcr})^{(k)})-z(I^{(k)}_{fcr})}{t^{(k)}}\right)\right\}\tag{12}$$

公式(12)表明:在适应度差距较小且温度较高时,系统更容易接受适应度较差的个体作为子代;而随着代数的增加,温度逐渐降低,系统接受较差个体的概率变小.

(4) 按约束调整解

经过 PSO 或 SA 演化后,会出现决策变量不满足约束ξ的情况,采用公式(13)调整不满足约束的维度 d.

$$I^i_{fcr}(d)=\begin{cases}[I^i_{fcr}(d)-1]\bmod(\xi(d)-1)+1, & I^i_{fcr}(d)>\xi(d)\\[I^i_{fcr}(d)-\xi_i(d)]\bmod(1-\xi(d))+\xi(d), & I^i_{fcr}(d)<1\\I^i_{fcr}(d), & I^i_{fcr}(d)\in[1,\xi(d)]\end{cases}\tag{13}$$

(5) 更新 p_i 和 g_i

在第 k 代演化结束后,若采用 PSO 演化策略,则需根据候选子代 $X'^{(k+1)}$和父代 $X^{(k)}$的适应度,更新公式(9)中个体的历史最优位置 p_i 及种群历史最优位置 g_i,如公式(14)和公式(15)所示.

$$p_i^{(k+1)}=\begin{cases}p_i^{(k)}, & z(I^i_{fcr}{}^{(k)})\leqslant z(p_i^{(k)})\\(I^i_{fcr})^{(k)}, & z((I^i_{fcr})^{(k)})>z(p_i^{(k)})\\(I^i_{fcr})^{(k)}\in X'^{(k+1)}\wedge i\in\{i\mid i\in Z^+,i\in[1,S]\}\end{cases}\tag{14}$$

$$g_i^{(k+1)}=\max_{p_i^{(k)}}(z(p_i^{(k)})),i=1,...,P_s\tag{15}$$

(6) 更新退火温度

在第 k 代演化结束后,若采用 SA 演化策略,则需计算 k+1 代的退火温度,如公式(16)所示.

$$t^{(k+1)}=\alpha t^{(k)}\tag{16}$$

其中,α为温度调节因子,控制相邻代数间温度下降的差异.

(7) 选择子代

在第 k 代演化结束后,从$|X^{(k)}\cup X'^{(k+1)}|$个个体中,采用锦标赛方法,选择 S 个个体构成最终子代 $X^{(k+1)}$.具体操作过程:对于$\forall x\in X^{(k)}\cup X'^{(k+1)}$,初始化其得分 s_x=0,随机选择另一个个体 $y\in X^{(k)}\cup X'^{(k+1)}$,比较个体的适应度,即,比较 $z(x.I_{fcr})$和 $z(y.I_{fcr})$.如果 $z(x.I_{fcr})\geqslant z(y.I_{fcr})$,则 $s_x=s_x$+1.上述比较过程执行 S 次,最终,根据个体评分 s_x 的排序,选择序列中前 S 个个体构成子代 $X^{(k+1)}$.

(8) 更新策略的选择概率

第 k 代演化结束后,根据 $X^{(k+1)}$中每个子代个体及其对应父代个体 $X^{(k)}$,对$\mu^{(k)}$进行如下更新.

针对 $X^{(k+1)}$中个体的决策变量 $I_{fcr}^{(k+1)}$,如果是来自于父代的演化,说明第 k 代使用的演化策略为当前的优势策略.为促使演化过程尽快收敛,应提高该演化策略 h 的选择概率$\mu^{(k)}(h)$,同时降低其他变异策略 l 的选择概率$\mu^{(k)}(l)$,并确保$\mu^{(k)}(h)+\mu^{(k)}(l)=1$,如公式(17)所示.

$$\begin{cases}\mu^{(k+1)}(h)=\mu^{(k)}(h)+\gamma(1-\mu^{(k)}(h))\\ \mu^{(k+1)}(l)=\mu^{(k)}(l)-\gamma\mu^{(k)}(l)\end{cases} \tag{17}$$

同理,若演化策略 h 为劣势策略,则使用公式(18)调整策略选择概率.

$$\begin{cases}\mu^{(k+1)}(l)=\mu^{(k)}(l)+\gamma(1-\mu^{(k)}(l))\\ \mu^{(k+1)}(h)=\mu^{(k)}(h)-\gamma\mu^{(k)}(h)\end{cases} \tag{18}$$

在公式(17)和公式(18)中,γ控制概率调整的步长.

