

## 基于全局 QoS 约束分解的 Web 服务动态选择\*

王尚广<sup>+</sup>, 孙其博, 杨放春

(北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)

### Web Service Dynamic Selection by the Decomposition of Global QoS Constraints

WANG Shang-Guang<sup>+</sup>, SUN Qi-Bo, YANG Fang-Chun

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

+ Corresponding author: E-mail: sguang.wang@gmail.com, http://www.bupt.edu.cn

Wang SG, Sun QB, Yang FC. Web service dynamic selection by the decomposition of global QoS constraints. *Journal of Software*, 2011, 22(7): 1426-1439. <http://www.jos.org.cn/1000-9825/3842.htm>

**Abstract:** With the growing number of alternative Web services in the open Web service environment, users have put forward new demands that bind the runtime of Web services, which requires as much short computation time as possible to satisfy user's end-to-end QoS requirements service composition. Therefore, this paper proposes a Web service dynamic selection approach, based on the decomposition of global QoS constraints, WSDSA (Web service dynamic selection approach). The WSDSA uses as adaptive adjustment method (AAM), based on fuzzy logic and adaptive particle swarm optimization algorithm (APSO), to adaptively decompose global QoS constraints to local constraints with the user's preferences, and then WSDSA can obtain the most appropriate composition service with local selection. Performance evaluations show WSDSA is very effective, and is able to reach optimal or near optimal results with a very low time cost, which satisfies the real time and dynamic service selection.

**Key words:** service selection; service composition; QoS; particle swarm optimization algorithm; fuzzy logic

**摘要:** 随着执行时绑定的 Web 服务的提出及其被大量地应用到开放式服务中,用户对支持快速和动态的 Web 服务组合提出了新的要求.即在组合过程中,用尽可能短的时间花费选择满足用户端到端的 QoS 要求的服务.针对上述要求,提出了一种基于全局 QoS 约束分解的动态服务选择方法(Web service dynamic selection approach,简称 WSDSA).该方法的主要思想是,通过基于模糊逻辑的自适应调整方法(adaptive adjustment method,简称 AAM)和自适应粒子群优化算法(adaptive particle swarm optimization,简称 APSO)将全局 QoS 约束自适应地分解为满足用户偏好的局部约束,然后利用局部最优获得最合适的组合服务.性能评价表明,WSDSA 方法具有较好的有效性,仅用较低的时间花费就能达到或接近全局最优解,能够满足服务选择的实时性和动态性.

**关键词:** 服务选择;服务组合;QoS;粒子群优化算法;模糊逻辑

中图法分类号: TP311 文献标识码: A

\* 基金项目: 国家自然科学基金(60821001); 国家重点基础研究发展计划(973)(2009CB320406, 2009CB320504); 国家高技术研究发展计划(863)(2008AA01A317)

收稿时间: 2009-10-25; 修改时间: 2010-01-20; 定稿时间: 2010-03-16

面向服务的计算模式能够无缝地把现存的各种商业应用服务组合起来,形成新的增值服务来满足用户日益增长的需求<sup>[1,2]</sup>。尤其是在开放动态的 Web 环境中<sup>[3]</sup>,如 Ad-Hoc, WSN 等,需要从众多的候选服务中选择最适当的服务。然而,随着网络形式的多样化(Internet, 3G, WSN 等)和各类服务的增加,满足相同功能需求的服务越来越多,在选择服务时不仅要满足功能方面的需求,还要考虑到服务质量(quality of service, 简称 QoS)这些非功能性的需求,如服务执行时间、费用、可靠性等<sup>[4,5]</sup>。如何从具有不同 QoS 属性的高度动态化的服务中,以一种有效的策略或方法在较短的时间内选择满足用户 QoS 需求的服务,已成为服务组合中所面临的一个重要问题<sup>[1,6]</sup>。

文献[7]给出一个以用户为中心的 Web 环境中的多媒体分发服务场景,如图 1 所示,一个智能手机用户向服务提供者请求最新新闻服务。服务提供者能提供的多媒体内容有及时新闻通知和专题影片,但这需要智能手机支持 MPEG 2 编码方式。对于其他智能手机用户,由于该服务提供者不具备适配能力,所以需要额外的服务来满足用户要求。例如,为了适应目标(智能手机或其他终端)的格式,需要将多媒体内容进行重新转换的译码服务;对及时新闻通知进行文本翻译的翻译服务;将文本流和视频流进行融合从而能够在智能手机的有限屏幕上显示的融合服务;对多媒体内容进行压缩以便支持无线连接传输的压缩服务。

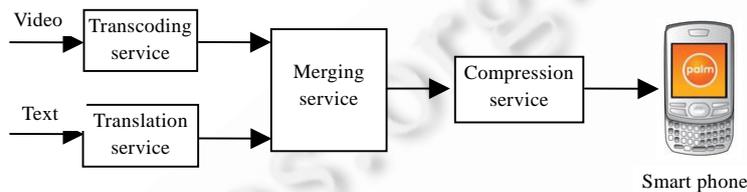


Fig.1 Web service composition example

图 1 Web 服务组合实例

对于上述服务请求,单个服务的功能是有限的,而通过将单个服务组合成功能更强大的服务,也就是将分散的逻辑单元加以集成,从而能够满足用户(服务请求者)的需求。本文将其简称为组合服务,而构成组合服务的单个服务称为服务组件。另外,在工业应用中,组合服务还必须在满足用户一些端到端的 QoS 要求后,才能最终将多媒体内容在智能手机上显示。因此,这就需要组合服务在服务开始和执行时就要选择与用户 QoS 要求相匹配的服务。

然而,由于 QoS 的动态性(如用户切换到较低的网络带宽)或者服务失败(如一些被选定服务无效)可能发生在服务运行期间<sup>[8]</sup>,因此,在应用过程中,对服务请求的快速响应和适应就显得非常重要。此时,具有快速服务选择能力的方法将会对整个组合服务的质量和性能产生巨大影响。所以,如何快速地选择满足用户 QoS 要求的候选服务,对服务的成功组合和组合质量有着重要的意义<sup>[4]</sup>。另一方面,已有的服务选择方法在规划过程中都过多地考虑服务的全局最优但却忽视了对计算时间或时间复杂性的关注,而这个往往导致组合服务在运行过程中一旦出现问题(如 QoS 的动态性或服务失败等),无法在用户要求时间内进行服务重计划。所以,在考虑真实环境的不确定性和动态性的情况下,研究快速的服务选择方法有着重要的现实意义<sup>[9]</sup>。

在 Web 服务组合技术中,Web 服务的选择技术是重要的一环,尤其是全局 QoS 要求的 Web 服务选择。即根据 QoS 要求从多个候选服务中选择合适的服务组件(被选中的候选服务),来满足组合服务整体 QoS 要求或满足服务组件当前所在组合位置上的 QoS 要求<sup>[10]</sup>。用户对组合服务所提出的 QoS 需求将成为对服务组件构成的整个组合逻辑的全局 QoS 约束。一个基于 QoS 感知的服务组合简单流程如下<sup>[11]</sup>:Web 服务工作流执行语言(如 BPEL)首先对一个抽象的组合请求的业务逻辑流程进行描述;其次,服务发现引擎利用现有的基础设施(如 UDDI)通过语法或语义对组合逻辑流程中的事务和存在的服务描述进行功能性匹配,找到有效的 Web 服务;然后,组合逻辑中的每一个事务匹配到了许多候选服务;最后,基于全局 QoS 感知的服务选择中间件从每个事务对应的众多候选服务中选择一个综合 QoS 属性值满足用户端到端的 QoS 要求的服务组件。在面向服务的环境,如果在组合服务运行时,一些服务偏离了 QoS 的当前约束,则必须更换这些服务(如图 1 所示的多媒体分发服务在未来国家突发事件或重大突发自然灾害预警中的应用)。因此,在实际应用中,快速的服务选择机制就变得至关

重要,本文的重点是研究基于非功能属性和应用性能的服务选择.

