



# 全局优化 Web 服务选择模型及算法研究

学 位 类 型 : 学术型学位

学科(专业学位类别) : 计算机科学与技术

作 者 姓 名 : 康国胜

作 者 学 号 : 0805109

导师姓名及职称 : 刘建勋 教授

实践导师姓名及职称 :

学 院 名 称 : 计算机科学与工程学院

论 文 提 交 日 期 : 2011 年 5 月 31 日

学 位 授 予 单 位 : 湖南科技大学

# Research on Global Optimal Web Service Selection Model and Algorithm

**Type of Degree** \_\_\_\_\_ Academic Degree \_\_\_\_\_

**Discipline** \_\_\_\_\_ Computer Science and Technology \_\_\_\_\_

**Candidate** \_\_\_\_\_ Guosheng Kang \_\_\_\_\_

**Student Number** \_\_\_\_\_ 0805109 \_\_\_\_\_

**Supervisor and Professional Title** \_\_\_\_\_ Prof. Jianxun Liu \_\_\_\_\_

**Practice Mentor and Professional Title** \_\_\_\_\_

**School** \_\_\_\_\_ Computer Science and Engineering \_\_\_\_\_

**Date** \_\_\_\_\_ May 31, 2011 \_\_\_\_\_

**University** \_\_\_\_\_ Hunan University of Science and Technology \_\_\_\_\_

## 学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：

日期： 年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

涉密论文按学校规定处理。

作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

## 摘要

Web 服务作为一种新型的分布式计算模式,以其自包含、良好的封装性、松散耦合、基于标准、高度可集成能力等优点成为当今工业界和学术界的关注热点。随着 Web 服务技术的快速发展,具有相同或相似功能的 Web 服务逐渐增多,在 Web 服务合成中的一个主要问题不再是能否找到满足用户功能需求的服务,而是如何从满足用户功能需求但服务质量各异的服务集中选择出最符合用户需求的服务。国内外许多学者对 Web 服务选择问题展开了大量的研究,并取得了一定的成果,但总体来说还存在许多亟待解决的难题。本文围绕 Web 服务选择问题,主要的研究工作有:

(1) 针对同一时刻存在多个请求同种功能服务情形的 Web 服务选择问题,以专有网络服务环境为研究对象,基于满意度研究该环境下的全局最优 Web 服务选择。本文利用层次分析法计算出各服务请求对各 Web 服务的 QoS 满意度与服务信誉满意度,将二者结合起来,计算出综合满意度。基于综合满意度进行 0-1 整数规划,使服务请求对 Web 服务的全局满意度最大,建立全局优化服务选择模型并设计相应的算法。仿真实验验证了该方法的有效性。

(2) 进一步地,考虑资源的充分利用,保证服务负载平衡。本文从服务请求者和服务提供者双方的角度以及长时间的角度考虑多服务请求情形的全局最优 Web 服务选择。基于带权的 Euclidean 距离计算 Web 服务对各服务请求的距离,即匹配度。基于匹配度进行 0-1 整数规划,使服务请求对 Web 服务的全局匹配度最大,建立全局优化服务选择模型。结合实际提出通用可行的解决多服务请求的 Web 服务全局优化服务选择算法 GOSSMR,使最多的 Web 服务请求需求得到满足但又尽可能不浪费资源,保证资源的合理分配和利用,避免负载问题的发生,提高系统的性能。为提高算法的执行效率,基于 Skyline 的方法对 GOSSMR 算法进行改进,提出 Skyline GOSSMR 改进算法。仿真实验比较两个算法的执行效率和系统的性能。

(3) 研究组合服务中 QoS 全局最优动态 Web 服务选择模型及算法。基于粒子群进化算法,本文设计一种用于解决服务组合中 QoS 全局最优服务动态选择问题的 PSO-GODSS 算法。该算法的主要思想是将问题表示为一个带 QoS 约束的多目标服务组合优化问题,通过理想点的方法将多目标向单目标转化,然后利用粒子群算法的智能优化原理进行算法设计及求解,最终产生一组满足约束条件的优化服务组合流程集。理论分析和实验结果表明该算法的可行性和有效性,且算法执行效率和收敛速度优于以往的多目标遗传算法。相对多目标遗传算法,本文提出的算法规则简单,易于编程实现,收敛速度快,搜索的全局性更好,能够快速找到全局最佳服务组合决策,且在实际应用中

是可行的。

关键词：Web 服务选择；服务组合；层次分析法；欧氏距离；Skyline；粒子群算法

# ABSTRACT

Web service, as a new distributed computing pattern, has become an industry and academic focus with the advantages, such as self-contained, modularization, loose coupling, standardization and highly integration. With the fast development of Web service technology, there is an increasing number of Web services with the same or similar function sharing on the Internet. Currently, the main problem in service composition is not how to discover the Web services which meet the users' functional requirement, but how to select the fitted service from services which meet the function requirement of users but have different service quality to meet the requirement of users. Many researchers did lots of study about Web service selection, and some achievements have been made. However, there are still some problems which require to be solved further. This paper, aiming at the problem of Web service selection, makes the following contributions:

(1) Aiming at the problem of Web service selection with multiple service requests requesting the same or similar functional Web service, global optimal Web service selection model under grid service environment is studied. Using AHP (Analytic Hierarchy Process) to calculate the satisfactions of service's QoS (quality of service) and reputation, and combining the two satisfactions to calculate the synthesized satisfaction, a 0-1 integer programming based on synthesized satisfaction and the corresponding algorithm are achieved, thereby working out the global optimal model for service selection and maximizing the global synthesized satisfaction of service requests to Web services. The effectiveness of our method is verified by simulations.

(2) Further, to make full use of service resources and ensure the load balance, global optimal Web service selection model and algorithm for multiple requests are studied. Euclidean distance with weights to suggest just such a projection based on QoS is employed. A 0-1 integral programming model to maximize the sum of matching degree is performed and consequently, a global optimal service selection policy is developed. The model, together with a universal and feasible optimal service selection algorithm, is employed to propose a global optimal service selection for multiple requests (GOSSMR). Furthermore, to enhance its efficiency, an improved algorithm Skyline GOSSMR is also proposed. Time complexity of the algorithms are analyzed and their feasibility and effectiveness verified by simulations. Finally, the work analyzes the performance of the system as well as the efficiency of the two

algorithms by the way of simulations.

(3) A dynamic Web service selection model and algorithm with QoS global optimal in Web service composition are studied. Based on the algorithm of PSO (Particle Swarm optimization), the PSO-GODSS (global optimal of dynamic Web services selection based on PSO) algorithm is proposed. The basic idea of the algorithm is to transform the original Web service selection problem into a multi-objective services composition optimization with global QoS constraints, which is further transformed into a single-object by using the method of ideal point. Then, the theory of intelligent optimization of PSO is exploited to produce a set of optimal service composition processes with QoS constraints. Theoretical analysis and experimental results indicate the feasibility and efficiency of this algorithm, and the execution efficiency and convergence rate of PSO-GODSS are much better than that of the multi-objective genetic algorithm used in prior work.

**Keywords: Web Service Selection; Service Composition; Analytic Hierarchy Process; Euclidean Distance; Skyline; Particle Swarm Optimization Algorithm**

## 目 录

摘 要.....	i
ABSTRACT.....	iii
<b>第一章 绪论</b>	
1.1 课题研究背景 .....	1
1.2 国内外研究现状 .....	1
1.3 支撑项目 .....	4
1.4 本文内容与结构 .....	4
<b>第二章 Web 服务相关技术与理论基础</b>	
2.1 Web 服务相关技术 .....	5
2.1.1 Web 服务的定义和特点 .....	5
2.1.2 Web 服务体系架构 .....	6
2.1.3 Web 服务关键技术 .....	7
2.2 理论知识 .....	10
2.2.1 层次分析法 .....	10
2.2.2 欧氏距离 .....	13
2.2.3 Skyline .....	13
2.2.4 粒子群算法 .....	14
2.3 本章小结 .....	15
<b>第三章 基于满意度的全局优化服务选择方法</b>	
3.1 基于 QoS 的网格服务选择框架 .....	17
3.2 网格服务的 QoS 计算模型 .....	18
3.3 网格服务全局优化服务选择 .....	19
3.3.1 提供者对各请求者综合满意度的评定 .....	19
3.3.2 全局优化服务选择模型及算法 .....	21
3.3.3 全局优化服务选择模型的扩展 .....	21
3.4 仿真实验及分析 .....	22
3.5 本章小结 .....	23
<b>第四章 基于匹配度的全局优化服务选择方法</b>	
4.1 基于 QoS 的集中式 Web 服务选择框架 .....	25
4.2 Web 服务的 QoS 计算模型 .....	27
4.3 全局优化 Web 服务选择 .....	27
4.3.1 QoS 标准化及偏好的计算 .....	28
4.3.2 匹配度的计算 .....	28

4.3.3 全局优化服务选择模型及算法 .....	29
4.4 仿真实验及分析 .....	31
4.5 本章小结 .....	35
<b>第五章 组合服务中 QoS 全局最优动态服务选择方法</b>	
5.1 服务组合问题描述 .....	37
5.2 PSO-GODSS 算法描述.....	40
5.2.1 MCOOP 形式化描述 .....	40
5.2.2 MCOOP 问题转化为单目标问题 .....	40
5.2.3 PSO-GODOSS 算法设计 .....	41
5.3 仿真实验及分析 .....	42
5.4 本章小结 .....	44
<b>第六章 结论与展望</b>	
6.1 结论 .....	45
6.2 展望 .....	46
<b>参考文献</b> .....	47
<b>致 谢</b> .....	51
<b>附录 A（攻读学位期间发表论文目录）</b> .....	53
<b>附录 B（攻读学位期间所获的奖项）</b> .....	55
<b>附录 C（攻读学位期间参加研究项目）</b> .....	57

# 第一章 绪论

## 1.1 课题研究背景

Web 服务作为一种新型的分布式计算模式，它统一使用 Internet 协议布置和调用，来自不同服务提供商的服务可被集成以提供复杂的组合服务。Web 服务及相关技术近年来得到迅速的发展，在电子商务、企业应用集成等领域扮演着越来越重要的角色。随着 Web 服务相关技术标准的不断提出和完善，支持 Web 服务及其流程的企业应用软件平台逐渐成熟<sup>[1]</sup>。Web 服务也越来越广泛用于各项事务的处理，例如：日程安排、旅游管理、个人事务管理、电子商务等应用<sup>[2-3]</sup>。

随着 Web 服务技术的日益成熟，越来越多稳定易用的 Web 服务共享在网络上。在 Web 服务合成中的一个主要问题不再是能否找到需要的服务，而是如何从满足用户功能需求，但服务质量各异(Quality of Service, QoS)<sup>[4-5]</sup>的服务集中选择出最符合用户需求的服务，即面向服务质量的 Web 服务选择问题成为目前 Web 服务研究领域的热点<sup>[6-8]</sup>。在 Web 环境下，同一时刻可能存在多个请求同种功能服务的请求，它们对所提供服务的 QoS 可能不尽相同。在这种大量相似服务共存竞争的环境下，如何采取适当的全局优化分配策略，实现 Web 服务和请求之间的供需配对，使尽量多的请求需求得到满足，保证资源的合理利用成为 Web 服务选择研究中一个新的难题。