4.3 算法的时间复杂度分析

对于优化算法,时间复杂度与问题规模 n 有密切关系.如果设 $z(n)$表示计算适应度的多项式函数,S 为种群规模,G 为算法固定迭代次数,则 PSO-SA 的时间复杂度分析如下.

算法每执行一次 PSO 演化过程的时间复杂度见表 1,算法每执行一次 SA 演化过程的时间复杂度见表 2.

Table 1 Time complexity analysis of PSO

表 1 PSO 时间复杂度分析

步骤名称	时间复杂度
更新粒子的速度和位移	$O(Sn)$
按约束调整解	$O(Sn)$
更新局部和全局最优粒子	$O(Sz(n))$

Table 2 Time complexity analysis of SA

表 2 SA 时间复杂度分析

步骤名称	时间复杂度
温度更新及位移调整	$O(S)$
按约束调整解	$O(S)$
退火概率选择	$O(Sz(n))$

算法每执行一次子代过程和策略选择概率更新过程的时间复杂度见表 3.

Table 3 Time complexity analysis of the population selection and the probability adjustment

表 3 子代选择及策略选择概率更新时间复杂度分析

步骤名称	时间复杂度
锦标赛选择子代	$O(S\log(Sz(n)))$
策略选择概率更新	$O(S)$

如果设 PSO-SA 执行 PSO 过程 G_1 次,执行 SA 算法 G_2 次,满足 $G=G_1+G_2$,则 PSO-SA 算法的时间复杂度为

$$O(2G_1Sn+2G_2S+GS+GSz(n)+GS\log(Sz(n))).$$

由于 $G>G_1$,$G>G_2$ 且 $z(n)\sim n$,化简后的时间复杂度为

$$O(3GSn+3GS+GS\log S+GS\log n).$$

如果 PSO-SA 采取固定迭代次数 G 以及种群大小 S,则算法的时间复杂度可进一步化简为

$$O(3GSn+GS\log n)\sim O(n+\log n).$$

在设定较低的迭代次数 G 以及种群大小 S 时,与传统二分图算法的时间复杂度 $O(n^3)$相比,PSO-SA 的时间复杂度 $O(n+\log n)$随 n 上升的速度要远低于 $O(n^3)$.

5 实验分析

5.1 数据准备

实验在 Intel Core Duo 2.66GHz 处理器和 2G 内存的主机上运行,通过 Python,OWL-S API 实现 PSO-SA 及其他对比算法.采用 OWL-S TC(OWL-S service retrieval test collection)[18] V4.0 中来自 5 个领域中 600 个 OWL-S 服务进行服务匹配实验.采用已提出的语义相似度计算方法[19],生成问题维度分别从 10~60 的 6 个匹配矩阵作为后续实验的数据集.

由于实际服务发现过程要求服务匹配能够在有限时间内及时返回相对较优的匹配方案,因此采取限定迭代次数作为终止条件来测试不同算法在数据集上的执行效果.

5.2 基本参数配置

根据第4.3节,算法的迭代次数G和种群规模S皆能影响算法效率.根据服务发现过程对服务匹配的效率要求,在后续实验中,对各种对比算法均设定同样较小的种群规模和较低的迭代次数,分析各算法在有限迭代次数内的寻优情况.实验基本参数的配置情况见表4.

Table 4 Basic configurations of the parameters of PSO-SA

表4 PSO-SA基本参数配置

参数名称	参数数值
SA初始温度$t^{(1)}$	$t^{(1)}$=10000
SA温度调节因子α	α=0.95
PSO调和因子λ_1和λ_2	$\lambda_1=\lambda_2=2$
PSO扰动因子r_1和r_2	均为[0,1]间的随机数
种群规模S	S=10
迭代次数G	G=100

5.3 实验内容

在实验中,横向对比算法包括Greedy为贪心算法[4]、GA为遗传算法、PSO为粒子群算法、SA为模拟退火算法.纵向对比组为串行协同和并行协同,串行协同包括PSOWSA[12],PSOWGSA[13]和HPSOSA[14],PSOWSA在每一代中顺序执行PSO和SA,PSOWGSA只对种群中全局最优位置执行SA变异,HPSOSA仅在PSO多次变异但最优值保持不变时,采取SA策略尝试跳出局部最优.并行协同的成果较少,主要以PSO-SP[17]为代表,它采取PSO和SP(single point)进行协同演化.