然而,在日益增多的 Web 服务组件中,具有相同功能和不同 QoS 的服务组件会不可避免地大量出现.这些服务组件会组合出成千上万的具有相同功能与不同 QoS 特征的组合服务<sup>[10]</sup>,使得基于 QoS 感知的服务选择问题成为 NP 难题.其研究难点在于解空间过于庞大,无法在多项式时间内完成搜索<sup>[12]</sup>,导致了基于 QoS 感知的服务选择时间开销过大.

目前,在基于全局 QoS 约束的研究工作中,文献[4,5]将基于全局 QoS 的服务选择问题映射为多维背包问题,并提出使用混合整数规划(mixed integer programming,简称 MIP)方法寻找最优组合.与上述方法类似,文献[3,13]也使用混合整数规划方法用于寻找最优解,但解决问题的侧重点不同,比如普通环境下的服务选择<sup>[3]</sup>等.当服务数量较少时,线性规划方法是很有用的;但当服务数量增加时,其时间复杂性将呈指数增加<sup>[11]</sup>,以至于无法在对时间花费敏感的组合服务中加以应用.

有些研究人员采用遗传算法来解决这类问题,如文献[1,8].相对于混合整数规划,遗传算法的优点是允许 QoS 属性带有非线性聚集功能,然而同样也存在时间复杂性过高的问题.文献[12]提出一种混合最优方法,用于发现近似最优解.该方法与求最优解的纯粹最优方法<sup>[4,5]</sup>相比非常有效,尤其是对于服务运行阶段的决策.另外,文献[12]还提出用于服务组合的两个模型,即联合模型和图模型,并将混合最优方法用于上述两个模型.对于联合模型,时间花费是多项式的;但对于图模型,时间花费仍然是指数的.但是,随着网络中 Web 服务数量的增加,对于使用混合最优方法的联合模型所需的时间花费仍然超出用户要求.尤其是在分布式环境下,时间花费更大<sup>[11]</sup>.

文献[11]也提出了一种混合最优方法,该方法是通过使用混合整数规划方法将全局 QoS 约束映射为质量水平,然后通过将获得的质量水平用于局部最优选择中的局部约束,从而能够快速的选择到满足全局约束的组合服务.该方法能够有效地应用在分布式环境中,较好地解决了基于全局约束的服务选择问题.然而,由于约束分解数量是固定不变的,自适应性不足导致该方法不能很好地满足用户偏好,例如在图 1 中,有些用户希望能够快速地得到组合服务,而对服务质量要求不高(如视频清晰度);而有些用户则希望能够在时间和服务质量中获得平衡.另外,当网络中 Web 服务数量有所增加时,该方法也存在时间花费增长较快的现象,很难满足用户的实时性需求.因此本文认为,单从组合服务搜索算法优化方面入手进行研究,时间性能提升的空间有限,必须从新的视角出发,寻找更进一步提高时间性能的有效途径.

针对上述问题,最理想的情况是在较短的时间内选择各个 QoS 属性值都是最优的服务组件返回给用户.但是实际运行中的服务并不能使这些 QoS 属性值都达到最优,因为有些 QoS 属性是相互冲突的(如执行时间短的服务,其执行费用往往比较高)或 QoS 是动态变化的<sup>[2]</sup>.所以,基于 QoS 感知的服务选择的最终结果,需要在服务选择的时间花费和全局 QoS 最优之间加以折衷,得到一个在开放式服务环境下满足用户 QoS 要求的时间花费较低的最优解或近似最优解.

因此,我们提出一种时间复杂度较低的服务动态选择方法——WSDSA(Web service dynamic selection approach).该方法使用基于模糊逻辑的调整方法(adaptive adjustment method,简称 AAM)和自适应粒子群优化算法(adaptive particle swarm optimization,简称 APSO)将全局 QoS 约束动态分解为局部约束,然后通过局部最优选择进行服务组合,从而解决了基于 QoS 感知的 Web 服务选择中的时间复杂性过高的问题.

我们提出 AAM 方法的目的是为了使得约束分解具有自适应性,从而能够更好地满足用户偏好.该方法通过对监控到的用户偏好信息及 WSDSA 相关输出信息进行模糊推理,最终得到一个满足用户偏好的分解数量.我们提出 APSO 算法的目的是为了能够在较低的时间花费内求解出最优局部约束,该算法针对粒子群优化算法在解决组合优化问题时普遍存在的早熟收敛问题,首先采用基于模糊逻辑的惯性权重的自适应方法,动态调整算法在不同收敛阶段的惯性权重取值;然后,提出一种对全局最优粒子进行线性变异的方法.

为了验证本文提出的方法,进行了性能评价.它包括两部分:有效性分析和仿真对比.分析结果表明,WSDSA 方法具有较低的时间复杂性和较好的有效性.仿真对比表明,该方法获得的组合服务在接近或达到全局最优解的基础上,有效地降低了服务组合的时间花费,较好地满足了对实时性和动态性要求较高的服务选择的需求.

本文第 1 节重点介绍全局 QoS 最优分解,包括 QoS 效用函数、WSDSA 方法以及该方法中的基于模糊逻

辑满足用户偏好的全局 QoS 约束动态分解.第 2 节详细讨论如何使用 APSO 算法求解全局最优质量标尺,包括惯性权重调整、线性变异方法、适应度函数和编码方式.第 3 节介绍性能评价,包括时间复杂性分析和仿真对比.第 4 节总结全文并指出下一步的研究内容.

## 1 全局 QoS 最优分解

为了便于对全局 QoS 最优分解进行介绍,本文首先给出服务组合中相关概念的形式化说明.假设一个组合服务为  $S$ ,如图 2 所示,该服务由多个服务类组成,如  $S_j \in S$ (如飞机订票服务),其中,  $S = \{S_1, \dots, S_n\}$ .服务类  $S_j$  包含多个功能属性相同但 QoS 不同(非功能属性)的候选服务,如  $s_j \in S_j$ (如南航飞机订票服务),其中,  $S_j = \{S_{j1}, \dots, S_{jl}\}$ ,表示有  $l$  个候选服务.上述服务类的信息均由服务代理进行管理,Web 服务可以通过订阅机制随时选择或丢弃服务类及候选服务.

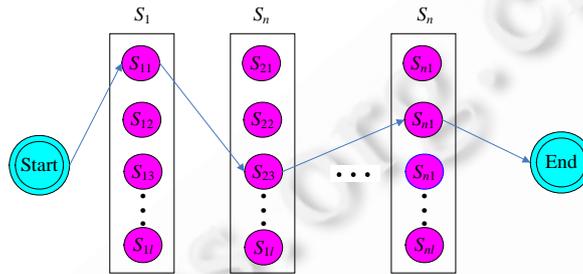


Fig.2 Related concepts for service composition  
图 2 服务组合中的相关概念

### 1.1 QoS 效用函数

Web 服务的 QoS 属性可以分为积极属性和消极属性两类.积极属性考虑 QoS 的最大化,如吞吐率、可用性;消息属性考虑 QoS 的最小化,如价格、响应时间.出于简化目的,本文仅考虑消极属性.对于积极属性,通过负值计算(乘以-1)可以将其转化为消极属性<sup>[11]</sup>.

文献[5,12]给出了 Web 服务中 QoS 的一般属性,如响应时间、价格、可靠性等;也有一些特殊领域中的属性,如多媒体带宽.这些属性能够通过定量计算得到相应的 QoS 属性值,如候选服务  $s$  的 QoS 属性向量  $Qs = \{q_1(s), \dots, q_r(s)\}$ ,  $q_i(s)$  表示  $s$  的第  $i$  个属性的数值函数,它决定了该属性质量的大小.这些 QoS 属性值可以从服务提供商处直接获得(如价格),也可以从先前执行的监测记录中获取(如响应时间),还可以根据用户的反馈得到(如信誉).组合服务的 QoS 属性值可以通过服务组件的 QoS 属性值和相应的组合类型(如顺序、概率、循环等)聚合获得,如组合服务  $S$  的 QoS 属性向量  $QS = \{q_1(S), \dots, q_r(S)\}$ ,  $q_i(S)$  表示  $S$  的第  $i$  个属性的聚合函数,它由被期望的 QoS 属性值的服务组件聚合得到.例如 QoS 属性中价格的聚合函数为  $q(S) = \sum_{j=1}^n q(s_j)$ ,可靠性的聚合函数为

$q(S) = \prod_{j=1}^n q(s_j)$ ,吞吐率的聚合函数为  $q(S) = \min_{j=1}^n q(s_j)$ .虽然服务的组合类型不同(如顺序、概率、循环等),但

均可通过文献[13,14]中的技术将其分解为顺序类型的属性聚合函数.