另一方面，单个的 Web 服务能够提供的功能有限，为更加充分利用共享的 Web 服务，有必要将共享的 Web 服务组合起来，提供更为强大的服务功能，加快系统开发的速度，快速满足用户需求<sup>[9]</sup>。因此，如何将若干个现存的 Web 服务整合起来以形成新的、满足用户需求的、增值的大粒度流程服务已成为新的应用需求和研究热点。服务组合流程模型由多个抽象服务结点组成，每个结点只包含功能需求描述，不指定具体的服务调用实例。在互联网环境中，存在多个满足相同功能需求而具有不同 QoS 参数(如执行时间、费用、可靠性等)的 Web 服务，如何从中选择满足各服务结点功能需求的具体服务，形成一个可执行的服务路径来完成用户的需求成为服务组合中的关键问题，本文称该问题为组合服务中的动态 Web 服务选择问题。

本文将针对以上提出的多服务请求情形的 Web 服务选择和组合服务中的动态 Web 服务选择两个问题展开研究，研究这两种情形下的全局优化 Web 服务选择模型及算法。

## 1.2 国内外研究现状

在基于 Web 服务的动态协作中，一方面服务使用方能够在更大范围内灵活选择服

务, 达到业务目标最佳的效果; 而另一方面服务提供方则能够在更大范围内服务更多客户, 获得更多利润<sup>[10]</sup>。随着网络上服务数量和种类的增加, 针对同一种服务需求可供选择的服务越来越多, 用户自然希望通过比较服务质量来选择 Web 服务。因此 Web 服务选择问题成为当前 Web 服务研究的一个热点。

2003 年, Ran<sup>[4]</sup>提出将 QoS 融入到 Web 服务发现, 在服务发现中考虑用户的功能需求和非功能需求。之后, 基于 QoS 的 Web 服务研究成为广泛的研究领域, 到目前为止, 已取得一些研究成果。Zeng<sup>[11]</sup>等人提出考虑多个标准(例如: 服务执行费用、服务执行时间和可用性等)进行服务组合, 并考虑用户的需求偏好和全局约束(例如: 费用预算等)。该文献基于 QoS 加权的方法, 提出了计算 Web 服务综合 QoS 的方法和服务执行流程的 QoS。组合服务中服务选择形式化为一个优化问题, 可采用线性规划模型进行解决。在此基础上, Zeng<sup>[12]</sup>等人进一步提出为组合服务提供一个中间件平台, 采用 QoS 属性的效用函数, 以此优化用户的满意度。基于整数规划, 给出了局部(任务级)服务选择和为服务进行全局任务分配的方法。然而该方法算法的时间效率低, 需要进一步改进。另外, 该文献假设 QoS 数据已经获得, 未讨论 QoS 的如何获得以及方法在实际应用中的可行性。因此, Liu、Anne、Ngu 和 Zeng<sup>[13]</sup>又提出 QoS 的计算以及动态 Web 服务选择的策略。通过对 Web 服务的 QoS 进行排序, 为用户选择 QoS 最佳的 Web 服务。Maximilien 和 Singh<sup>[14]</sup>提出为自治的 Web 服务选择提供一个基于代理的体系架构, 使得用户可以对服务进行评价。在此基础上, 文献[15]中进一步考虑服务的信誉度来进行服务选择。然后, 为获得服务的质量和可信性, Maximilien 和 Singh<sup>[16]</sup>为服务动态选择提供一个代理框架和 QoS 本体。通过用户的参与合作, 来决定服务的质量和可信性。Yu 和 Lin<sup>[17-18]</sup>提出端对端 QoS 约束的服务选择模型, 最大化其具体的效用函数。将服务选择问题模型化为一个多维选择的背包问题, 并基于多维多选择的 0-1 背包算法解决该模型。然而, 该算法还不足以满足一些实时的环境。Danilo 和 Barbara<sup>[19]</sup>将服务组合问题模型化为一个混合整数线性规划问题, 使得全局和局部约束能够同时具体化, 优化每个可能的服务组合执行路径。但是, 该算法不适合动态和实时的需求情况。为解决该问题, 基于局部选择策略, 结合全局优化, Alrifai 和 Risse<sup>[20]</sup>提出两阶段的解决办法。首先, 使用混合整数规划(MIP), 将全局 QoS 约束分解成局部约束; 然后, 采用分布式的局部选择找到满足局部约束的最好的 Web 服务。同样, 为提高算法的执行效率, Tang<sup>[21]</sup>等人提出了一个启发式服务组合方法, 旨在过滤掉 QoS 较差的服务, 最后求得一个接近最优解的组合服务。文献[22-23]研究 Web 服务推荐, 利用用户及服务之间的相似度对服务的 QoS 值进行预测, 以供服务推荐使用, 但并没有提出具体的服务选择方法。文献[24-25]讨论多个服务组合或执行流程的调度问题, 能够在一定程度上改善服务的性能。

从上面的关于 Web 服务选择问题的研究中可以看出, 有以下问题尚待研究:

1. 现有的 Web 服务选择方法主要实现单个服务请求者的最优服务选择, 很少关注在某一时刻或很小的一个时段内整个服务集群或者服务网络的全局最优问题。事实上, 某一个时刻或很短一个时段内, 可能存在多个请求同种功能服务(包括原子服务和组合服务)的服务请求, 它们对所提供服务的 QoS 要求可能不尽相同, 所以采用已有的基于单个服务请求的最优选择模式, 由于请求同种功能服务的多个服务请求之间存在竞争或冲突, 可能导致某些满足服务请求 QoS 要求却未被选择的服务一直闲置。一些原本具有较高 QoS 质量的服务服务于较低 QoS 需求的服务请求, 而一些需要较高 QoS 质量的服务请求因无法满足需求, 反而暂时得不到服务, 需要长时间等待, 造成资源浪费和全局资源分配不均衡, 致使网络的整体服务质量降低。因此, 应采取适当的全局优化分配策略, 实现服务提供者和服务请求者之间供求配对, 使所有的请求者整体利益最大化, 同时也保证资源的最大化使用。从提供商的角度看不盲目地接受请求, 需按照自身的服务能力分配给最合适的请求者; 从请求者的角度看不盲目地选择最优的服务, 而是使所有请求者在整体上达到最大化的满足, 从而满足双方利益最大化的需求。另外, 从长时间的角度来看, 某一时刻所有的请求者不一定都要选择服务质量最优的服务, 而是选择使所有请求者全局最佳匹配的服务, 使得资源得以合理利用, 从而从长时间上看可使得更多的服务请求得到满足。

2. 在前面介绍的国内外研究当中, 其中文献[11,13,18-21,26-27]研究服务组合的优化问题。目前关于 Web 服务选择的研究, 大多是基于 QoS 局部最优的原则<sup>[13,28]</sup>, 即对于满足组合流程中单个服务结点功能需求的一组服务, 根据 Web 服务的各个 QoS 参数信息进行加权排序, 并以此为依据分别为各个服务结点选择加权和最大的服务来执行其功能。局部最优服务选择的方法中为各个服务结点独立地选择服务, 没考虑服务之间的相互关联性, 不能解决服务组合的 QoS 全局最优问题。对于 QoS 全局最优问题, 目前的研究工作不多<sup>[11,17,19-20,29-30]</sup>, 它们的主要思想主要是通过把 Web 服务组合流程的各个 QoS 参数计算公式线性加权转化为一个单目标, 利用线性规划基本原理来解决服务组合中 QoS 全局最优动态服务选择问题, 或者研究求解该模型的各种算法。总之, 它们尚存在以下问题: (1) 把组合流程中的各个 QoS 参数线性化为一个目标函数, 优化结果单一, 不能解决多目标优化问题; (2) 由于加权和的结果不仅对权重敏感, 而且用户对问题有一定的认识, 使得加权的方法在实际应用中不可行; (3) 产生的优化结果为单个解, 用户没有选择的余地; (4) 算法的时间复杂大, 随着候选服务数量的增加, 整个服务组合的性能将会受到影响。所以, 研究 Web 服务组合中基于服务质量的服务选择问题的智能优化算法及其实现显得尤为重要。文献[31]针对现有服务组合算法的不足, 基于遗传算法提出一种解决服务组合中动态选择 QoS 全局最优问题的算法。它将服务动态选择全局最优问题转化为一个带 QoS 约束的多目标组合优化问题, 利用多目标遗传算

法的智能优化原理，同时优化多个目标函数，最终产生一组满足约束条件的 Pareto 优化服务组合流程集，解决了传统方法中难以解决的问题，但算法的执行效率还有待提高。

### 1.3 支撑项目

本文研究得到国家自然科学基金项目“基于行为及其可信扩展的可信服务组合方法研究”(项目编号: 90818004)、教育部新世纪人才项目“基于 SOA 的企业间协同服务技术研究”(项目编号: NCET-10-0140)以及湖南省教育厅资助项目“基于信誉策略的可信 Web 服务合成模型与方法研究”(项目编号: 09k085)的资助。

### 1.4 本文内容与结构

本文主要研究内容的逻辑结构图如图 1.1 描述所示。第三章和第四章分别从不同的角度研究了多服务请求情形的全局优化 Web 服务选择问题。而第五章研究的是组合服务中的 QoS 全局最优动态服务选择问题。具体而言，第三章以专有网格服务环境为研究对象，提出了基于 QoS 的网格服务选择框架。从单一时刻的优化角度出发，通过使用层次分析法，使得在某一时刻所有请求者的满意度达到最大，建立了全局优化服务选择模型及算法。第四章从长时间的角度考虑资源的充分利用，依托实验室开发的 Web 服务超市平台，建立了基于 QoS 的服务选择框架。保证服务不负载的情况下，通过为所有服务请求选择全局匹配度最大的服务，建立了全局优化服务选择模型及算法。使得从长时间的角度看，服务资源能够得到合理的利用。第五章针对组合服务中多目标多约束的组合服务优化选择问题，通过理想点的方法将多目标向单目标转化，利用粒子群优化算法进行算法设计和求解，最终产生一组满足约束条件的优化服务组合流程集。

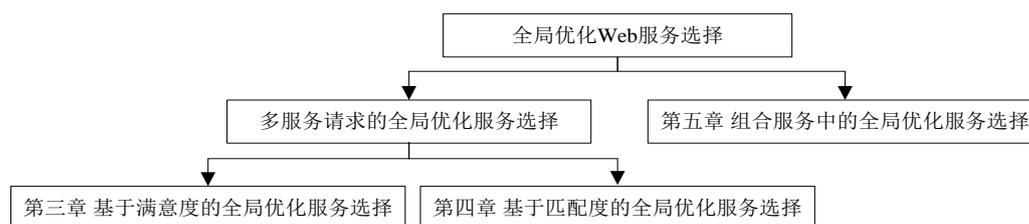


图 1.1 本文研究内容的逻辑结构图

Fig. 1.1 Logic structure diagram of the main research contents

本文的组织结构如下：论文的第一章是绪论；第二章介绍 Web 服务相关技术和本文的研究基础；第三章提出基于满意度的全局优化 Web 服务选择模型及算法；第四提出基于匹配度的全局优化 Web 服务选择模型及算法；第五章研究组合服务中 QoS 全局最优动态 Web 服务选择模型及算法；第六章总结了本文的研究成果和创新点，并提出了下一步的研究计划。