5.4 实验结果分析

5.4.1 主要参数取值

(1) γ取值分析

根据公式(17)和公式(18),当γ>0.5时,即使种群中仅有0.5S个个体的变异来自父代,也会产生$\mu^{(k+1)}(h)\to 0$和$\mu^{(k+1)}(l)\to 1$的效果,即,在第k+1代,PSO-SA必然演化策略l.因此,理论上,选取较大的γ会导致策略选择概率变化过于敏感,造成算法的过分拟合.故为进一步确定γ的取值,分析γ=0.1,0.2,0.3,0.4,0.5时,PSO-SA获得匹配结果适应度的统计量,以10维和60维实验结果为例进行说明,见表5和表6.表中数值为独立实验50次所得最优匹配精度的平均值、标准差和中位数.

Table 5 Test statistics of various γ on d=10

表5 在10维问题上γ不同取值的统计结果

	0.1	0.2	0.3	0.4	0.5
平均值	6.997 6	7.003 9	6.995 8	6.991 6	6.998 3
标准差	0.030 6	0.018 8	0.031 5	0.033 5	0.035 4
中位数	7.008 7	7.012 6	7.006 6	6.995 7	7.011 4

Table 6 Test statistics of various γ on d=60

表6 γ不同取值在60维问题上的统计结果

	0.1	0.2	0.3	0.4	0.5
平均值	47.093 5	47.385 0	47.080 2	47.086 7	47.020 0
标准差	0.746 8	0.508 0	0.668 1	0.800 8	0.686 1
中位数	46.989 8	47.373 7	46.934 9	47.323 4	47.130 0

根据实验结果,当γ=0.2时,各项统计量结果较优.后续实验中,γ=0.2.

(2) $\mu^{(1)}$的取值分析

在不同问题维度的数据集上,$\mu^{(1)}$(PSO)=0.1,0.2,…,0.9时,PSO-SA的最优匹配方案的适应度增长趋势相同.在同一维度中,采取不同$\mu^{(1)}$(PSO)的初值所获得标准偏差的平均值为0.105,最终匹配方案的适应度间的差异较小.由上述实验结果,PSO-SA优化效果不依赖于$\mu^{(1)}$(PSO)的取值.在后续仿真实验中,$\mu^{(1)}$(PSO)=$\mu^{(1)}$(SA)=0.5.

5.4.2 各算法求解精度对比实验

在问题规模为d=10,30,60的数据集上,分别运行横向对比组、纵向对比组以及PSO-SA算法50轮,计算各

轮在 100 代时所获得的匹配方案的统计量:最大值 Max、最小值 Min、平均值 Avg、标准差 Std 和中位数 Med.横向对比组的统计量见表 7,纵向的统计量见表 8 和表 9.

Table 7 Test statistics of the matching solutions between PSO-SA and other transverse algorithms

表 7 PSO-SA 及横向对比组中各算法求解匹配方案的统计量

维度	算法	最大值	最小值	平均值	方差	中位数
d=10	PSO-SA	7.024 024	6.973 552	7.003 893 7	0.018 788 414	7.012 608
	GA	7.022 248	6.998 104	7.014 721 94	0.005 117 766	7.015 184 5
	SA	6.651 891	6.428 753	6.441 276 18	0.045 560 978	6.428 753
	PSO	7.150 216	6.358 192	7.067 608 56	0.744 146 565	7.188 517
	Greedy	5.870 664	5.870 664	5.870 664		5.870 664
d=30	PSO-SA	26.646 476	24.052 79	25.356 598 5	0.587 055 125	25.311 265
	GA	23.900 531	22.214 44	22.934 858 28	0.401 525 958	22.887 487
	SA	22.913 837	19.780 268	21.113 655 24	0.641 418 488	21.097 305 5
	PSO	26.195 84	20.284 376	23.748 515 04	1.727 437 527	24.405 836
	Greedy	6.010 848	6.010 848	6.010 848		6.010 848
d=60	PSO-SA	48.629 135	46.397 711	47.384 989 62	0.508 027 999	47.373 726 5
	GA	46.295 939	44.150 834	44.820 384 1	0.473 684 729	44.691 559
	SA	44.783 262	41.542 214	43.362 673 4	0.797 909 171	42.982 154 5
	PSO	48.628 528	37.896 751	44.491 809 14	3.038 247 583	44.972 909
	Greedy	16.723 721	16.723 721	16.723 721		16.723 721