在 Web 服务组合中,由于候选服务具有多个 QoS 属性,导致其 QoS 属性值的单位或范围不尽相同,不利于从全局 QoS 最优角度对每个候选服务的 QoS 属性值进行计算或评估.因此,QoS 效用函数被提了出来,它用来将服务类中的每个候选服务的 QoS 属性向量  $Qs$  映射到一个实数值,通过该值对每个候选服务进行排序或分类,便于选择到满足 QoS 约束的服务组件.本文中的 QoS 效用函数与文献[11,12]中的方法相似(即简单加权方法),其构造方法是,将 Web 服务中的候选服务(或服务类)与其对应的最大值或最小值进行比较,从而将多个 QoS 属性值进行归一化处理(范围在 0~1),使其转化到一个综合衡量的实数值(独立于每个具体属性的单位或范围),其

中,加权值(范围在 0~1)代表用户的优先级或个人偏好.例如,在顺序聚合模型中,候选服务  $s \in S_j$  和组合服务  $S$  的 QoS 效用函数  $U(s)$  和  $U(S)$  可以通过下式计算得到:

$$U(s) = \sum_{k=1}^r \frac{Q_{j,k}^{\max} - q_k(s)}{Q_{j,k}^{\max} - Q_{j,k}^{\min}} \cdot w_k \left( \text{if } Q_{j,k}^{\max} - Q_{j,k}^{\min} = 0, U(s_k) = \sum_{k=1}^r 1 \cdot w_k \right) \quad (1)$$

$$U(S) = \sum_{k=1}^r \frac{Q_k^{\max} - q_k(S)}{Q_k^{\max} - Q_k^{\min}} \cdot w_k \left( \text{if } Q_k^{\max} - Q_k^{\min} = 0, U(S_k) = \sum_{k=1}^r 1 \cdot w_k \right) \quad (2)$$

且

$$Q_{j,k}^{\min} = \min_{\forall s_{ji} \in S_j} q_k(s_{ji}), \quad Q_k^{\min} = \sum_{j=1}^n Q_{j,k}^{\min},$$

$$Q_{j,k}^{\max} = \max_{\forall s_{ji} \in S_j} q_k(s_{ji}), \quad Q_k^{\max} = \sum_{j=1}^n Q_{j,k}^{\max}.$$

上式中:  $w_k \in R^+$  代表用户偏好,并且  $\sum_{k=1}^r w_k = 1$ ;  $Q_{j,k}^{\min}$  表示服务类  $S_j$  中所有候选服务中第  $k$  个属性的最小值;  $Q_k^{\min}$  表示所有服务类中第  $k$  个属性的最小值;类似地,  $Q_{j,k}^{\max}$  和  $Q_k^{\max}$  为最大值.

众所周知,基于全局 QoS 约束的服务选择是一个最优化问题,也是一个 NP 难题,该问题的重点是从所有可能的服务组合中选择一个 QoS 效用函数值最大且满足全局 QoS 约束的组合服务.例如,如果全局 QoS 约束  $CS = \{C_1, \dots, C_m\}, 0 \leq m \leq r$ , 求得的组合服务  $S = \{s_1, \dots, s_n\} (s_j \in S_j, S_j \text{ 为服务组件}, S_j \text{ 为服务类})$  是最优组合服务,则必须满足以下两个条件<sup>[12]</sup>:

- 组合服务中所有服务类的 QoS 效用函数值  $U(S)$  最大;
- 组合服务的 QoS 聚合值  $q(S) \leq C, \forall C_k \in CS$ .

然而,为了寻找最优的服务组合,需要对所有的候选服务进行服务组合.例如,对于单一路径的组合请求,如果该组合服务有  $n$  个服务类,每个服务类有  $l$  个候选服务,那么将会有高达  $l^n$  个候选服务被选择.如果服务组合是多路径的,那么将会有更多的候选服务被选择<sup>[8]</sup>.可以看出,对上述服务的穷举搜索将会导致时间复杂度过高,耗费昂贵的时间资源,因此,对于实时性较强或在 Web 服务运行时才进行服务选择的工业应用是不合适的,尤其是对候选服务众多和 QoS 动态性较强的工业应用.

## 1.2 WSDSA

本节提出了时间复杂度较低的服务动态选择方法,即 WSDSA.该方法的主要思想是,在服务类中的所有候选服务中引入自适应质量标尺.所谓质量标尺,是指将每个服务类中所有候选服务的所有 QoS 自适应地划分为若干个离散的质量集合,而每个质量标尺代表了一个集合.将每个服务类的候选服务划分为若干个质量标尺后,通过模糊逻辑和 APSO 选择满足全局约束的最优质量标尺,最后选择质量标尺下 QoS 效用函数最大的候选服务作为服务组件,最终在较短的时间内得到一个全局最优或接近全局最优的组合服务.WSDSA 方法包括如下 4 个步骤:

步骤 1. 对质量标尺初始化.对每一个服务类  $S_j$  中的 QoS 属性进行质量标尺初始化,将其初始分解为  $p(p \in N)$  个质量标尺,如图 3 所示,各个质量标尺之间为离散关系,如下所示:

$$Q_{j,k}^{\min} \leq q_{jk}^1 \leq \dots \leq q_{jk}^p \leq Q_{j,k}^{\max}.$$

如果在每次组合服务前给出  $p$  初始值,由于  $p$  值固定,不同的组合服务实例所选择的服务类和候选服务也不尽相同,将无法根据用户偏好进行服务的动态组合.因此,本文根据监测当前的用户偏好值、适应度和时间花费之间的关系状态,提出了基于模糊逻辑<sup>[15]</sup>的质量标尺数量自适应调整方法 AAM 用于动态调整  $p$  的值,使其具有自适应性,从而获得满足用户偏好的最优分解数量.

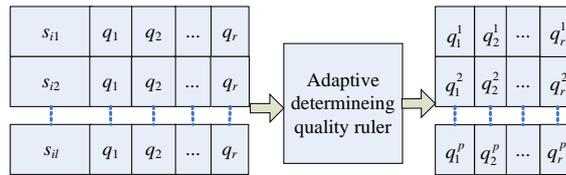


Fig.3 Quality ruler

图 3 质量标尺

AAM 方法可以根据用户偏好(即将用户上下文信息与 QoS 进行映射,更多信息可参考文献[16])和服务选择运行环境动态调整质量标尺数量.该方法有两个输入变量和一个输出变量.两个输入变量分别为质量标尺数量  $p$  和质量标尺性能  $Fp$ ,一个输出变量为质量标尺数量  $p$  的变化比例  $cp$ .其中, $Fp$  定义为

$$Fp = \frac{Fv_{\max} - Fv}{Fv_{\max} - Fv_{\min}} \tag{3}$$

且

$$Fv = \eta \cdot \frac{CF}{CT}, CF = \frac{Fitness_{\max} - Fitness}{Fitness_{\max} - Fitness_{\min}}, CT = \frac{Time_{\max} - Time}{Time_{\max} - Time_{\min}}$$

上式中, $Fv$  为当前质量标尺性能, $Fv_{\max}$  和  $Fv_{\min}$  分别为目前的质量标尺最大性能和最小性能, $\eta \in (0,1)$  为用户个性化需求的个人偏好(可通过文献[16]的方法动态获得并进行归一化处理), $CF$  为 WSDSA 方法中 APSO(第 2.2 节)的适应度函数值的归一化值, $CT$  为所用时间花费(不包括 WSDSA 方法中步骤 4 的时间花费)的归一化值, $Fitness$  为当前质量标尺性能, $Fitness_{\max}$  和  $Fitness_{\min}$  分别为目前质量标尺的最大性能和最小性能, $Time$  为当前时间花费, $Time_{\max}$  和  $Time_{\min}$  分别为目前的最大时间花费和最小时间花费.