## 第二章 Web 服务相关技术与理论基础

Web 服务作为一种新型的分布式构件模型，在电子商务、企业集成等领域扮演着重要的角色，并不断影响现代企业应用的开发与部署，它的不断成熟和发展为服务计算技术的发展与应用提供最佳的支撑技术<sup>[32]</sup>。本章将介绍 Web 的相关技术以及本文研究工作的理论知识，为后续章节中模型及算法的介绍奠定基础。

### 2.1 Web 服务相关技术

服务计算是在面向动态、多变、复杂的互联网环境中而提出的一门以 Web 服务、面向服务的体系架构为基础支撑技术，以服务组合为主要软件开发方法，以面向服务的软件分析和设计原则为基本理念的新的计算科学<sup>[32-33]</sup>。在服务计算的技术体系中，服务是最重要的核心概念。这里所提到的服务是指基于网络环境的具有自适应、自描述、模块化和良好互操作能力等特点的软件实体，而 Web 服务是服务计算中所强调的服务概念中的一种具体表现形式和功能载体，它是一种基于 Web 环境的具有自适应、自描述、模块化并具有良好的互操作能力的应用程序<sup>[29, 34]</sup>。

本节将介绍 Web 服务的相关技术，包括 Web 服务的定义和特点、Web 服务的体系架构和 Web 服务的关键技术。

#### 2.1.1 Web 服务的定义和特点

Web 服务是一种面向服务架构的技术，通过标准的 Web 协议提供服务，目的是保证不同平台的应用服务可以互操作。然而，不同的组织对于 Web 服务的概念和内涵有着不同的理解和认识<sup>[32]</sup>。

IBM 认为，Web 服务是用 XML 描述的一组可通过 XML 进行消息传递的操作，这些操作可由网络访问完成目标任务。服务描述提供与该服务交互时所必需的所有细节，包括消息格式、传输协议和位置。

Microsoft 认为，Web 服务是一个向其它应用提供数据和服务的应用逻辑单元。应用程序通过 Web 协议和数据格式访问 Web 服务，如 HTTP、XML 和 SOAP，而无需关心每个 Web 服务的具体实现。

SUN 认为，Web 服务是软件构件，这类构件具有被发现、可重用和再组合的特性，用于解决用户的问题要求。

W3C 认为，Web 服务是一个支持互操作的，机器能通过网络进行交互的系统。它采用 WSDL 描述服务接口，采用 SOAP 消息通过 HTTP 协议实现服务之间、服务和客

户之间的通信。

从以上的定义可以看出，Web 服务最大的特点在于它的互操作和重用性。互操作能力使得 Web 服务成为一种最有吸引力的、用于异质异构应用和系统无缝集成的黏合剂，而重用性使得 Web 服务成为软件服务化的最佳载体。与其他的软件实体相比，Web 服务具有如下特性：

可描述：可以通过一种服务描述语言进行自我描述；

可发布：可以在注册中心注册其描述信息并进行发布；

可查找：通过向服务注册中心发送查询请求可以找到满足查询条件的服务，获取服务的调用信息；

可绑定：通过服务的描述信息可以生成可调用的服务实例或服务代理；

可调用：使用服务描述信息中的访问细节可以实现服务的远程调用；

可组合：可以与其他服务组合在一起形成新的大粒度增值服务。

### 2.1.2 Web 服务体系架构

Web 服务的体系架构包括服务请求者、服务提供者和服务注册中心，这三种角色通过发布、查找和绑定这三个操作进行交互<sup>[35]</sup>，这些角色都围绕 Web 服务本身和 Web 服务描述展开。如图 2.1 所示为 Web 服务的体系架构图。

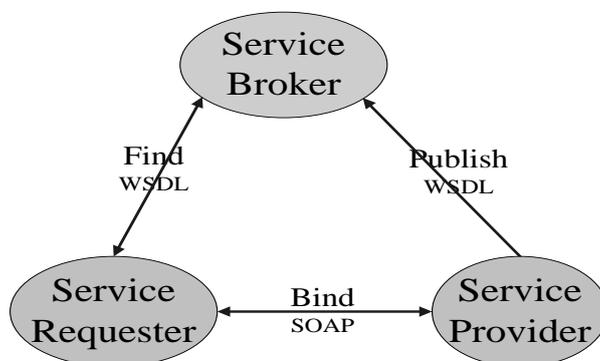


图 2.1 Web 服务的体系架构

Fig. 2.1 Web service architecture

Web 服务体系架构中三种角色的作用分别如下：

**服务提供者(Service Provider):** Web 服务的拥有者，采用 WSDL 文档描述 Web 服务，通过注册方式将 Web 服务的描述信息发布到服务注册中心，供消费者查询和调用服务。

**服务请求者(Service Requester):** Web 服务的调用者，也称为消费者，在服务注册中心中查找所需要的 Web 服务，从而获得服务提供者的地址和 Web 服务的描述信息，调用服务应用程序。从体系结构上看，它是查询和调用 Web 服务的客户端应用程序。

**服务注册中心(Service Broker):** 服务发现的支持者，提供 Web 服务注册功能，存储

服务注册的 Web 服务描述信息,服务请求者可以通过服务注册中心直接查询所需的 Web 服务。

Web 服务体系架构中包含的三种操作分别如下:

**发布(Publish):** 服务提供者将 Web 服务描述信息发布到服务注册中心,供服务请求者访问。

**查找(Find):** 服务请求者在服务注册中心查找所需的 Web 服务。

**绑定(Bind):** 服务请求者在服务注册中心查找到所需的 Web 服务后,根据服务描述信息绑定 Web 服务,实现 Web 服务调用。

### 2.1.3 Web 服务关键技术

Web 服务系列标准的建立促进了 Web 服务的迅速发展,本小节主要介绍三大基础标准,即简单对象访问协议(Simple Object Access Protocol, SOAP)、Web 服务描述语言(Web Service Description Language, WSDL)和服务注册与发现协议(Universal Description, Discovery and Integration, UDDI)<sup>[36]</sup>。图 2.2 展示了一般 Web 服务体系架构中各种标准所扮演的角色。

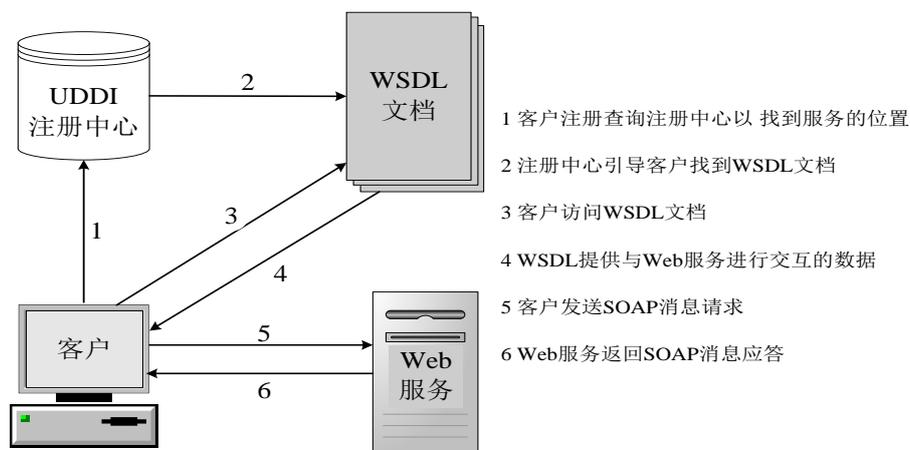


图 2.2 Web 服务交互中的 SOAP、UDDI 和 WSDL

Fig. 2.2 Role of all standards in Web service architecture

#### 1. SOAP

SOAP 是一个基于 XML 的可扩展消息信封格式,需同时绑定一个传输协议<sup>[37]</sup>。这个协议通常是 HTTP 或 HTTPS,但也可能是 SMTP 或 XMPP,它是一种在分布式环境下交换结构化信息的轻量级协议,由 W3C 首次发布。目前的最新版本为 SOAP 1.2,于 2001 年 7 月被 W3C 提升为一个推荐标准。SOAP 这种无状态的单向消息交换协议,其设计目的是简单性和可扩展性,主要由以下 4 部分组成。

**SOAP 信封(SOAP Envelope):** 它包含消息头(Header)和消息主体(Body)两个主要部

分，每一个部分都可以进一步分块。其中消息头是可选的，而消息主体则必须存在。这两部分并没有显示的功能定义，一般而言 SOAP 消息头用来存放与应用无关的信息，而 SOAP 消息主体包含应用相关的数据信息，可以报告消息处理过程中出现的错误。

**SOAP 编码规则(Encoding Rules):** 定义一个数据的编码机制，通过这样一个编码机制来说明应用程序中需要使用的数据类型，它遵循 XML 模式规范的结构和数据类型定义，包括简单类型和复杂类型。

**SOAP RPC 表示(RPC Representation):** 定义如何表示远程的调用和消息返回。

**SOAP 绑定(SOAP Binding):** 定义节点间采用何种底层传输协议来实现 SOAP 消息的交换。

## 2. WSDL

WSDL 是一种用来描述 Web 服务并说明如何与该 Web 服务通信的 XML 语言<sup>[36]</sup>，它由 Ariba、Intel、IBM 和 Microsoft 共同提出。目前最新版本 WSDL 2.0 已被提升为一个推荐标准。

WSDL 从两个级别来描述 Web 服务：抽象级别和具体级别。在抽象级别，WSDL 通过发送/接受的消息来描述一个 Web 服务，其中消息采用 XML Schema 来描述。操作(Operation)通过消息交换模式将消息关联在一起；消息交换模式制定发送/接受消息的序列、多重性以及逻辑上的发送者和接受者。接口(Interface)以独立于传输协议和交换格式的方式将一组操作组织在一起。在具体级别，绑定(Binding)为接口指定传输协议和交换格式；端点(Endpoint)把网络地址和绑定关联在一起；服务针对同一通用接口的端点组织在一起。