Table 8 Test statistics of the matching solutions between PSO-SA and other serial co-evolutional algorithms

表 8 PSO-SA 及串行协同对比组中各算法求解匹配方案的统计量

维度	算法	最大值	最小值	平均值	方差	中位数
d=10	PSO-SA	7.024 024	6.973 552	7.003 893 7	0.018 788 414	7.012 608
	PSOWSA	7.023 931	6.762 774	6.952 877 26	0.075 652 091	6.990 127 5
	PSOWGSA	6.872 089	6.312 011	6.535 616 26	0.126 063 766	6.511 615
	HPSOSA	7.023 931	6.841 363	7.000 085	0.041 288	7.018 555
d=30	PSO-SA	26.646 476	24.052 79	25.356 598 5	0.587 055 125	25.311 265
	PSOWSA	24.497 896	21.537 138	22.826 9442 8	0.710 071 926	22.811 175
	PSOWGSA	21.329 614	17.715 3	19.416 291 62	0.952 757 701	19.386 161 5
	HPSOSA	26.404 695	22.620 489	24.982 140 22	0.782 056 64	25.121 887
d=60	PSO-SA	48.629 135	46.397 711	47.384 989 62	0.508 027 999	47.373 726 5
	PSOWSA	46.661 154	43.462 838	44.814 270 74	0.678 503 563	44.734 443
	PSOWGSA	44.460 552	39.306 49	40.915 059 66	1.064 610 438	40.799 645 5
	HPSOSA	48.422 928	43.147 897	46.005 058 14	1.319 603 152	45.836 086

Table 9 Test statistic of the matching solutions between PSO-SA and other paralleled co-evolutional algorithms

表 9 PSO-SA 及并行协同对比组中各算法求解匹配方案的统计量

维度	算法	最大值	最小值	平均值	方差	中位数
d=10	PSO-SA	7.024 024	6.973 552	7.003 893 7	0.018 788 414	7.012 608
	PSO-SP	7.023 56	6.853 64	6.996 214 36	0.032 570 812	7.007 292
d=30	PSO-SA	26.646 476	24.052 79	25.356 598 5	0.587 055 125	25.311 265
	PSO-SP	26.117 443	22.236 26	24.110 030 16	0.867 341 185	24.058 075
d=60	PSO-SA	48.629 135	46.397 711	47.384 989 62	0.508 027 999	47.373 726 5
	PSO-SP	47.386 556	44.376 875	46.036 255 6	0.712 939 457	46.116 916

- 横向对比组分析

当 *d*=10 时,PSO-SA 的平均值上低于 PSO 和 GA 结果 0.06 和 0.01.当 *d*>10 时,PSO-SA 在除标准差外的各统计量上均优于组内其他算法.虽然 GA 在所有维度的问题中均具有较低的方差,但其所得到的平均匹配值劣于 PSO-SA,说明 GA 在寻优过程中出现了早熟的情况.

- 串行协同演化对比组分析

当 *d*=10 时,PSO-SA 运行 50 轮所获得的中位数略劣于 HPSOSA.在其他维度的各项统计量上,PSO-SA 均优于 PSOWSA,PSOWGSA 和 HPSOSA.

- 并行协同演化对比组分析

从各维度的统计结果上看:虽然 PSO-SP 采取与 PSO-SA 类似的演化策略概率选择方式,但是由于使用不同演化策略,在服务匹配问题上,PSO-SP 的寻优效果劣于 PSO-SA.