AAM 方法首先对两个输入变量  $p, Fp$  和输出变量  $cp$  进行模糊化,设其模糊子集都为  $\{L, M, B\}$ ,其相应的语言变量为  $L(\text{little})=\text{小}, M(\text{middle})=\text{中}, B(\text{big})=\text{大}$ .将  $p, Fp$  和  $cp$  的大小量化为不同等级并得到各自相应的论域  $FX, FY$  和  $FZ$ ,使用三角形和梯形隶属函数进行模糊化,如图 4(a)~图 4(c)所示,分别为两个输入变量  $Fp, p$  和输出变量  $cp$  的隶属度函数.为了防止  $p$  的无限调整,可以设置限定条件,如当  $\sum_t^{t+ss} 1_{\{Fp \geq AK\}} > T$  时(其中, $AK$  为用户上下文调整阈值, $ss=1, \dots, itmax, t$  为 APSO 的当前迭代次数, $ss$  为迭代间隔, $itmax$  为迭代次数, $T=1, \dots, ss$ ),调整停止.

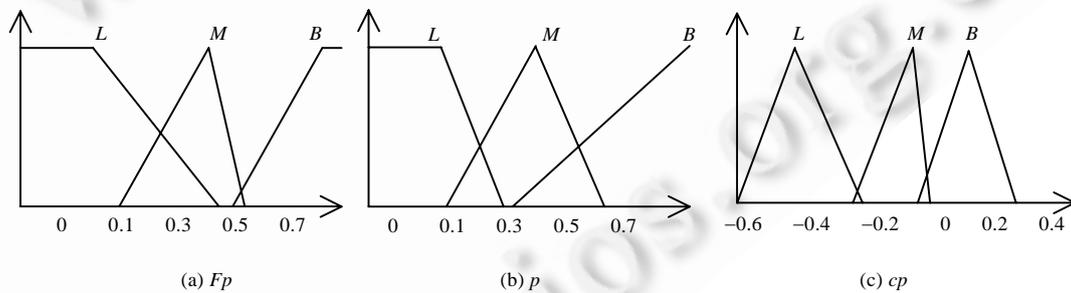


Fig.4 Membership functions

图 4 隶属度函数

本文选取 IF-THEN 的形式,共使用 9 个模糊规则,其曲面投影如图 5 所示,使用该规则,通过模糊推理得到输出模糊变量  $cp$ .由于输出变量是模糊量,而输出结果要求为精确量,本文使用重心法<sup>[15]</sup>对  $cp$  进行解模糊,得到精确的输出结果.

在得出  $cp$  值后,新的质量标尺数量  $p$  的值可按下式进行自适应的调整:

$$p_{\text{new}} = p_{\text{old}} \times (1 + cp) \tag{4}$$

其中,  $p.new \in N, 1 \leq p.new \leq \frac{l}{m}$  (详细分析在第 3.1 节给出),  $l$  为候选服务数量,  $m$  为全局 QoS 约束数量.

通过使用模糊逻辑, 质量标尺数量  $p$  可以根据具体组合服务的 QoS 属性通过 AAM 自适应地调整, 以达到满足用户偏好的服务动态组合目的和用户最佳体验效果.

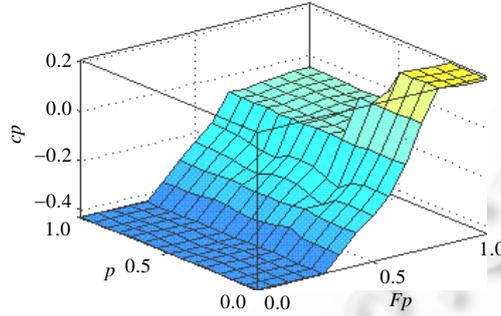


Fig.5 Fuzzy rules  
图 5 模糊规则

步骤 2. 计算质量标尺权重. 对每个质量标尺的权重  $b_{jk}$  进行计算,  $b_{jk}$  代表在该服务类中质量标尺内的候选服务被选中的概率, 计算公式如下:

$$b_{jk}^z = \frac{U(q_{jk}^z) \cdot g(q_{jk}^z)}{U_{\max} \cdot l}, 1 \leq z \leq p.$$

上式中,  $b_{jk}^z$  表示第  $z$  个质量标尺的权重,  $g(q_{jk}^z)$  表示服务类  $S_j$  中属于该质量标尺内的候选服务的数量,  $l$  表示该服务类中候选服务的数量,  $U(q_{jk}^z)$  表示该服务类中、该质量标尺内的所有候选服务的 QoS 效用函数的最大值,  $U_{\max}$  表示该服务类中所有候选服务的 QoS 效用函数的最大值.

步骤 3. 求解最优质量标尺. 由于不同服务类的 QoS 属性取值范围不尽相同, 如果对全局约束直接划分, 将会导致某些服务类的 QoS 属性价值丢失, 以至于无法选择到满足用户需求的组合服务. 我们提出一种自适应粒子群优化算法 (APSO) 用于求解最优质量标尺 (即最优局部约束, 也可称为最优粒子), 更详细的介绍将在第 2 节给出.

步骤 4. 通过局部最优选择最优候选服务. 在局部最优选择中, 如果仍然使用公式 (1) 中候选服务的 QoS 效用函数进行选择, 得到的服务组件将会偏离全局最优而导致局部最优. 原因是, 公式 (1) 中每个属性的分母均是该服务类中最大值与最小值的差距, 导致其更偏重于局部属性. 因此, 需要一个新的 QoS 效用函数使其偏重于全局属性, 而下式中的 QoS 效用函数恰恰能够满足这一要求:

$$U(s_{ji}) = \sum_{k=1}^r \frac{Q_{j,k}^{\max} - q_k(s)}{Q_k^{\max} - Q_k^{\min}} \cdot w_k \left( \text{if } Q_k^{\max} - Q_k^{\min} = 0, U(s_{ji}) = \sum_{k=1}^r 1 \cdot w_k \right) \quad (5)$$

上式中,  $w_k \in R^+$  代表用户偏好, 并且  $\sum_{k=1}^r w_k = 1$ .

通过上述 4 个步骤, 在获得每个服务类中的最优质量标尺后, 对偏离最优质量标尺的候选服务给予丢弃, 根据用户偏好和 QoS 效用函数选择最优的候选服务进行服务组合. 通过运行 WSDSA 方法, 能够在较低的时间花费内, 得到全局最优或接近全局最优的组合服务, 第 3 节给出详细的分析和验证.

## 2 全局 QoS 最优分解

在本文第 1.2 节给出的 WSDSA 方法的步骤 3 中, 需要求解最优质量标尺, 该问题是带约束的多目标优化问题. 常用的方法对于规模较大的问题求解时间开销太大, 不能在合理的时间内求得一个最优的解决方案, 尤其是

对超大规模企业应用和具有成千上万开放服务的基础设施而言,组合服务的响应时间将是无法忍受的.为此,本文提出了一种自适应粒子群优化算法,用于解决最优质量标尺.

本节主要介绍 APSO 算法及其应用(寻找到的最优粒子,即是服务类  $S_j$  中选择满足全局 QoS 约束分解的最优质量标尺).

## 2.1 PSO算法

与遗传算法类似,PSO(particle swarm optimization)<sup>[17]</sup>也是基于群体与适应度.粒子群中的个体,即粒子代表问题的一个可能解,每个粒子具有位置和速度两个特征,粒子位置对应的目标函数值即为该粒子的适应度,通过适应度来衡量粒子的优劣.算法首先初始化一群随机粒子,然后通过迭代找到最优解,在每一次迭代中,粒子通过跟踪两个“极值”来更新自己:一个是粒子本身所找到的最优解,称为局部最优  $p_{best}$ ;另一个是目前整个粒子群发现的最优解,称为全局最优  $p_{gbest}$ .粒子在找到上述两个最优解后,根据下面两个公式来更新自己的速度与位置<sup>[18]</sup>:

$$v_{id}^{t+1} = w \times v_{id}^t + c_1 \times r_1 \times (p_{best}(t) - x_{id}^t) + c_2 \times r_2 \times (p_{gbest}(t) - x_{id}^t) \quad (6)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (7)$$

其中,  $x_{id}$  代表第  $i$  个粒子在第  $d$  维空间中的位置,  $v_{id}$  代表速度,  $r_1, r_2$  是服从  $U(0,1)$  分布的随机数,  $c_1, c_2$  被称为学习因子(一般地,  $c_1=c_2=2.0$ ),  $w$  是惯性权重(一般在  $0.1\sim 0.9$  之间).