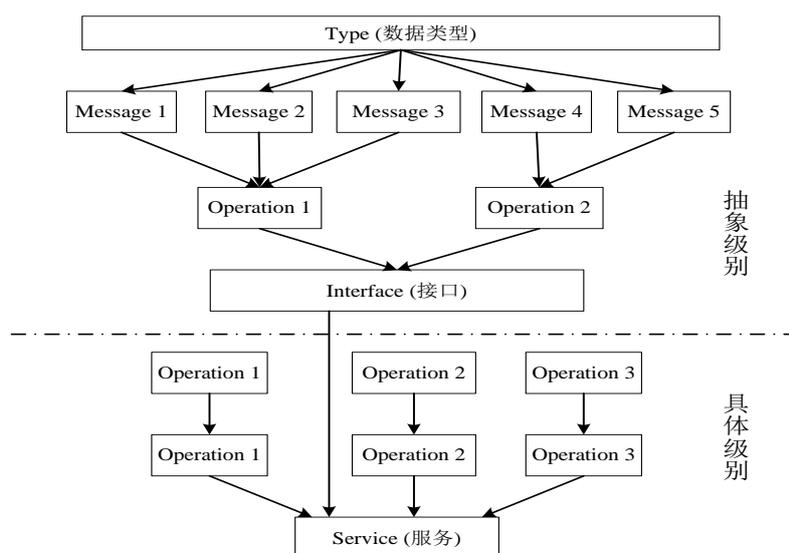


图 2.3 WSDL 文档元素

Fig. 2.3 WSDL document elements

一个 WSDL 2.0 文档包含以下元素：

(1) 类型：<type>元素定义 Web 服务中可使用的数据类型。为达到最大限度的平台无关性，通常使用 XML Schema 语法来定义数据类型和数据结构。

(2) 消息：消息应用组件(Message Reference Component)包括消息引用(Message Reference)、指向(Direction)和消息(Message)三个元素。其中消息引用指定该消息的消息模式；指向用于指明它是接受还是发送消息；消息元素中包含的是用<types>元素定义的类型信息表示的具体消息内容。当使用 SOAP 绑定时，一个 WSDL 消息元素将对应于一个 SOAP 消息的主体部分，这一映射过程是自动进行的。

(3) 操作：<operation>元素描述一组消息的集合。消息交换模式决定操作中各消息的先后顺序以及多重性。可选的消息交换模式有：In-Only、Robust In-Only、In-Out、In-Multi-Out、Out-Only、Robust Out-Only、Out-In、Asynchronous Out-In、Out-Multi-In。

(4) 接口：<interface>是 Web 服务的抽象描述，既不包含服务的具体地址也不包含服务所绑定的具体传输协议。接口具有继承性，这样可以方便地重用已有接口。它替代 WSDL 1.1 中定义的<portType>元素。

(5) 绑定：<binding>元素描述给定端口的具体消息格式和传输协议。绑定可以是通用的，也可以指定具体的接口；可以为整个接口或接口中的一个具体操作定义一个绑定。WSDL 2.0 规范中定义针对 HTTP、SOAP 和 MIME 的绑定。

(6) 端口：<endpoint>元素把网络地址和一个具体的绑定关联在一起。它替代 WSDL 1.1 中定义的<port>元素。

(7) 服务：<service>元素把指定接口相关的端点组合在一起。

下面是一 Web 服务的 WSDL 文件片段：

```
<interface name="PriceInterface">
  <operation name="getPrice" pattern="http://www.w3org/ns/wsdl/in-out">
    <input messageLabel="getPriceRequest" element="tns: request"/>
    <output messageLabel="getPriceResponse" element="tns: response">
  </operation>
</interface>
<binding name="PriceInterfaceHttpBinding"
  interface="tns:PriceInterface" Type="http://www.w3.org/ns/wsdl/http">
</binding>

<service name="PriceService" interface="tns: PriceInterface">
  <endpoint name="PriceServiceHttpEndpoint"
    Binding="tns: PriceInterHttpBinding"
    Address="http://www.jtang.org/price"/>
</service>
```

### 3. UDDL

对于服务提供商来说,一个无法被用户发现或方便使用的 Web 服务是没有意义的。UDDI 规范的目的就是提供一个中介机构,为服务提供者和服务请求者建立沟通的桥梁,使得服务使用者更容易发现或找到目标服务。UDDI 由 Ariba、IBM、Microsoft 等多家公司于 2000 年共同建立。目前的最新版本是 UDDI 3.0.2。

UDDI 规范旨在提供这样一个基础构架平台:服务提供者,包括企业、组织或其他能够提供服务的个体,可以将自己及所提供的服务发布到平台上,服务使用者可以方便地检索其所需要的服务并获得使用服务的全部信息<sup>[38]</sup>。UDDI 基于一系列已有的工业标准(包括 HTTP、XML、XML Schema 及 SOAP),定义一个可交互的用于向 Internet 和内部网络发布 Web 服务的基础架构,并给出一系列接口实现对其自动访问。

UDDI 通过 HTTP 协议和 SOAP 协议与服务使用者及服务提供者交互。首先,服务提供者需要在 UDDI 注册中心创建一个代表自己的企业实体并提供相关的机构信息;然后,服务提供者在具体服务实现的同时,将其所提供的服务描述信息发布到 UDDI 服务注册中心;当 Web 服务描述发布到 UDDI 后,该服务描述将被注册到其提供者的企业实体下。服务请求者使用 UDDI 注册中心提供的检索功能可以找到服务描述,描述信息帮助服务请求者了解如何定位及使用这一特定的服务,在 UDDI 帮助下服务请求者与服务提供商之间建立一种绑定关系,服务请求者可以利用从服务描述中获取的服务描述信息调用该服务。

## 2.2 理论知识

本节将介绍本文研究工作的理论知识,包括:层次分析法、欧拉距离、Skyline 和粒子群算法。

### 2.2.1 层次分析法

层次分析法是萨蒂(Saaty)<sup>[39-40]</sup>等人在 20 世纪 70 年代提出的一种决策方法。利用各种信息作出优化决策是人们在实践中经常需要解决的问题。例如,厂商在激烈的市场竞争中希望选择优化经营策略;用人单位希望从应聘者中选到优秀的人才;高中毕业生希望找出最合适的报考学校和专业;顾客希望选择最满意的商品等等。在试图解决这类问题时,要考虑的因素往往很多,而且有些因素可以量化,有些因素只有定性关系。因而常常要考虑如何将定性关系定量化,以及用什么计算方法来得到优化决策。层次分析方法就是将定性因素、半定性因素量化的一种方法<sup>[41]</sup>。它将问题中考虑的各种因素层次化,并逐层比较多种有关的因素,为分析和解决决策、预测或控制等方面的问题提供一个定

量化的方法。下面将层次分析法的基本步骤介绍如下：

1. 建立层次结构模型

将问题所包含的因素分层。一般划分为：最高层、中间层和最低层。最高层表示解决问题的目的；中间层为实现总目标而采取的各种措施、方案，进一步可细分为策略层、约束层、准则层等；最底层是用于解决问题的各种措施、方案等。当某个层包含因素超过 9 个时，可将该层次再划分为若干子层。

2. 构造成对比较阵

要比较  $n$  个因素  $x_1, x_2, \dots, x_n$  对目标  $z$  影响的大小，即要确定这  $n$  个因素对目标  $z$  的相对重要性。我们用两两比较的方法将各因素的“重要性”量化。

每次取两个因素  $x_i$  与  $x_j$ ，用正数  $a_{ij}$  表示  $x_i$  与  $x_j$  的重要性之比。由全部比较结果得到矩阵  $A=(a_{ij})$ ，称作成对比较阵。显然有  $a_{ij} = 1/a_{ji}$ ， $a_{ij} > 0$ ， $1 \leq i, j \leq n$ 。  $a_{ij}$  的取值方法可参考萨蒂的方法。萨蒂引用数字 1、2、...、9 及倒数 1/2、1/3、...、1/9 作为标度，其意义是

$x_i$ 比 $x_j$	相同	稍重要	重要	很重要	绝对重要
$a_{ij}$	1	3	5	7	9

在每两个等级之间有一个中间状态， $a_{ij}$  分别取值 2、4、6、8。

3. 一致性检验

如果决策人对  $n$  个决策因素的比较具有逻辑的一致性，则成对比较阵中的元素  $a_{ij}$  之间应有下面的关系：

$$a_{ij} \cdot a_{jk} = a_{ik}, \quad 1 \leq i, j, k \leq n \tag{2.1}$$

这是因为在决策人心中，其实每个因素的重要性都有一个重要性指标。设因素  $x_i$  的重要性指标为  $w_i$ ，则根据  $a_{ij}$  表示  $x_i$  与  $x_j$  的重要性之比，即

$$a_{ij} = \frac{w_i}{w_j}$$

于是， $x_i$  与  $x_j$  的重要性之比乘上  $x_j$  与  $x_k$  的重要性之比应为  $x_i$  与  $x_k$  重要性之比。

$$a_{ij} \cdot a_{jk} = \frac{w_i}{w_j} \cdot \frac{w_j}{w_k} = a_{ik}, \quad 1 \leq i, j, k \leq n \tag{2.2}$$

称满足(2.1)式的成对比较阵  $A$  为一致矩阵。然而实际上由于人的思维活动不可避免地带有主观性和片面性，一般来说，所构造出来的成对比较阵  $A$  常常不是一致阵，即可能出现  $a_{ij} \cdot a_{jk} \neq a_{ik}$ 。因此，必须对成对比较阵  $A$  进行一致性检验。

直接对一切可能的  $i, j, k$  验证等式(2.1)是非常繁琐的，转而分析一致矩阵的特征，以找到一种检验矩阵一致性的简便方法。

设  $A$  是一致矩阵。用对应的  $w_i$  表示出来,

$$A = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} \\ w_1 & w_2 & \dots & w_n \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} \\ w_1 & w_2 & \dots & w_n \\ \dots & \dots & \dots & \dots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \dots & \frac{w_n}{w_n} \\ w_1 & w_2 & \dots & w_n \end{bmatrix}$$

由简单的计算可以得到

$$\begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} \\ w_1 & w_2 & \dots & w_n \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} \\ w_1 & w_2 & \dots & w_n \\ \dots & \dots & \dots & \dots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \dots & \frac{w_n}{w_n} \\ w_1 & w_2 & \dots & w_n \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = n \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