综合上述实验结果,在 d=10 时,虽然 PSO-SA 可获得相对较优的匹配方案,但 PSO 通常具有更好的寻优精度.在 d>10 时,其他算法均无法获得比 PSO-SA 更优的匹配方案.随着用户需求的日益复杂以及服务库中候选服务数量的上升,具有高维特征的复合服务匹配问题将远多于低维特征的原子服务匹配问题.然而,一旦出现了低维度的匹配问题,也可以直接采用本文所提出的匹配方案转换方法,并直接使用 PSO 算法进行优化.

5.4.3 各算法实际收敛过程对比实验

在问题规模为 d=10,30,60 的数据集上,分别运行横向对比组、纵向对比组和 PSO-SA,记录各算法每一代的最优匹配方案的适应度 z,分析算法的实际收敛情况.横向对比组中各算法的收敛情况如图 3~图 5 所示,纵向对比组中各算法的收敛情况如图 6~图 8 所示.在各幅收敛图中,横坐标表示算法迭代代数,纵坐标表示算法获得匹配方案的适应度 z.

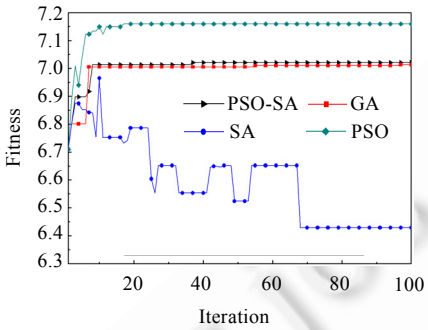

Fig.3 Convergence of the transverse algorithms group (d=10)

图 3 横向对比组中各算法收敛情况(d=10)

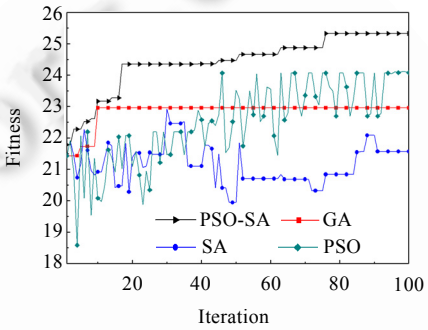

Fig.4 Convergence of the transverse algorithms group (d=30)

图 4 横向对比组中各算法收敛情况(d=30)

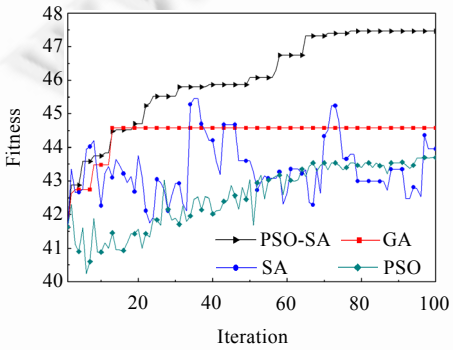

Fig.5 Convergence of the transverse algorithms group (d=60)

图 5 横向对比组中各算法收敛情况(d=60)

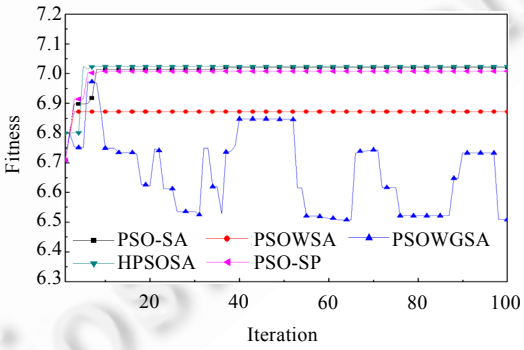

Fig.6 Convergence of the vertical algorithms group (d=10)

图 6 纵向对比组中各算法收敛情况(d=10)

- 横向对比组分析

在 d=10 时,PSO-SA 的收敛情况与 GA 接近,但最终收敛情况劣于 PSO.SA 算法出现了严重的退化现象.随着问题维度的上升,PSO-SA 和 GA 均表现出稳定进化的特征,但 GA 出现了早熟现象,这一过程也解释了在表 7 中 GA 的标准差较低的原因.单纯采用 PSO 和 SA 对服务匹配问题进行优化时,匹配方案的适应度随迭代过程而发生波动和退化现象,导致算法无法收敛.