PSO 作为一种并行优化算法,可以用于解决大量非线性、不可微和多峰值的复杂问题优化,并已广泛用于科学和工程领域,如函数优化、模式分类和资源分配等领域<sup>[19]</sup>,但是,这些算法以及大部分应用都是针对连续优化问题的.如何将其应用于离散空间优化问题,特别是组合优化问题,已成为粒子群算法的一个重要的研究方向.近年来,许多研究者提出了解决组合优化问题的许多改进型 PSO 算法,如文献[20]将遗传算法的交叉、变异应用到粒子群优化算法中,并通过对公式(6)、公式(7)重新定义特定的加减等运算来解决组合优化问题,并取得了较好的效果;其他方法,如 HPSO<sup>[21]</sup>,FAPSO<sup>[22]</sup>等也获得了较好的应用效果.因此,本文也使用粒子群优化算法用于最优质量标尺的求解.但所不同的是,本文针对 PSO 应用于组合优化问题时普遍存在的早熟收敛问题<sup>[23]</sup>,提出了一种自适应粒子群优化算法,即 APSO.该算法的自适应性主要体现在:动态地解决在不同收敛阶段的惯性权重取值<sup>[24]</sup>问题并提出了对全局最优进行自适应线性变异的方法.

## 2.2 APSO算法

由于粒子群优化算法结构相对简单,所以运行速度很快.但是,算法容易陷入局部最优,出现早熟收敛现象<sup>[19]</sup>.因此,我们采用文献[24]中的模糊逻辑方法动态地调整在不同收敛阶段的惯性权重  $w$  的取值,使算法自适应地调整全局系数,兼顾了搜索效率和搜索精度.

该方法将当前种群最优性能指标( $NCBPE \in [0,1]$ )和当前的惯性权重  $w \in [0.2,1.1]$  作为输入变量,将惯性权重的变化值( $w\_change \in [-0.12,0.12]$ )作为输出变量,然后使用  $w\_change$  动态调整  $w$  的值:

$$w(t+1) = w(t) \times (1 + w\_change) \quad (8)$$

其中,  $t$  为算法的迭代次数,更详细的信息可参考文献[24].

通过该方法,可以使粒子在进化前期以较大惯性权重值保证算法的全局搜索能力,而在收敛的后期阶段以较小惯性权重加快收敛,从而改善了算法的收敛性.然而对于许多复杂的组合优化问题,试图通过调整一个全局系数(如  $w$ )来提高搜索精度的余地是有限的,而通过与其他算法融合(如遗传算法,量子等)来研究算法中各项参数的方法也都在不同程度上增加了算法的复杂性<sup>[19]</sup>.因此,针对惯性权重在摆脱局部极值上所存在的局限性,本文在公式(8)的基础上,提出一种运用线性变异因子调整全局最优的自适应调整方法.

**定义 1.** 设粒子群中某个粒子在  $t$  时刻的位置为  $x(t)$ ,  $\tilde{p}$  为搜索空间内的任意位置,则粒子收敛定义如下:

$$\lim_{t \rightarrow +\infty} x(t) = \tilde{p}.$$

该定义表明,粒子的收敛是指粒子最终停留在搜索空间内某一固定位置  $\tilde{p}$ <sup>[25]</sup>.然而文献[26]指出,当满足上述定义时,并不能使粒子完全收敛.文献[26]提出,还必须有

$$\lim_{t \rightarrow +\infty} StdDev(x^t) = \lim_{t \rightarrow +\infty} \sqrt{E((x^t)^2) - (E(x^t))^2} = 0.$$

即当 $|p_{gbest} - p_{best}|=0$ 时,粒子才能最终收敛到一点.因此,当停滞发生时,由定义 1 并不能保证粒子收敛,可能存在早熟现象,而只有当 $\lim_{t \rightarrow +\infty} StdDev(x^t) = 0$ 时,才能保证粒子达到最优位置,因此给出定义 2.

**定义 2.** 设粒子群中某个粒子在  $t$  时刻的位置为  $x(t)$ ,  $\tilde{p}$  为搜索空间内的任意位置,则当粒子停滞时粒子收敛定义如下:

$$\lim_{t \rightarrow +\infty} x(t) = \tilde{p} \text{ 且 } \lim_{t \rightarrow +\infty} StdDev(x^t) = 0.$$

因此,为了更进一步提高算法的性能,根据定义 2,本文提出了一种全局最优的自适应线性变异方法.该方法选用  $kn$  作为粒子出现早熟的线性变异因子, $k$  作为变异的触发阈值,对全局最优  $p_{gbest}$  进行随机线性变异.如果

$\lim_{t \rightarrow +\infty} StdDev(x^t) \neq 0$  且  $\sum_{t=1}^{t+s} 1_{\{\lim_{t \rightarrow +\infty} E(x^t) = \tilde{p}\}} \geq k, k > 1, s > 1, k, s \in N$ , 则认为粒子出现早熟,增加线性变异因子为

$$p'_{gbest} = km \times p_{gbest} \tag{9}$$

其中, $km \in [0.1, 0.3]$ .

由公式(6)可知,算法通过自适应调整(包括惯性权重动态调整和全局最优线性变异)后的形式为

$$v_{id}^{t+1} = w(t) \times v_{id}^t + c_1 \times r_1 \times (p_{best}(t) - x_{id}^t) + c_2 \times r_2 \times (p'_{gbest}(t) - x_{id}^t) \tag{10}$$

且

$$p'_{gbest}(t) = \begin{cases} km \times p_{gbest}(t), & \text{if } \sum_{t=1}^{t+s} 1_{\{\lim_{t \rightarrow +\infty} E(x^t) = \tilde{p}\}} \geq k \text{ and } \lim_{t \rightarrow +\infty} StdDev(x^t) \neq 0 \\ p_{gbest}(t), & \text{others} \end{cases}$$

### 2.3 适应度函数设计及编码

假设在服务类  $S_j$  中,  $b_{jk}^z$  为其由第  $z$  个质量标尺的第  $k$  个属性产生的权重,其所需质量标尺为  $q_{jk}^z$ , 所对应的约束为  $C_k$ , 则将质量标尺选择映射到 APSO 的适应度函数为

$$f = \max \sum_{j=1}^n \sum_{k=1}^m \sum_{z=1}^d \ln(p_{jk}^z) x_{jk}^z \tag{11}$$

其中,对  $p_{jk}^z$  为便于计算取对数,适应度函数还需满足以下两个约束条件:

$$\begin{aligned} \forall k: \sum_{j=1}^n \sum_{z=1}^d q_{jk}^z x_{jk}^z &\leq C_k, 1 \leq k \leq m \\ \forall j, k: \sum_{z=1}^d x_{jk}^z &= 1, 1 \leq j \leq n, 1 \leq k \leq m \end{aligned} \tag{12}$$

其中,  $x_{jk}^z \in \{0, 1\}$ .  $x_{jk}^z = 1$  表示在服务类  $S_j$  中质量标尺  $q_{jk}^z$  被选中, 否则,  $x_{jk}^z = 0$ .

设计出适应度函数后,为了让 APSO 找到问题的一个解,首先需要构造一个合适的粒子给问题编码.在本文中,粒子使用一个整数数组来表示,数组中的项目数为组成我们服务的服务类数,数组中的每个元素为整数(小数时应取整)并依次包含一个与服务类相匹配的质量标尺数组项目索引.数组中元素的最大值为质量标尺的数量,最小值为 1.因此,在应用 APSO 求解最优质量标尺问题时,可将粒子的位置编码为  $n$  维数组.将粒子的速度也编码为  $n$  维数组,其中每一个元素的值为介于  $[-V_{max}, +V_{max}]$  的实数值.这样不但整个粒子群拥有全局最优粒子  $p_{gbest}$ , 而且每个粒子还拥有局部最优粒子  $p_{best}$ , 两者均是一个  $n$  维数组.