即:  $n$  是矩阵  $A$  的特征值, 其对应的特征向量是  $(w_1 w_2 \dots w_n)^T$ 。

事实上, 可以证明:  $n$  阶成对比较阵  $A$  是一致矩阵, 当且仅当  $A$  的最大特征值  $\lambda_{\max}(A) = n$ 。因此, 只需要计算  $A$  的最大特征值就可判断  $A$  是否为一致矩阵。如果  $A$  不具有一致性, 可以证明  $\lambda_{\max}(A) > n$ , 而且  $\lambda_{\max}(A)$  越大, 不一致程度越严重。此时  $\lambda_{\max}(A)$  对应的特征向量  $Y$  就不能真实反映  $X = \{x, x_2, \dots, x_n\}$  在目标  $z$  中所占的比重。令

$$CI = \frac{\lambda_{\max}(A) - n}{n - 1}$$

将  $CI$  作为衡量一个成对比较阵  $A$  不一致程度的标准, 称  $CI$  为一致性指标。

当成对比较矩阵  $A$  的最大特征值  $\lambda_{\max}(A)$  稍大于  $n$ , 这时称  $A$  具有满意的一致性。用这样的方法定义的一致性是不严格的, 还必须给出度量指标。萨蒂提出用平均随机一致性指标  $RI$  检验成对比较阵  $A$  是否具有满意的一致性。平均随机一致性指标  $RI$  是这样得到的: 对于固定的  $n$ , 随机构造成对比较阵  $A'$ , 其中  $a'_{ij}$  是从  $1, 2, \dots, 9, 1/2, 1/3, \dots, 1/9$  中随机抽取的。这样的  $A'$  一般是不一致的, 取充分大的子样得到  $A'$  的最大特征值的平均值  $\lambda'_{\max}(A)$ , 定义

$$RI = \frac{\lambda'_{\max} - n}{n - 1}$$

对于 1-9 阶成对比较阵  $A$ , 萨蒂用大小为 100~500 的子样, 对不同的  $n$  算出  $RI$  值如下

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

令  $CR = CI / RI$ , 则  $CR$  称为随机一致性比率, 可以用  $CR$  代替  $CI$  作为一致性检验的

临界值。当  $CR < 0.1$  时, 认为成对比较阵具有满意的一致性, 否则就必须重新调整成对比较阵  $A$ , 直至达到满意的一致性为止。这时计算  $A$  的最大特征值  $\lambda_{\max}(A)$  对应的特征向量  $Y$  (可以证明, 适当选择  $Y$  可使其各分量非负), 再求得  $Y$  的标准化向量  $Y'$  (各分量之和为 1 的特征向量),  $Y'$  就可以作为各因素的相对权值。

在实践中, 也可以采用下述方法计算  $\lambda_{\max}$  和相应特征向量的近似值。

对成对比较阵  $A = (a_{ij})$ , 令

$$u_k = \frac{\text{第 } k \text{ 行元素之和}}{\text{矩阵所有元素之和}} = \frac{\sum_{j=1}^n a_{kj}}{\sum_{i=1}^n \sum_{j=1}^n a_{ij}}$$

称  $U = (u_1, u_2, \dots, u_n)^T$  为  $n$  个因素  $x_1, x_2, \dots, x_n$  的权重向量, 它反映  $n$  个决策对象的优劣、主次等的对比。

### 2.2.2 欧氏距离

欧氏距离是广泛用于聚类分析和判别分析中常用的距离<sup>[42]</sup>。设  $x = (x_1, x_2, \dots, x_p)^T$  为所关心的  $p$  个指标, 对其进行  $n$  次观测得  $n$  组数据

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T, i = 1, 2, \dots, n$$

称这  $n$  组观测值为  $n$  个样品。这时, 每个样品可看成  $p$  维空间的一个点,  $n$  个样品组成  $p$  维空间的  $n$  个点, 我们自然用各点之间的距离来衡量各样品之间的靠近程度。如式(2.3)所示, 为欧氏距离的公式。

$$d(x_i, x_j) = \left[ \sum_{k=1}^p (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}} \quad (2.3)$$

在实际情况下, 变量之间存在一定的关系, 通常通过乘以一个权重系数(所有系数之和为 1, 且都为正)来修正平方差项, 并为重要的变量设置大的权重。带权重的欧氏距离如式(2.4)所示。

$$d(x_i, x_j) = \left[ \sum_{k=1}^p w_k (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}} \quad (2.4)$$

### 2.2.3 Skyline

Skyline 方法最早是由 Borzsonyi<sup>[27, 43-47]</sup> 等人提出并用于数据库查询优化的方法, 随后广泛应用于多标准决策、数据挖掘等领域。在 Skyline 中给出两个多维属性数据点之间的支配关系和 Skyline 集合的概念。相应地, 本节给出两个 Web 服务之间的支配关系和

Skyline 服务集合的定义。

**定义 2.1 支配关系:** 设  $S = \{s_1, s_2, \dots, s_m\}$  是一个包含  $m$  个 Web 服务的集合, 每个服务是一个具有  $n$  维服务质量的向量  $s_i = \{q_{i1}, q_{i2}, \dots, q_{in}\}$ , 即  $n$  维空间中的一个数据点。若对于服务  $s_i$  和  $s_j$  的  $n$  维服务质量向量中的任意一维服务质量, 均有  $q_{ik}$  的值优于或者等于  $q_{jk}$ , 且存在至少一维服务质量  $q_{ip}$  的值优于  $q_{jp}$ , 则称  $s_i$  支配  $s_j$ , 记为  $s_i \succ s_j$ , 其中“ $\succ$ ”表示支配关系。

**定义 2.2 Skyline 服务集合:** 在 Web 服务中, 所有不被其它任何一个服务所支配的服务构成的集合就是 Skyline 服务集合。

## 2.2.4 粒子群算法

粒子群优化算法(Particle Swarm Optimization, PSO)是源于对鸟类捕食的行为研究, 于 1995 年由 Kennedy 博士和 Eberhart 博士共同提出的一种基于群智能(Swarm Intelligence)方法的演化计算(evolutionary computation)技术<sup>[48-50]</sup>。与遗传算法相比较, PSO 的优势在于简单、容易实现, 同时又有深刻的智能背景, 既适合科学研究, 又适合工程应用<sup>[51]</sup>。它虽然也是从随机解出发, 通过迭代寻找优化解, 也是通过适应值来评价解的优劣, 但是它没有遗传算法的交叉和变异操作, 而是通过追随当前搜索到的优化值来寻找全局优化值。自 1998 年以来, IEEE 进化计算等多个国际学术会议上都举行粒子群算法的专题讨论, 而 2003 年在美国印第安纳波利斯举行的首届 IEEE 群智能研讨会和 2005 年在美国加利福尼亚举行的第二届 IEEE 群智能研讨会, 则将 PSO 算法的研究推向一个新的高潮<sup>[52-53]</sup>。因此, 粒子群算法自提出以后在很多领域的应用中获得成功。

粒子群优化算法采用简单的速度位移模型, 微粒在目标搜索空间  $D$  中飞行, 通过群体中微粒间的协作与竞争产生的群体智能指导优化搜索<sup>[54-55]</sup>。PSO 中, 每个优化问题的潜在优化解都是搜索空间的一颗粒子, 所有粒子都有一个由目标函数决定的适应值(fitness value), 每个粒子还有一个速度决定它们的飞行方向和距离。然后粒子就追随当前的优化粒子在解空间中搜索。PSO 初始化为一群随机粒子(随机解), 然后通过迭代找到优化解。在每一次迭代中, 粒子通过跟踪两个“极值”来更新自己。一个是粒子本身迄今为止寻找到的优化值对应的位置, 这个位置称之为局部最佳位置  $P_i$ ; 另一个是整个种群目前找到的优化值对应的最佳位置, 这个位置称之为全局最佳位置  $P_g$ 。第  $i$  个粒子的速度和位置更新公式如下:

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot r_1 \cdot (P_i(t) - X_i(t)) + c_2 \cdot r_2 \cdot (P_g(t) - X_i(t)) \quad (2.5)$$

$$X_i(t+1) = X_i(t) + V_i(t) \quad (2.6)$$

其中： $V_i(t+1)$ 为该粒子当前的移动速度； $V_i(t)$ 为粒子上一代的移动速度； $c_1$ 和 $c_2$ 为加速常数，通常在 $[0,2]$ 之间取值； $r_1$ 和 $r_2$ 为 $(0,1)$ 之间的两个相互独立的随机函数；粒子的最大移动速度被限制在一定范围之内，即 $V_i \in [V_{\min}, V_{\max}]$ ； $X_i \in [X_{\min}, X_{\max}]$ 控制粒子在一次迭代中最大的移动距离。 $w$ 为上一代粒子的速度对于当前代粒子速度影响的权重。 $w$ 的确定如下

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (2.7)$$

其中 $w_{\max}$ ， $w_{\min}$ 分别为开始时的和结束时的权重； $iter_{\max}$ 为最大迭代次数， $iter$ 为当前迭代次数。一般取 $w_{\max} = 0.9$ ， $w_{\min} = 0.4$ 。参数的具体设置根据具体情况而定<sup>[56-57]</sup>。

在搜索的过程中，对每个粒子位置的适应值不断地进行评价。当粒子当前位置的适应值优于自身之前所经历过的所有位置的适应值时，将当前的位置作为该粒子的局部最佳位置 $P_i$ ；同样，当粒子当前位置的适应值优于全局最佳位置的适应值时，将该粒子的位置赋给全局最佳位置 $P_g$ 。PSO算法具有鲜明的生物社会背景：认知行为和社会行为，即在寻求一致的认知过程中，个体往往记住它们的认知，同时考虑其它同伴的认知，当个体察觉同伴的认知较好时，将进行适应性调整。公式(2.5)中的第一部分 $w \bullet V(t)_i$ 是个体自身的惯性；第二部分 $c_1 \bullet r_1 \bullet (P_i(t) - X_i(t))$ 是对个体的认知部分；第三部分 $c_2 \bullet r_2 \bullet (P_g(t) - X_i(t))$ 是群体的认知部分。这三个部分之间的相互平衡和制约决定算法的主要性能。由于包含了对个体经历和群体经历中优化值的认知，因此该算法是一种启发式搜索算法。

## 2.3 本章小结

本章介绍了Web服务的相关技术以及本文研究工作的理论基础知识。其中，Web服务相关技术包括Web服务的定义和特点、Web服务的体系架构和Web服务的关键技术；本文研究工作的理论基础知识包括层次分析法、欧氏距离、Skyline和粒子群算法，为后续章节中模型及算法的介绍做下铺垫。



## 第三章 基于满意度的全局优化服务选择方法

在大量功能相同或相似服务共存竞争的环境下，面向服务质量的 Web 服务选择成为研究热点<sup>[13-14, 19, 58-60]</sup>。已有的服务选择方法通常假设同一时刻请求某种功能服务的请求者只有一个，并为其选择一个局部最佳服务。然而在 Web 服务环境下，同一时刻可能存在多个服务提供者和多个服务请求者，服务请求者对服务的 QoS 要求可能不尽相同，要使网格服务资源利用率达到最优，需要进行整体规划，制定最优服务选择方案<sup>[61]</sup>。从提供商的角度看不盲目地接受请求，需按照自身的服务能力分配给最合适的请求者；从请求者的角度看不盲目地选择最优的服务，而是使所有请求者在整体上达到最大化的满足，从而满足双方利益最大化的需求。

由于整个 Internet 网络上某一时刻的 Web 服务个数和服务请求者个数无法统计，显然无法进行全网的全局优化。然而，在一个专有的网格服务环境下某一时刻的服务提供者和服务请求者数量是固定的，实现全局最优服务选择具有可行性。因此，本章以专有网格服务环境为对象，考虑专有网格服务环境下某时刻存在 N 个服务提供者和 M 个服务请求者情形的 Web 服务全局最优选择问题，并探讨这种新型的服务选择与分配方式。