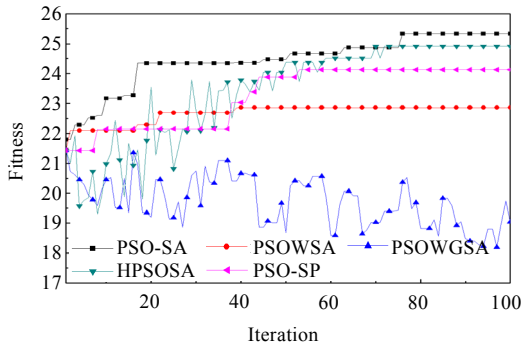

Fig.7 Convergence of the vertical algorithms group (d=30)

图 7 纵向对比组中各算法收敛情况(d=30)

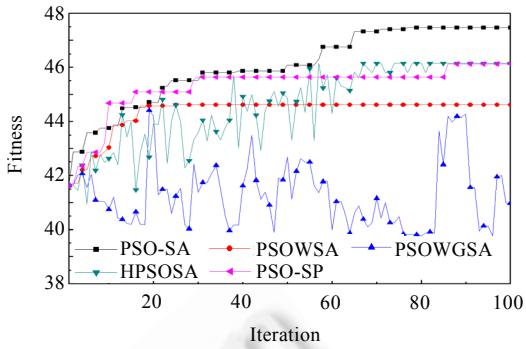

Fig.8 Convergence of the vertical algorithms group (d=60)

图 8 纵向对比组中各算法收敛情况(d=60)

- 纵向对比组分析

在 d=10 时,PSO-SP,PSO-SA 与 HPSOSA 的收敛情况类似.虽然 PSOWSA 也表现出稳定的进化特征,但容易陷入局部最优解,表现为趋同特征.PSOWGSA 算法则整体出现波动较大及退化现象.随着问题维度的上升,PSO-SP,PSO-SA 及 PSOWSA 均表现出稳定进化特征,但 PSOWGSA 和 HPSOSA 出现了严重的波动情况,导致算法无法收敛.

综合上述分析,PSO-SA 能够在有限迭代次数内克服经典优化算法和现有协同演化算法在处理匹配问题时出现的早熟(GA,PSOWGSA)、难以收敛(PSO,PSOWGSA,HPSOSA)以及退化(SA,PSOWGSA)等问题,实现稳定进化的收敛过程,对不同维度匹配问题具有较好的适应性.与 PSO-SP 相比,PSO-SA 更适合处理服务发现的匹配问题.

5.4.4 各算法在线响应时间的对比实验

分别运行 PSO-SA 以及纵向对比组中各种算法 50 轮,计算每轮算法运行 100 代所消耗时间的平均值,判断各算法的在线响应时间,如图 9 所示,横坐标为问题维度,纵坐标为各算法的在线响应时间.

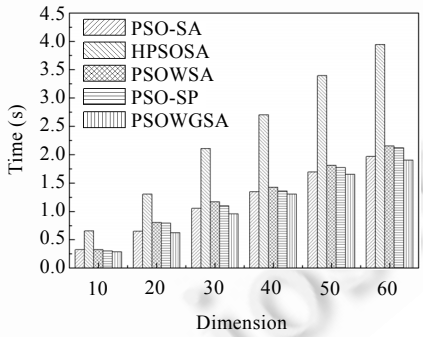

Fig.9 Online response time of the algorithms

图 9 各算法在线响应时间

在同一维度中,各算法消耗时间的关系:PSOWSA>PSOWGSA>HPSOSA>PSO-SA>PSO-SP.其中,PSOWSA 所消耗时间近似是 PSO-SA 消耗时间的 2 倍.原因为:PSOWSA 的每一次迭代相当于先后执行了 PSO 操作和 SA 操作,即,等价于 PSO-SA 执行了两代演化操作.HPSOSA 与 PSOWGSA 在演化过程中仅针对最优个体执行 SA 策略,但其仍然是一种串行演化过程,相当于在执行了 PSO 操作的基础上,有选择地执行 SA 操作,导致算法的在线响应时间微高于 PSO-SP 和 PSO-SA.PSO-SP 采用的单点操作无需进行概率选择过程,其在线响应时间较

PSO-SA 略低.各算法的在线响应时间均随着维度的上升呈现线性提高,实际相应时间满足第 4.3 节的分析结果.