### 3 性能评价

性能评价的目的是对我们所提出的方法进行验证,本节中的性能评价主要分为有效性分析和仿真对比两部分.其中,仿真对比主要用来与纯粹最优方法<sup>[13]</sup>和混合最优方法<sup>[11]</sup>在时间花费和最优度上进行仿真对比.仿

真对比中不仅使用随机获取的 Web 信息数据,而且还使用真实的 Web 信息数据.

### 3.1 有效性分析

QoS 约束服务组合问题的复杂性由下面 3 个因素决定:服务类  $n$ 、对应服务类中的候选服务  $l$ (假设所有服务类中的候选服务数量均相等)及全局 QoS 约束属性  $m$ .文献[11]指出,基于 QoS 全局约束的服务组合问题是多选择多空间的背包问题.当运用混合整数规划方法(属于纯粹最优方法)求解时,最坏情况下的时间复杂度为指数函数  $N(2^{n-l})$ .对于候选服务较少的服务组合,该方法的时间复杂性是可以接受的;但如果有成千上百的候选服务可提供相同的功能(如图 1 所示),则  $N(2^{n-l})$ 的时间复杂性是无法忍受的,进而导致服务选择的响应时间超出运行时间.

在 WSDSA 中,我们使用模糊逻辑和自适应粒子群优化算法,将 QoS 全局约束分解为局部约束,使全局最优服务选择转化为局部最优服务选择.在局部选择中,通过使用公式(5)的局部效用函数求得其局部最优,相应的时间复杂性为线性函数  $N(l)$ .由于通过并行的局部最优选择不受服务类的影响,因此,该步骤中 WSDSA 的时间复杂度仍为线性函数.当然,WSDSA 的时间复杂度还与 QoS 约束分解部分的时间复杂性有关.在约束分解中,时间复杂度由  $n \cdot m \cdot p$  决定,其中, $n$  是服务类数量, $m$  是全局 QoS 约束属性数量, $p$  为质量标尺数量.可以看出,WSDSA 的时间复杂性不依赖于服务类中的候选服务数量  $l$ ,因此,当  $1 \leq p < l/m (p \in N)$  时,WSDSA 方法的搜索空间较小.

在  $n \cdot m \cdot p$  的搜索空间中,假设一个质量标尺组合实例  $S$ ,经过一次迭代后得到一个新的组合路径.新路径通过增加或减少旧路径的质量标尺而获得;同时,通过核查约束条件来确定新路径是否符合要求.如果新路径符合要求,则将其与旧路径相比较,如果好于后者,则代替旧路径;否则给予保留.经过多次迭代后,通过多次比较,最后得到的路径就是已对比路径中的最优路径.因此,可以通过增加迭代次数,得到所有质量标尺的最优组合路径.

由第 3.3 节中的 APSO 算法可知,假设  $f(1, i)$  为第 1 次迭代后质量标尺路径的适应度函数(适应度函数用于评价组合路径中服务质量的优劣,适应度越大,代表组合路径的服务质量越好),则新路径能够通过公式(7)获得:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}.$$

新的适应度函数  $f(t, i)$  可以通过公式(11)获得,如果  $f(t, i) \geq f(t-1, i)$ ,则保留新路径.如果适应度函数的值  $f(t_m, i) \geq f(t_{m-1}, i)$ ,并且  $t_m \geq t_{m-1}$ ,那么经过多次迭代后得到的组合路径不劣于前者.

对于新路径的产生,通过公式(10)可以保留更多的有利信息(旧路径与当前最好路径在速度方向的比较),则由  $p_{best}(t) \geq p_{best}(t-1), p_{gbest}(t) \geq p_{gbest}(t-1)$ ,更好的质量标尺的路径可以通过公式(7)、公式(10)快速得到.因此,本文提出的 WSDSA 算法是有效的.

### 3.2 仿真对比

为了准确地验证本文提出的方法,我们分别进行了时间花费仿真对比和最优度仿真对比,无论采用哪种仿真对比,都采用相同的数据集、测试用例、全局 QoS 约束及软硬件环境.实验中,我们使用两种数据集:第 1 个数据集是 QWS 真实数据集,该数据集包含了 2 500 个真实 Web 服务(每个服务包含 9 个 QoS 属性),对该数据集更详细的介绍和描述可参考文献[27-29];第 2 个数据集通过随机产生.为了保证对比条件相同,对于 QWS 数据集,两种方法所用测试用例中的服务类数量为 5~25,每个服务类的候选服务数量从 50~500;对于随机生成数据集,服务类数量为 5~100,每个服务类中的候选服务数量从 50~5 000.通过设置顺序组合类型的不同服务类数量  $n$  和候选服务数量  $l$ ,产生不同的测试用例.其中,QoS 属性总数为 5,全局约束数量为 3(从 9 个属性中随机选取).在 WSDSA 方法中,惯性权重初始值  $w=0.8, c_1=2.0, c_2=2.0, kn=0.28, s=7, k=5, km \in (1+2 \cdot \text{rand})/10$ ,迭代次数  $imax=30$ ,粒子种群数量  $Np=15$ .仿真对比中,所有方法分别运行 50 次取平均值.实验环境为:Pentium Dual 2.4GHz,2.0GB RAM,Windows XP,MATLAB 7.6,lpsolve5.5.

#### 3.2.1 时间花费仿真对比

首先讨论 WSDSA 方法与纯粹最优方法及混合最优方法关于时间花费的仿真对比.

图 6 给出了 WSDSA 方法与纯粹最优方法在不同服务类数量下的仿真对比结果.实验中,约定纯粹最优方法的时间花费为  $CT_1$ ,混合最优方法的时间花费为  $CT_2$ ,WSDSA 方法的时间花费为  $CT_3$ .

由仿真结果可以看出,无论是 QWS 真实 Web 数据集,还是随机生成数据集,CT3(平均为 329.2ms)远远小于 CT1(平均为 7 947.3ms)和 CT2(平均为 1 753.4ms),表明 WSDSA 方法在不同服务类数量下的时间性能要优于其他两种方法.原因在于,与纯粹最优方法相比,WSDSA 方法的搜索空间的大小由  $n \cdot m \cdot p$  ( $n$  是服务类数量,  $m$  是全局 QoS 约束属性数量,  $d$  为 QoS 质量标尺数量,  $1 \leq p < l/m$ ) 来决定,相当于候选服务数量  $l$  的增加对搜索空间的增大影响较小(纯粹最优方法的搜索空间为  $n \cdot m \cdot l$ ),因而时间花费较少;与混合最优方法相比,WSDSA 方法是通过群体中粒子之间的协作和信息共享来并行寻找最优解的,相当于从当前全局最优粒子和局部最优粒子获得更新信息,至少保证了粒子可以获得上述粒子相近质量的解,因而避免了大量冗余的搜索,所以,时间花费比混合最优方法更具有优势,不像其他方法,随着服务类数量的增加,时间复杂性增加较快.这与文献[20,30]中对粒子群算法在组合优化中的仿真结果类似.

图 7 给出了 WSDSA 方法与上述两种方法在候选服务数量不同情况下,时间花费的仿真比较结果.实验中,约定纯粹最优方法的时间花费为  $ST1$ ,混合最优方法的时间花费为  $ST2$ ,WSDSA 方法的时间花费为  $ST3$ .

从仿真结果可以看出,随着候选服务的增加, $ST3$ (平均为 348.7ms)仍然要小于  $ST1$ (平均为 4 061.5ms)和  $ST2$ (平均为 469.2ms),表明 WSDSA 方法在不同候选服务数量下的时间性能仍然优于其他两种方法.WSDSA 方法优于纯粹最优方法的原因是,将全局 QoS 约束通过 APSO 算法分解为局部约束,相当于通过局部最优选择满足全局最优的服务组件,从而可以在较小的搜索范围内花费较少的时间获得解.WSDSA 方法优于混合最优方法的原因是,质量标尺  $p$  的数量能够自适应地调整,相当于根据实际候选服务数量以及用户偏好寻找到一个较好的平衡点,从而可以将全局 QoS 约束进行最佳分解,为减少服务选择的时间开销发挥了作用 ( $p_{new} = p_{old} \times (1 + cp)$ ).当然,在候选服务数量较少的情况下,WSDSA 方法的时间优势在候选服务数量较多的情况下,体现得就不明显.