本章利用层次分析法计算出请求者对各服务的 QoS 满意度与服务信誉满意度，将二者结合起来，计算出综合满意度，并基于综合满意度进行 0-1 整数规划，建立全局最优网格服务选择模型。最后，通过仿真实验，验证该方法的有效性。

### 3.1 基于 QoS 的网格服务选择框架

网格服务环境中，请求者对网格服务的需求千变万化，反映出请求者多样性的 QoS 需求；服务的 QoS 是动态变化的，其信誉度亦不可预知。因此，既需要根据请求者的 QoS 需求判断出其偏好，又必须及时地收集和更新服务 QoS 的数据信息，并评价其信誉，以此为所有请求者做出正确的服务选择。

本章提出的网格服务选择框架结构中主要包含 3 个参与角色：服务请求者、服务提供者以及负责 QoS 度量并评价服务信誉度与决策全局优化服务选择的 QoS 管理中心。本框架与面向服务的 OGSA 架构兼容，可以在其之上直接应用和实施<sup>[62-64]</sup>。本框架适合网格这种自治的环境，在服务请求者真正调用网格服务之前，管理中心为其提供桥梁，为请求者提供满意的服务。基于 QoS 的网格服务选择框架如图 3.1 所示。QoS Registry 接受服务的注册和发布；User Feedback Collection 接受来自用户对所调用服务的反馈评价，为服务 QoS 的更新提供依据；QoS Monitor 监控服务的调用过程，检测调用时的服务 QoS 数据；Service Reputation Evaluation 根据监测到的服务 QoS 数据和用户反馈的

QoS 评价数据度量服务的信誉度; Service Distribution 根据服务 QoS 数据和服务请求 QoS 需求数据, 为所有的服务请求者分配符合用户需求的服务, 制定全局优化服务分配策略。

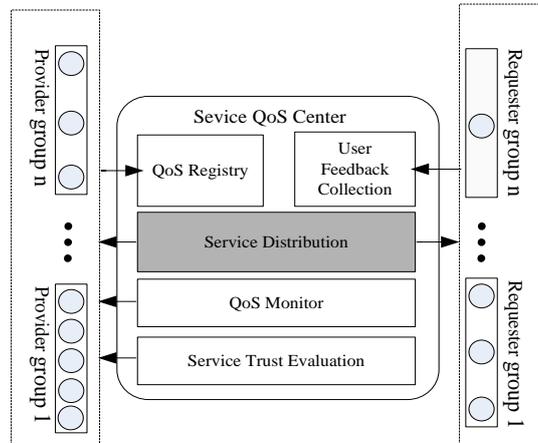


图 3.1 基于 QoS 的网络服务选择框架

Fig. 3.1 Grid service selection framework based on QoS

QoS 管理中心的功能包括：(1) 接受服务提供者的 QoS 注册；(2) 收集服务请求者的 QoS 反馈；(3) 服务执行过程的 QoS 信息监控；(4) QoS 的计算与服务分配；(5) 服务可信度评价。

该框架既可以解决一个服务请求者请求某种功能服务的服务选择问题, 也可以解决同一时刻多个服务请求者请求某种功能服务的服务选择问题。

### 3.2 网络服务的 QoS 计算模型

网络服务的 QoS 值是通过 QoS 管理中心在服务应用过程中不断收集和计算而得到的。因为不同的服务质量有不同的计算方法和重要性, 且动态变化, 在赋予各个质量分量不同权值的同时, 必须保持 QoS 值的新鲜度, 实时计算服务的 QoS 值。

网络服务  $ws$  的 QoS 值用公式(3.1)进行计算:

$$QoS(ws) = \sum_{i=1}^n w_i q_i, \text{ 其中 } \sum_{i=1}^n w_i = 1 \quad (3.1)$$

式中,  $q_i$  表示第  $i$  个质量分量, 包括执行时间 Time(服务的执行时间是从服务请求发出到收到该请求结果之间的时间延迟)、执行价格 Price(服务的执行价格是用户调用服务  $ws$  需支付的费用)、可用性 Availability(服务的可用性是服务在一定时间内可用的概率)、可靠性 Reliability(服务的可靠性是服务能正确提供它所注册服务的概率, 即服务成功完成次数与总调用次数的比值)。服务的质量分量并不限于这些方面, 在计算公式中, 该分量的数目很容易进行扩展, 在本章中考虑用户最关心的四个方面, 即执行时间、价格、

可用性和可靠性； $w_i$ 表示各个分量的权值，根据用户 QoS 需求计算获得。

### 3.3 网络服务全局优化服务选择

为弥补采用不同度量方法的 QoS 指标之间的差异，需要对 QoS 质量属性的数据进行归一化处理，将它们的值转化到[0,1]之间<sup>[62, 65]</sup>，值越大意味着 QoS 越好。本章采用文献[4]中的极差变换方法对 QoS 质量属性的数据进行归一化处理。公式(3.2)用于处理负属性(取值越大，质量越低，例如：执行时间)，公式(3.3)用于处理正属性(取值越大，质量越高，例如：可用性)。对于同一时刻存在 N 个提供者，M 个请求者，将 N 个服务的正负 QoS 分量分别按公式(3.2)、(3.3)进行归一化处理，得到 N 个服务的标准 QoS 向量。同样，从 M 个用户请求中提取对于 QoS 的要求，进行归一化处理。

$$v_i = \begin{cases} \frac{q_{\max} - q_i}{q_{\max} - q_{\min}}, & \text{if } q_{\max} - q_{\min} \neq 0 \\ 1, & \text{if } q_{\max} - q_{\min} = 0 \end{cases} \quad (3.2)$$

$$v_i = \begin{cases} \frac{q_i - q_{\min}}{q_{\max} - q_{\min}}, & \text{if } q_{\max} - q_{\min} \neq 0 \\ 1, & \text{if } q_{\max} - q_{\min} = 0 \end{cases} \quad (3.3)$$

其中， $q_i$ 和 $v_i$ 分别为某个 QoS 属性数据归一化之前和之后的值， $q_{\max}$ 和 $q_{\min}$ 分别为该质量属性所有数据中的最大、最小值。为处理方便，将归一化结果中的 0 值更改为 0.1，本章后面所提到的 QoS 值均是指标准化处理之后的值。

本章的优化选择模型考虑服务的非功能需求，即服务 QoS 和信誉度，本章信誉度的获得参见文献[66]的方法。因此服务 QoS 和信誉度是计算 Web 服务提供者和服务请求者之间是否匹配的两个指标。计算出请求者对各服务的 QoS 满意度与服务信誉的满意度，将二者结合起来，计算综合满意度，基于综合满意度进行 0-1 整数规划，所有请求者的满意度之和最大化，即可得到网络服务环境下全局最优服务选择的结果。

**定义 3.1 满意度：**为提供者提供的服务满足请求者需求的程度。例如：A 提供的质量值为 a，B 请求的质量值为 b，则 B 对 A 的满意度为 a/b。

#### 3.3.1 提供者对各请求者综合满意度的评定

考虑对 QoS 满意度的评定，设请求者 i 对 QoS 的目标需求为  $R_i = (q_{i1}, q_{i2}, q_{i3}, q_{i4})$ ，提供者 j 提供的 QoS 值为  $P_j = (p_{j1}, p_{j2}, p_{j3}, p_{j4})$ ，QoS 的四个分量分别代表：执行时间、价格、可用性和可靠性。下面分 4 个步骤来对 QoS 的满意度进行评定。

**Step1:** 用户对不同 QoS 分量的偏好

根据请求者对各维 QoS 的需求, 计算其 QoS 偏好向量  $W_i$ 。从单个请求者出发来求得各维 QoS 的偏好, 这样可以使复杂的问题简单化。为求请求者对各维 QoS 的不同偏好, 采用层次分析法的思想, 在这种方法中先建立成对比较阵。对请求者  $i$ , 可得到成对比较阵  $A$  如下:

$$A = \begin{bmatrix} \frac{q_{i1}}{q_{i1}} & \frac{q_{i1}}{q_{i2}} & \frac{q_{i1}}{q_{i3}} & \frac{q_{i1}}{q_{i4}} \\ \frac{q_{i2}}{q_{i1}} & \frac{q_{i2}}{q_{i2}} & \frac{q_{i2}}{q_{i3}} & \frac{q_{i2}}{q_{i4}} \\ \frac{q_{i3}}{q_{i1}} & \frac{q_{i3}}{q_{i2}} & \frac{q_{i3}}{q_{i3}} & \frac{q_{i3}}{q_{i4}} \\ \frac{q_{i4}}{q_{i1}} & \frac{q_{i4}}{q_{i2}} & \frac{q_{i4}}{q_{i3}} & \frac{q_{i4}}{q_{i4}} \end{bmatrix}$$

其中  $a_{12} = q_{i1}/q_{i2}$ , 即表示对请求者  $i$ , 服务时间对服务价格的重要性之比为  $a_{12}$ 。显然, 该矩阵为一致性矩阵(其性质见文献[67])。对比矩阵  $A$  的最大特征根为  $\lambda = 4$ , 对应的特征向量为  $(q_{i1}, q_{i2}, q_{i3}, q_{i4})$ , 归一化后得偏好向量  $W_i = (w_{i1}, w_{i2}, w_{i3}, w_{i4})^T$ , 其中  $w_{ik}$  的计算如下:

$$w_{ik} = q_{ik} / \sum_{j=1}^4 q_{ij} \quad (3.4)$$

通过偏好向量, 可知各维 QoS 对请求者  $i$  的重要性。用同样的方法, 可求得各维 QoS 对于其他请求者的重要性。

**Step2:** 确定同一请求者  $i$  对各提供者每一维 QoS 的满意度

对于同一请求者  $i$  和任意提供者  $j$ , 提供者满足请求者各维 QoS 期望要求的程度为两向量  $P_j$  和  $R_i$  对应元素的比值, 即  $(p_{j1}/q_{i1}, p_{j2}/q_{i2}, p_{j3}/q_{i3}, p_{j4}/q_{i4})$ 。将  $M$  个提供者对请求者  $i$  的比值组合构成矩阵  $B_{M \times 4}$ 。

**Step3:** 确定请求者  $i$  对各提供者 QoS 的满意度

令  $C_{Ni}$  为请求者  $i$  对各提供者的满意度向量, 则  $C_{Ni} = B \times W_i$ 。

**Step4:** 重复(2), (3)两步, 按照 Step3 的方法分别求出其他请求者对各提供者的满意度向量。将求得的满意度向量  $C_{N1}, \dots, C_{Ni}, \dots, C_{NM}$  组合在一起得到  $M$  个请求者对  $N$  个提供者的 QoS 满意度矩阵  $D_{N \times M} = (C_{N1}^T, \dots, C_{Ni}^T, \dots, C_{NM}^T)$ 。

对 QoS 的满意度评定之后, 考虑提供者信誉的满意度评定。取信誉度值最大的提供者的满意度为 1, 则其他任意一个提供者的信誉度值与最大信誉度值的比值作为该提供者 QoS 信誉的满意度。