综合上述分析,并行协同演化算法的在线响应时间要低于串行协同演化算法.PSO-SP 虽然具有最低的响应时间,但综合第 5.4.2 节和第 5.4.3 节的实验结果,PSO-SP 的寻优精度和收敛效果均劣于 PSO-SA;且 PSO-SA 的在线响应时间在不同维度匹配问题上仅比 PSO-SP 高 0.05s,在用户可接受范围内,满足服务发现对匹配算法高效的响应需求.

6 结束语

服务发现技术能够在海量服务中快速获取满足用户需求的候选服务,而匹配结果是决定服务发现质量的关键.针对服务匹配过程中存在的匹配概念狭窄、匹配方案的自然表示难以被优化算法处理及匹配算法的时间复杂度较高等问题,提出了适用于求解服务匹配问题的协同演化算法 PSO-SA.实验结果表明:PSO-SA 在处理不同维度的匹配问题时,能够在有限的迭代次数内获得比其他算法更高适应度的匹配方案,体现出较高的问题适应性以及高效的响应能力.

在后续的研究过程中,尝试引入其他演化策略,如人工蜂群算法,与现有 PSO-SA 结合,吸收各算法的演化优势,形成多策略协同演化过程.同时,服务匹配仅作为 PSO-SA 的一方面应用,对于其他组合优化问题,还需设计基于问题启发的解的转换方法及适应度计算方法,并进一步验证 PSO-SA 的适应性.

References:

[1] Kuang L, Deng SG, Li Y, Wu J, Wu ZH. Using inverted indexing to facilitate composition-oriented semantic service discovery. Ruan Jian Xue Bao/Journal of Software, 2007,18(8):1911−1921 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/1911.htm

[2] Wang PW, Jin Z, Liu HY. The function description and discovery of Internetware entites. Science in China (Series F: Information Sciences), 2009,39(12):1271−1287 (in Chinese with English abstract).

[3] Zeng LZ, Benatallah B, Ngu A, Dumas M. Qos-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004,30(5):311−327. [doi: 10.1109/TSE.2004.11]

[4] Qiu JN, Du QY, Cui Y. Research on integrated semantics matching algorithm in service matching model. Journal of Dalian University of Technology, 2007,47(6):914−919 (in Chinese with English abstract).

[5] Deng SG, Yin JW, Li Y, Wu J, Wu ZH. A method of semantic Web service discovery based on bipartite graph matching. Chinese Journal of Computers, 2008,31(8):1364−1375 (in Chinese with English abstract).

[6] Srinivasan N, Paolucci M, Sycara K. Semantic Web service discovery in the OWL-S IDE. In: Gul A, ed. Proc. of the Hawaii Int'l Conf. on System Sciences (HICSS 2006). Los Alamitos: IEEE Computer Society Press, 2006. 1−10. [doi: 10.1109/HICSS.2006.431]

[7] Yue K, Wang XL, Zhou AY. Underlying techniques for Web services: A survey. Ruan Jian Xue Bao/Journal of Software, 2004, 15(3):428−442 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/428.htm

[8] Qiu T, Hu XH, Li PF, Ma HT. A semantic matchmaking system mechanism for Web service discovery based on OWL-S. Acta Electronica Sinica, 2010,38(1):42−47 (in Chinese with English abstract).

[9] Ou WJ, Zeng C, Xiang XM ,Peng ZY, Li DY. Efficient Web service query approach based on concept Relaxation. Chinese Journal of Computers, 2011,34(12):2381−2390 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.02381]

[10] Zhang PY, Huang B, Sun YM. Web services matching mechanism based on semantics and QoS-aware aspect. Journal of Computer Research and Development, 2010,47(5):780−787 (in Chinese with English abstract).