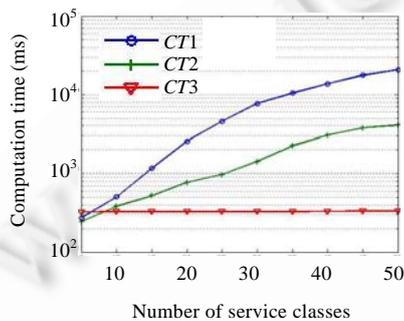


Fig.6 Computation time comparison 1

图 6 时间花费仿真对比 1

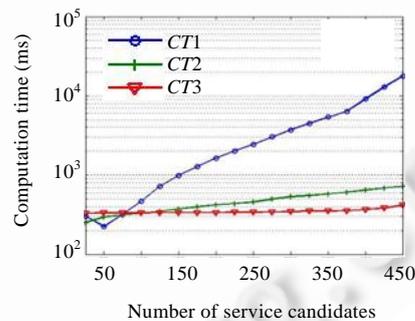


Fig.7 Computation time comparison 2

图 7 时间花费仿真对比 2

### 3.2.2 最优度仿真对比

由于 WSDSA 方法求得的解与混合最优方法类似,均是最优解或接近最优解,因此需要对 WSDSA 方法与纯粹最优方法及混合最优方法进行最优度仿真对比.

实验中为便于比较,约定纯粹最优方法的总效用函数值为  $optim1$ ,混合最优方法的总效用函数值为  $optim2$ ,WSDSA 方法的总效用函数值为  $optim3$ .最优度依次为  $opt1, opt2, opt3$ ,其中,  $opt1 = optim1/optim1$  (纯粹最优方法的最优度为 100%),  $opt2 = optim2/optim1$ ,  $opt3 = optim3/optim1$ .

图 8 和图 9 给出了 WSDSA 方法与纯粹最优及混合最优方法在不同候选服务数量和服务类数量下组合效果的仿真对比结果.

结果表明,WSDSA 方法能够平均得到 98.2% 以上的最优度,几乎接近纯粹最优方法的最优解水平,对于全局 QoS 约束严格要求的组合服务是可以接受的,原因是这种微乎其微的差距在大规模的工业实际应用中几乎察觉不到,这与由网络因素所引起的微小差距产生的影响相当.

从仿真结果还可以看出,WSDSA 方法平均获得 98.29% 的最优度,高于混合最优方法平均获得 97.15% 的最

优越.原因在于,WSDSA 方法分解全局 QoS 约束时使用了自适应粒子群算法,相当于通过种群间的信息共享机制(追随当前搜索到的最优解来寻找全局最优)提高了解的质量,从而可以获得更优解.另外,APSO 中的惯性权重自适应调整和自适应线性变异方法在粒子求解过程中有效地缓解了早熟收敛现象,为 WSDSA 方法获得较高的最优度起到了积极作用.

从用户角度上分析,3 种方法在最优解上是基本一致的,但是由于 WSDSA 方法的时间成本较少,从而使用户的 QoE(quality of experience)<sup>[31]</sup>要明显好于上述两种纯粹最优方法(如图 1 所示的多媒体服务).

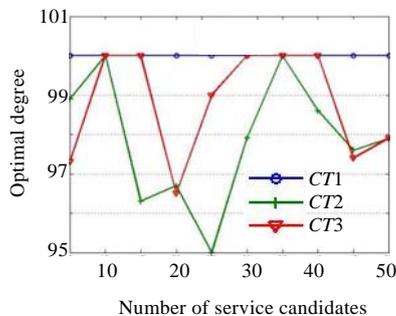


Fig.8 Optimization comparison 1

图 8 最优度仿真对比 1

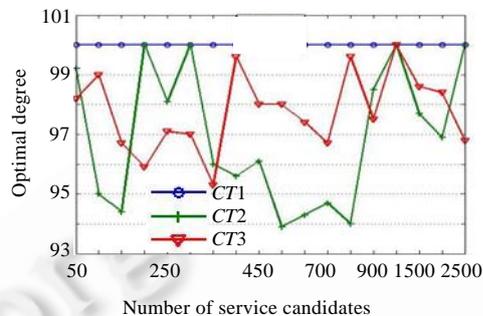


Fig.9 Optimization comparison 2

图 9 最优度仿真对比 2

#### 4 结论及未来工作展望

对于目前的研究工作,研究人员提出的方法在解决 QoS 约束的组合服务选择问题中,普遍存在时间复杂性过高的问题.尤其是当网络中的 Web 服务大量增加时,导致无法在用户要求的时间内获得服务组合,用户的 QoE 变差,服务附加值增长缓慢,甚至下降.本文提出了一种基于全局 QoS 约束自适应分解的 Web 服务动态选择方法,即 WSDSA.WSDSA 首先将每个服务类中的所有候选服务按照全局 QoS 约束动态分解为若干个质量标尺.其中,为了更好地满足用户偏好,我们提出了基于模糊逻辑的约束分解数量调整方法,该方法通过对获得的用户偏好信息及 WSDSA 中的相关信息进行模糊推理,最终得到满足用户偏好的分解数量;其次,求出每个质量标尺的权重;然后,根据质量标尺权重求解出每个服务类中的最优质量标尺.其中,为了能够快速而高效地获得每个服务类的最优标尺以及克服粒子群优化算法求解组合优化问题时存在的早熟收敛现象,我们提出了自适应粒子群优化算法——APSO;最后,将获得的最优质量标尺用于局部约束,进而通过局部最优方法得到满足用户偏好要求的组合服务.性能评价结果表明,WSDSA 在接近纯粹全局最优方法获得的最优解的同时,缩短了计算时间,能够满足对实时性和动态性要求较高的服务选择的需求,具有较好的有效性.

虽然我们的研究在服务选择的自适应性和实时性上具有较大优势,但仍有进一步改进的必要:APSO 在求解最优质量标尺时,初始粒子是随机的,如果在全局 QoS 约束分解过程中考虑粒子的初始值(而非随机赋值),将进一步提高服务选择的实时性;在全局 QoS 约束的自适应分解中,模糊子集数量选取得较少,导致模糊规则相对稀疏,对复杂情况的推理决策能力不强.增加模糊子集的数量将是进一步的研究重点,未来的研究工作还包括全局 QoS 约束分解与分布式服务发现协议的研究相结合.

**致谢** 在此,对提供 QWS 数据集的加拿大圭尔夫大学的 Eyhab Al-Masri 和 Qusay H. Mahmoud 博士表示感谢.

#### References:

- [1] Liu SL, Liu YX, Zhang F, Tang GF, Jing N. A dynamic Web services selection algorithm with QoS global optimal in Web services composition. *Journal of Software*, 2007,18(3):646–56 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/646.htm> [doi: 10.1360/jos180646]