对服务 QoS 和信誉的满意度评定之后, 最后计算出各请求者对各提供者的综合满意度。设  $r$  为 QoS 与信誉的比例系数, 当  $r=1$  时, 表示 QoS 与信誉同等重要, 其中  $r$  的取值根据请求者对信誉度的认可程度而定。综合满意度  $Q_j = \text{QoS 满意度} + \text{信誉满意度} \times r$ , 即算出各提供者对各请求者的综合满意度矩阵  $Q$ 。

### 3.3.2 全局优化服务选择模型及算法

设提供者的被选情况 0-1 矩阵为  $X$ ,  $x_{ij}$  是其中的元素。

$$x_{ij} = \begin{cases} 1 & \text{第}i\text{个服务分配给第}j\text{个服务请求} \\ 0 & \text{第}i\text{个服务未分配给第}j\text{个服务请求} \end{cases}$$

要使选择合理, 发挥提供者本身能力的优势, 则应使总的综合满意度最大。为每个请求者选择一个服务, 列出下列全局优化选择模型:

$$\max = \sum_{i=1}^N \sum_{j=1}^M x_{ij} Q_{ij} \quad (3.5)$$

$$s.t. \begin{cases} 0 \leq \sum_{j=1}^M x_{ij} \leq 1 & i = 1, \dots, N \\ \sum_{i=1}^N x_{ij} = 1 & j = 1, \dots, M \end{cases} \quad (3.6)$$

算出  $x_{ij}$  的值, 确定矩阵  $X$  即为所求得的最优选择方案。根据所得的最优选择方案, 同请求者的需求再进行一次比较, 若存在某个请求者所选的服务不足以满足其需求时, 我们认为为实现全局优化和使请求者同样能够得到服务, 请求者应当适当地降低要求。

根据上述的网格服务环境下的全局优化服务选择模型, 列出算法描述如下:

---

**Algorithm** Global optimal service selection based on grid service environment

---

**Input:** QoS // QoS of grid service and QoS requirement of request

**Output:** X // Scheme of grid services global optimal selection

Begin

1. new\_QoS=standardization(QoS); //standardize the value of QoS with equation (3.2) and (3.3)
2. W=preference(new\_QoS); // Calculate preference for each service request with equation (3.4)
3. D=satisfaction(new\_QoS, W); // Calculate QoS satisfaction from each provider to each request
4. s\_reputation=reputation\_satisfaction(reputation) // Calculate reputation satisfaction for each request
5. Q=D+s\_reputation\*r // Calculate integrated satisfaction for each provider with each request
6. X=LP(Q); // solve 0-1 integer linear programming model

End

---

图 3.2 网格服务环境下的全局优化服务选择算法

Fig. 3.2 Global optimal service selection algorithm based on grid service environment

### 3.3.3 全局优化服务选择模型的扩展

3.3.2 节中的模型及算法只考虑一个服务在同一时刻只服务于一个请求。而实际情

况中，有些 Web 服务可多线程执行，即可同时服务于多个请求。因此，需要对 3.3.2 节中的模型进行扩展。对于可多线程执行的 Web 服务，将整数规划的约束条件修改成如下式(3.7)即可。

$$s.t. \begin{cases} 0 \leq \sum_{j=1}^M x_{ij} \leq T(ws_i) - C(ws_i) & i = 1, \dots, N \\ \sum_{i=1}^N x_{ij} = 1 & j = 1, \dots, M \end{cases} \quad (3.7)$$

其中， $T(ws_i)$  表示服务  $ws_i$  至多可同时服务于  $T(ws_i)$  个请求， $C(ws_i)$  表示服务  $ws_i$  当前正服务于  $C(ws_i)$  个请求。当  $\sum_{i=1}^N (T(ws_i) - C(ws_i)) \geq M$  时，则部分服务请求必须等待。因此，实际情况中，服务器应提供足够大的服务能力，以满足实际应用中服务请求数量需求。

### 3.4 仿真实验及分析

该模型考虑请求者对 QoS 的需求、提供者的 QoS 和信誉度这 3 个影响请求者和提供者相互之间的因素，将请求者对 QoS 的需求转化为相应的偏好向量，提供者的 QoS 和信誉度转化为相应的满意度，使问题变得简单和清晰化。模型中对各种情况进行分析，其参数的设置可以根据实际需要取一定的值，使模型具有较强的普遍性和实用性。并且设计的解法简单易行，具有一定的通用性。从表面上看，该模型要求用整数规划以保证  $x_{ij}$  只能取 0 或 1，然而该分配问题可看作运输问题的特殊情况。因此，即使不限制，最优解也将取 0 或 1。该线性规划模型可以借助 Lingo 软件方便地进行求解，尽管模型规模可能很大。在实际问题中，针对 N 较大，而 M 相对较小情况，可以在求解前淘汰掉一些 QoS 较差的 Web 服务，保留部分 QoS 较好的服务供选择，以减小 N 的值，提高模型求解的效率。在下一章中将详细提到利用该思想在 Skyline 服务集合中进行服务选择的方法。下面以一个例子来说明该模型算法的有效性。有 5 个服务提供者满足 3 个请求者的功能需求，它们的 QoS 及信誉度如表 3.1 所示，服务请求对服务 QoS 的需求如表 3.2 所示。

表 3.1 Web 服务的 QoS

Tab. 3.1 QoS and reputation of Web services

	Time	Cost	Availability	Reliability	Reputation
ws <sub>1</sub>	10	500	1	0.9	0.7
ws <sub>2</sub>	15	100	0.8	0.7	0.6
ws <sub>3</sub>	5	200	0.6	0.8	0.8
ws <sub>4</sub>	20	300	0.9	1	0.9
ws <sub>5</sub>	15	200	0.7	0.6	0.8

表 3.2 服务请求的 QoS 需求  
Tab. 3.2 QoS demand of service requests

	Time	Cost	Availability	Reliability
$r_1$	15	200	1	0.9
$r_2$	20	200	0.8	0.7
$r_3$	15	300	0.7	0.8

将 QoS 值进行标准化后, 取  $r=1$ , 利用 3.3.1、3.3.2 节中的模型及算法得出分配结果为:  $x_{13} = x_{31} = x_{42} = 1$ , 其余  $x_{ij} = 0$ 。从上面例子可以看出, 所选择的结果正是对 3 个请求者最佳分配的服务, 说明该模型算法的有效性, 能够反映用户真实的 QoS 需求。

### 3.5 本章小结

本章根据目前服务选择的研究对实际的服务选择问题进行有效性的分析, 得出提供者和请求者双方应有需求准则。提出 Web 服务全局优化选择模型, 解决专有网格服务环境下同一时刻存在多服务提供者和多服务请求者的服务选择问题, 使提供者和请求者双方的利益均能从整体上得到优化。利用层次分析法计算出请求者对各服务的 QoS 满意度与服务的信誉满意度, 将二者结合起来, 计算出服务的综合满意度, 并基于综合满意度进行 0-1 整数规划, 建立全局最优网格服务选择模型, 使资源得到合理分配, 最大化系统整体利益。实验结果表明本方法具有较好的实用性和有效性。该章提出的模型同样可以应用于其他可以统计出服务提供者和请求者数量的集中式服务选择环境, 解决该环境下的服务选择优化问题。



## 第四章 基于匹配度的全局优化服务选择方法

在大量相似 Web 服务共存竞争的环境下, 同一时刻请求同种功能服务的请求可能存在多个, 全局优化服务选择成为 Web 服务选择研究领域的一个新难题。这不仅是在网络服务环境下存在的问题, 在一般的 Web 服务情形下也是很正常的<sup>[24-25]</sup>, 例如: 在节假日期间, 人们倾向于使用 Web 服务网上订票<sup>[68]</sup>。要使 Web 服务资源利用率达到最优, 需进行整体规划。第三章主要从请求者的角度考虑所有请求者的满意度, 通过约束条件满足了某一时刻 Web 服务的负载平衡问题, 但未从长时间的角度考虑资源的最佳利用。实际上, 从长时间的角度来看, 某一时刻所有的请求者不一定要选择服务质量最优的服务, 而是选择使所有请求者全局最佳匹配的服务, 使得资源得以合理利用, 从而从长时间来看可使更多的服务请求得到满足。本章利用带权 Euclidean 距离计算 Web 服务对各请求者的距离(即匹配度)使 Web 服务对请求者的匹配度之和最大, 从服务请求者和服务提供者双方的角度和长时间的角度来考虑 Web 服务选择问题, 利用 0-1 整数规划建立全局优化选择模型, 并结合实际提出通用可行的解决多冲突服务请求的全局优化选择算法 GOSSMR (Global optimal Web service selection for multiple requests), 使最多的 Web 服务请求需求得到满足但又尽可能不浪费资源, 保证资源的合理利用, 避免负载问题的发生, 提高系统性能。为提高算法的效率, 本文基于 Skyline 的方法对 GOSSMR 进行改进, 提出 Skyline GOSSMR 算法。降低候选服务的规模, 提高算法的执行效率, 并分析算法的时间复杂度。最后, 通过仿真实验, 比较两种算法的执行效率, 验证模型算法的可行性和有效性。

### 4.1 基于 QoS 的集中式 Web 服务选择框架

Web 环境中, 请求者对 Web 服务的需求千变万化, 反映出请求者多样性的 QoS 需求<sup>[10]</sup>; 同时由于网络环境的不确定性, 服务的 QoS 是动态变化的, 其信誉度亦不可预知。因此, 必须及时地收集和更新服务的 QoS 数据信息, 并评价其信誉, 以此为系统提供服务选择的依据。

本章提出的服务选择框架中主要包含 3 种角色: 服务请求者、服务提供者以及负责 Web 服务选择的选择器 Selector。服务请求者是请求调用 Web 服务的用户; 服务提供者是提供 Web 服务的供应商, 它实现具体的 Web 服务应用程序, 并将 Web 服务发布到服务注册中, 供服务请求者使用; 选择器为请求者和服务提供桥梁, 为请求者提供满意的服务。本框架基于本实验室做的 Web 服务超市系统<sup>[69]</sup>, Web 服务好比超市中的商品, 而服务请求就像超市中的顾客。不同的是, 系统将为请求者提供全局最优的选择, 而不