[11] WolPert DH, Maeteady WG. No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation, 1997,1(1):67−82. [doi: 10.1109/4235.585893]

[12] Behnamian J, Ghomi SMT. Development of a PSO-SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. Expert System and Applications, 2010,37(2):974−984. [doi: 10.1016/j.eswa.2009.05.079]

[13] Kathpal S, Vohra R, Singh J, Sawhney RS. Hybrid PSO-SA algorithm for achieving partitioning optimization in various network application. In: Rajesh R, ed. Proc. of the Procedia Engineering. Holland: Elsevier Press, 2012. 1728−1734. [doi: 10.1016/j.proeng.2012.06.210]

[14] Lhassane I, Mahmoud M, René S, Maha IA. Hybrid PSO-SA type algorithms for multimodal function optimization and reducing energy consumption in embedded systems. Applied Computational Intelligence & Soft Computing, 2011,2011(3):1−11. [doi: 10.1155/2011/138078]

[15] Dong HB, He J, Huang HK, Hou W. Evolutionary programming using mixed mutation strategy. Information Sciences, 2007,177(1): 312−327. [doi: 10.1016/j.ins.2006.07.014]

[16] Dong HB, Huang HK,Yin GS, He J. An overview of the research on co-evolutionary algorithms. Journal of Computer Research and Development, 2008,45(3):454−463 (in Chinese with English abstract).

[17] Zhang X, Dong HB. A new co-evolutionary programming using operators of single-point and particle swarm. In: Han Y, ed. Proc. of the Int'l Conf. on Computer Application and System Modeling. Piscataway: IEEE Computer Society, 2010. 5528−5532. [doi: 10.1109/ICCASM.2010.5620277]

[18] Wei DP, Wang T, Wang J. Web service discovery by integrating structure and reference features of description documents. Ruan Jian Xue Bao/Journal of Software, 2011,22(9):2006−2019 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3887.htm [doi: 10.3724/SP.J.1001.2011.03887]

[19] Yin GS, Cui XH, Dong YX. Design on Web services similarity calculation framework. In: Muchin VE, ed. Proc. of the E-Business and Information System Security (EBISS 2010). Piscataway: IEEE Computer Society Press, 2010. 1−4. [doi: 10.1109/EBISS.2010.5473429]

附中文参考文献:

[1] 邝砾,邓水光,李莹,吴健,吴朝晖.使用倒排索引优化面向组合的语义服务发现.软件学报,2007,18(8):1911−1921. http://www.jos.org.cn/1000-9825/18/1911.htm

[2] 王璞巍,金芝,刘红岩.网构软件实体的功能描述及其发现.中国科学(F 辑:信息科学),2009,39(12):1271−1287.

[4] 裘江南,仲秋雁,崔彦.服务匹配模型中综合语义匹配方法研究.大连理工大学学报,2007,47(6):914−919.

[5] 邓水光,尹建伟,李莹,吴健,吴朝晖.基于二分图匹配的语义 Web 服务发现方法.计算机学报,2008,31(8):1364−1375.

[7] 岳昆,王晓玲,周傲英.Web 服务核心支撑技术:研究综述.软件学报,2004,15(3):428−442. http://www.jos.org.cn/1000-9825/15/428.htm

[8] 邱田,胡晓惠,李鹏飞,马恒太.基于 OWL-S 的服务发现语义匹配机制.电子学报,2010,38(1):42−47.

[9] 欧伟杰,曾承,项小明,彭智勇,李德毅.基于概念松弛的高效 Web 服务查询方法.计算机学报,2011,34(12):2381−2390. [doi: 10.3724/SP.J.1016.2011.02381]

[10] 张佩云,黄波,孙亚民.一种基于语义与 QoS 感知的 Web 服务匹配机制.计算机研究与发展,2010,47(5):780−787.

[16] 董红斌,黄厚宽,印桂生,何军.协同演化算法研究进展.计算机研究与发展,2008,45(3):454−463.

[18] 魏登萍,王挺,王戟.融合描述文档结构和参引特征的 Web 服务发现.软件学报,2011,22(9):2006−2019. http://www.jos.org.cn/1000-9825/3887.htm [doi: 10.3724/SP.J.1001.2011.03887]

崔晓晖(1984－),男,山东招远人,博士,讲师,CCF 会员,主要研究领域为服务计算,演化算法.

董红斌(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为演化算法,数据挖掘.

印桂生(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,网构软件.