- [2] Yang FC, Su S, Li Z. Hybrid QoS-aware semantic Web service composition strategies. *Science in China (Series F—Information Sciences)*, 2008,51(11):1822–1840. [doi: 10.1007/s11432-008-0159-1]
- [3] Funk C, Schultheis A, Linnhoff-Popien C, Mitic J, Kuhmunch C. Adaptation of composite services in pervasive computing environments. In: *Proc. of the IEEE Int'l Conf. on Pervasive Services*. New York: Institute of Electrical and Electronics Engineers Computer Society, 2007. 242–249. <http://www.icpsconference.org/2007/> [doi: 10.1109/PERSER.2007.4283922]
- [4] Zeng LZ, Benattallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. *IEEE Trans. on Software Engineering*, 2004,30(5):311–327. [doi: 10.1109/TSE.2004.11]
- [5] Zeng LZ, Benattallah B, Dumas M, Kalagnanam J, Sheng QZ. Quality driven Web services composition. In: *Proc. of the 12th Int'l Conf. on World Wide Web*. New York: ACM, 2003. 411–421. <http://www2003.org/> [doi: 10.1145/775152.775211]
- [6] Li Z, Yang FC, Su S. Fuzzy multi-attribute decision making-based algorithm for semantic Web service composition. *Journal of Software*, 2009,20(3):583–596 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3341.htm> [doi: 10.3724/SP.J.1001.2009.03341]
- [7] Wagner M, Kellerer W. Web services selection for distributed composition of multimedia content. In: *Proc. of the 12th ACM Int'l Conf. on Multimedia (ACM Multimedia 2004)*. New York: Association for Computing Machinery, 2004. 104–107. <http://www.mm2004.org/> [doi: 10.1145/1027527.1027546]
- [8] Zhang CW, Su S, Chen JL. Genetic algorithm on Web services selection supporting QoS. *Chinese Journal of Computers*, 2006, 29(7):1029–1037 (in Chinese with English abstract).
- [9] Liu XZ, Huang G, Mei H. Discovering homogeneous Web service community in the user-centric Web environment. *IEEE Trans. on Services Computing*, 2009,2(2):167–181. [doi: 10.1109/TSC.2009.11]
- [10] Zhang CW, Su S, Chen JL. DiGA: Population diversity handling genetic algorithm for QoS-aware Web services selection. *Computer Communications*, 2007,30(5):1082–1090. [doi: 10.1016/j.comcom.2006.11.002]
- [11] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition. In: *Proc. of the 18th Int'l Conf. on World Wide Web*. 2009. <http://www2009.org/> [doi: 10.1145/1526709.1526828]
- [12] Yu T, Zhang Y, Lin KJ. Efficient algorithms for Web services selection with end-to-end QoS constraints. *ACM Trans. on the Web*, 2007,1(1):1–26. [doi: 10.1145/1232722.1232728]
- [13] Ardagna D, Pernici B. Adaptive service composition in flexible processes. *IEEE Trans. on Software Engineering*, 2007,33(6): 369–384. [doi: 10.1109/tse.2007.1011]
- [14] Jang JH, Shin DH, Lee KH. Fast quality driven selection of composite Web services. In: *Proc. of the 4th European Conf. on Web Services (ECOWS 2006)*. Institute of Electrical and Electronics Engineers Computer Society, 2006. 87–96. <http://dblp.uni-trier.de/db/conf/ecows/ecows2006.html> [doi: 10.1109/ECOWS.2006.21]
- [15] Klir GJ, Yuan B. *Fuzzy Sets and Fuzzy Logic Theory and Applications*. Prentice-Hall PTR, 1995.
- [16] Chuang SN, Chan ATS. Dynamic QoS adaptation for mobile middleware. *IEEE Trans. on Software Engineering*, 2008,34(6): 738–752. [doi: 10.1109/TSE.2008.44]
- [17] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proc. of the '95 IEEE Int'l Conf. on Neural Networks*. IEEE, 1995. 1942–1948. <http://www.ourlocal.com/ieee/?c=4057> [doi: 10.1109/ICNN.1995.488968]
- [18] Demarcke P, Rogier H, Goossens R, De Jaeger P. Beamforming in the presence of mutual coupling based on constrained particle swarm optimization. *IEEE Trans. on Antennas and Propagation*, 2009,57(6):1655–1666. [doi: 10.1109/TAP.2009.2019923]
- [19] Hu W, Li ZS. A simpler and more effective particle swarm optimization algorithm. *Journal of Software*, 2007,18(4):861–868 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/861.htm> [doi: 10.1360/jos180861]
- [20] Hu CH, Chen XH, Liang XM. Dynamic services selection algorithm in Web services composition supporting cross-enterprises collaboration. *Journal of Central South University of Technology*, 2009,16(2):269–274. [doi: 10.1007/s11771-009-0046-y]
- [21] AlRashidi MR, El-Hawary ME. Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects. *IEEE Trans. on Power Systems*, 2007,22(4):2030–2038. [doi: 10.1109/TPWRS.2007.907375]
- [22] Bajpai P, Singh SN. Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market. *IEEE Trans. on Power Systems*, 2007,22(4):2152–2160. [doi: 10.1109/TPWRS.2007.907445]
- [23] Zhang CS, Sun JG, Ouyang DT. A self-adaptive discrete particle swarm optimization algorithm. *Acta Electronica Sinica*, 2009, 37(2):299–304 (in Chinese with English abstract).
- [24] Shi YH, Eberhart RC. Fuzzy adaptive particle swarm optimization. In: *Proc. of the Congress on Evolutionary Computation 2001*. Institute of Electrical and Electronics Engineers Inc., 2001. 101–106. <http://bi.snu.ac.kr/cec2001/> [doi: 10.1109/CEC.2001.934377]

- [25] Lü ZS, Hou ZR. Particle swarm optimization with adaptive mutation. *Acta Electronica Sinica*, 2004,32(3):416–420 (in Chinese with English abstract).
- [26] Poli R. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Trans. on Evolutionary Computation*, 2009,13(4):712–721. [doi: 10.1109/TEVC.2008.2011744]
- [27] Al-Masri E, Mahmoud QH. Discovering the best web service. In: *Proc. of the 16th Int'l Conf. on World Wide Web (WWW 2007)*. New York: Association for Computing Machinery, 2007. 1257–1258. <http://www2007.org/> [doi: 10.1145/1242572.1242795]
- [28] Al-Masri E, Mahmoud QH. QoS-Based discovery and ranking of Web services. In: *Proc. of the 16th Int'l Conf. on Computer Communications and Networks 2007 (ICCCN 2007)*. Institute of Electrical and Electronics Engineers Inc., 2007. 529–534. <http://icccn.org/icccn07/> [doi: 10.1109/ICCCN.2007.4317873]
- [29] Al-Masri E, Mahmoud QH. Investigating Web services on the World Wide Web. In: *Proc. of the 17th Int'l Conf. on World Wide Web 2008 (WWW 2008)*. New York: Association for Computing Machinery, 2008. 795–804. <http://www2008.org/> [doi: 10.1145/1367497.1367605]
- [30] Hu CH, Wu M, Liu GP, Xu DZ. Approach to constructing Web service workflow based on business spanning graph. *Journal of Software*, 2007,18(8):1870–1882 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1870.htm> [doi: 10.1360/jos181870]
- [31] Tasaka S, Yoshimi H. Enhancement of QoE in audio-video IP transmission by utilizing tradeoff between spatial and temporal quality for video packet loss. In: *Proc. of the 2008 IEEE Global Telecommunications Conf. (GLOBECOM 2008)*. New York: Institute of Electrical and Electronics Engineers Inc., 2008. 1–6. <http://www.ieee-globecom.org/2008/> [doi: 10.1109/GLOCOM.2008.ECP.279]

#### 附中文参考文献:

- [1] 刘书雷,刘云翔,张帆,唐桂芬,景宁.一种服务聚合中 QoS 感知全局最优服务动态选择算法. *软件学报*,2007,18(3):646–656. <http://www.jos.org.cn/1000-9825/18/646.htm> [doi: 10.1360/jos180646]
- [6] 李祯,杨放春,苏森.基于模糊多属性决策理论的语义 Web 服务组合算法. *软件学报*,2009,20(3):583–596. <http://www.jos.org.cn/1000-9825/3341.htm> [doi: 10.3724/SP.J.1001.2009.03341]
- [8] 张成文,苏森,陈俊亮.基于遗传算法的 QoS 感知的 Web 服务选择. *计算机学报*,2006,29(7):1029–1037.
- [19] 胡旺,李志蜀.一种更简化而高效的粒子群优化算法. *软件学报*,2007,18(4):861–868. <http://www.jos.org.cn/1000-9825/18/861.htm> [doi: 10.1360/jos180861]
- [23] 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. *电子学报*,2009,37(2):299–304.
- [25] 吕振肃,侯志荣.自适应变异的粒子群优化算法. *电子学报*,2004,32(3):416–420.
- [30] 胡春华,吴敏,刘国平,徐德智.一种基于业务生成图的 Web 服务工作流构造方法. *软件学报*,2007,18(8):1870–1882. <http://www.jos.org.cn/1000-9825/18/1870.htm> [doi: 10.1360/jos181870]



王尚广(1982—),男,河南沈丘人,博士,CCF 学生会员,主要研究领域为服务计算,下一代网络安全.



杨放春(1957—),男,博士,教授,博士生导师,主要研究领域为通信软件,网络安全,网络智能化.



孙其博(1975—),男,博士,副教授,主要研究领域为网络智能化,下一代网络安全.