是由用户自己任意选择。基于 QoS 的 Web 服务选择整体框架如图 4.1 所示。

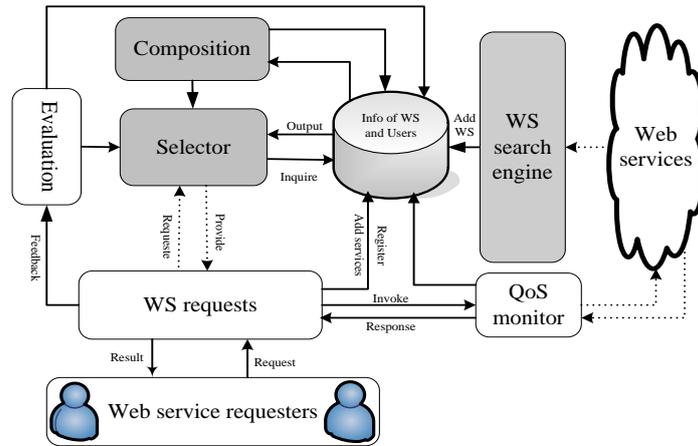


图 4.1 基于 QoS 的 Web 服务选择框架

Fig. 4.1 Web service selection framework based on QoS

框架中的数据库存储 Web 服务相关的信息和请求者信息，即用户的信息。WS search engine 单元从互联网中搜索服务，将其存入数据库。目前，系统已搜索到将近 6000 多个服务供请求者使用。另一方面，请求者可以向数据库手动添加 Web 服务或注册请求者信息；Composition 单元负责服务组合，为请求者提供更加丰富的选择；Evaluation 单元收集来自请求者的反馈；QoS monitor 监测来自 Web 服务调用的数据。多个请求者向 Selector 单元发出请求，Selector 单元在数据库中查找满足请求者功能需求的服务或组合服务，并根据 Web 服务的 QoS 信息和服务请求的需求信息进行匹配，制定全局服务选择策略，将各自匹配的结果返回给请求者。该单元包括：QoS 的计算、QoS 的标准化、QoS 量化以及请求者的偏好计算等，如图 4.2 所示为 Selector 所包含的构件。请求者确认接收后，开始发出调用服务请求，在服务调用过程中，QoS monitor 将检测服务调用的情况。请求者收到服务返回的结果，即完成调用。之后，请求者可随时登陆系统，对所调用过的服务进行评价，Evaluation 单元将收集处理来自请求者对服务的评价数据。

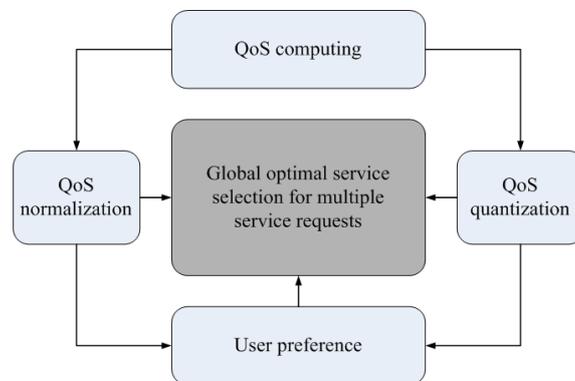


图 4.2 Web 服务选择的选择器

Fig. 4.2 Web service selector

## 4.2 Web 服务的 QoS 计算模型

Web 服务的 QoS 通过管理系统在应用过程中不断收集和计算得到。不同的服务质量有不同的计算方法和重要性，且动态变化，在赋予各个质量分量不同权值的同时，须保持 QoS 值的新鲜度，借鉴 Web 服务的 QoS 计算方法，实时地计算 Web 服务的 QoS 值。Web 服务的 QoS 值用(4.1)公式进行计算：

$$QoS(ws) = \sum_{i=1}^n w_i q_i, \text{ 其中 } \sum_{i=1}^n w_i = 1 \quad (4.1)$$

式中， $q_i$  表示第  $i$  个质量分量，包括执行价格 Price、执行时间 Time、可用性 Availability、可靠性 Reliability 和信誉度 Reputation (所有请求者对该 Web 服务评价的平均值)。Web 服务的质量分量不限于这些方面，其分量的数目容易根据实际情况进行扩展或缩小，本章只考虑请求者常用的这五个方面； $w_i$  表示分量的权值，根据 QoS 需求计算获得。注意到请求者对 Web 服务信誉等级的评价是模糊的，不妨设相应的模糊评语集为{很好、好、一般、差}，对应数值为{5, 4, 3, 2}。注意到，请求者对不满意程度的敏感远远大于对满意程度的敏感，即请求者对 Web 服务的信誉程度降低一级可能导致请求者极大的抱怨，但对信誉程度增加一级只能引起满意程度的少量增长。为此，根据实际情况取偏大型柯西分布隶属函数<sup>[70]</sup>

$$f(x) = \begin{cases} [1 + \alpha(x - \beta)^{-2}]^{-1}, & 1 \leq x \leq 3 \\ a \ln x + b, & 3 \leq x \leq 5 \end{cases} \quad (4.2)$$

其中  $\alpha, \beta, a, b$  为待定常数。令  $f(5) = 1$ ,  $f(3) = 0.8$ ,  $f(1) = 0.01$ ，可得模糊评语集的量化值为{1, 0.9126, 0.8, 0.5245}。量化之后，将所有请求者对 Web 服务的评价求平均值，即为该 Web 服务的信誉度。由于信誉是对 Web 服务的综合评价，因此将 QoS 的计算公式(4.1)进行修正为：

$$QoS(ws) = (1 - \alpha) \sum_{i=1}^4 w_i q_i + \alpha \cdot rep, \text{ 其中 } \sum_{i=1}^4 w_i = 1, 0 \leq \alpha \leq 1 \quad (4.3)$$

式中， $rep$  代表信誉度，参数  $\alpha$  表示请求者对 Web 服务信誉重视程度的差异，其取值需要根据请求者对信誉度的认可程度而定。例如：当  $\alpha = 1/2$  时，表示其他 QoS 的综合和信誉度同等重要。

## 4.3 全局优化 Web 服务选择

Web 服务选择实质上是一个优化问题，即从一组功能相同或相似的 Web 服务中选择性能较优的 Web 服务。对于同一时刻存在  $N$  个 Web 服务  $M$  个请求者情形的 Web 服

务选择问题，必须进行全局优化。Web 服务请求者对 Web 服务的 QoS 需求各不相同，因此需要根据请求者的 QoS 偏好来选择合适的 Web 服务。

### 4.3.1 QoS 标准化及偏好的计算

对第 4.2 节中所述的 Web 服务 QoS 模型，将 Web 服务 QoS 或请求者需求的 QoS 模型化为一个 5 维的向量。设第  $i$  个 Web 服务提供的 Web 服务为  $cs_i$ ， $cs_i$  的 QoS 为  $P_i = (p_{i1}, p_{i2}, p_{i3}, p_{i4}, rep_i)$ ；第  $j$  个请求者为  $r_j$ ， $r_j$  的 QoS 需求为  $R_j = (q_{j1}, q_{j2}, q_{j3}, q_{j4}, rep_j)$ 。为弥补采用不同度量方法的 QoS 指标之间的差异，需要对 QoS 质量属性的数据进行归一化处理，将它们的值转化到  $[0,1]$  之间<sup>[62]</sup>，QoS 值越大表示服务质量越好。本章采用文献[4]中的极差变换的方法对 QoS 质量属性的数据进行归一化处理（见 3.3 节）。QoS 数据归一之后，各维 QoS 的单位均变为无量纲化，则可根据公式(4.4)计算  $r_j$  对 QoS 需求的偏好向量  $W_j = (w_{j1}, w_{j2}, w_{j3}, w_{j4})$ ，其中

$$w_{jk} = q_{jk} / \sum_{k=1}^4 q_{jk} \quad (4.4)$$

### 4.3.2 匹配度的计算

本章借鉴分类领域中向量分类法的思想，将 5 维 QoS 数据的集合视为多维空间，5 维向量视为空间中的一个点，则满足请求者 QoS 需求的 Web 服务匹配问题即为对象之间的距离测量问题。本章在传统 Euclidean 距离测量方法<sup>[62]</sup>的基础之上添加权值，测量两个向量之间的相似程度，以此衡量 Web 服务和请求者需求的匹配度，值越小表示匹配度越大。如式(4.5)， $dis(cs_i, r_j)$  表示  $cs_i$  和  $r_j$  之间的匹配度。该匹配度计算模型不仅可满足请求者指定 QoS 的量值需求，还可满足请求者对 QoS 的偏好需求。计算得出的距离则反映出  $cs_i$  与  $r_j$  之间的匹配度，距离越近，匹配度越高。

$$dis(ws_i, r_j) = \sqrt{(1-\alpha) \sum_{k=1}^4 w_{jk} (p_{ik} - q_{jk})^2 + \alpha (rep_i - rep_j)^2}, \text{ 其中 } \sum_{k=1}^4 w_{jk} = 1, 0 \leq \alpha \leq 1 \quad (4.5)$$

本节使用了模糊模式识别的方法定义匹配度，然而该方法不能区分相同数值的“剩余”与“欠缺”两种情况的不同。因此，针对该匹配问题，对匹配度的定义需要做些修正。为区分相同匹配度数值的“剩余”与“欠缺”两种情况的差别，在利用式(5)计算匹配度之前，先进行判断。设  $A = (1-\alpha) \sum_{k=1}^4 w_{jk} (p_{ik} - q_{jk}) + \alpha (rep_i - rep_j)$ ，若  $A < 0$ ，表示不满足请求者的需求，则令  $dis(ws_i, r_j) = \text{Max}$ ，Max 为一个相对较大的数；否则，按式(5)计算  $dis(ws_i, r_j)$  的值。如图 4.3 所示，为二维情况下的“剩余”与“欠缺”情形。可看出，处在 A 区的服务是完全满足用户需求的，即有  $A > 0$ ；在 C 区的服务是完全不满足用户需求

的，即有  $A < 0$ ；而处在B和C两区的服务是部分满足用户需求的，既有  $A > 0$  的服务，也有  $A < 0$  的服务。在B和C区中的服务，对于  $A > 0$  的服务，可认为服务在整体上是满足用户需求的。

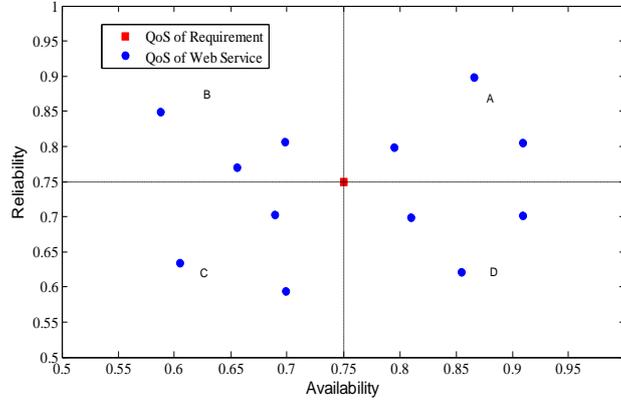


图 4.3 二维情况下的“剩余”与“欠缺”情形  
Fig. 4.3 Situation of “surplus” and “deficiency”

### 4.3.3 全局优化服务选择模型及算法

设有  $N$  个 Web 服务均处于空闲，有  $M$  个服务请求正请求该类服务。为使尽量多的请求者需求得到满足，实现资源选择的优化，需为 Web 服务对请求者进行合理的选择，使所选择方案的全局匹配度最大，即距离之和最小。设 Web 服务选择情况的 0-1 矩阵为  $X$ ， $x_{ij}$  是其中的元素

$$x_{ij} = \begin{cases} 1 & \text{第}i\text{个服务分配给第}j\text{个服务请求} \\ 0 & \text{第}i\text{个服务未分配给第}j\text{个服务请求} \end{cases}$$

以总距离之和最小为目标函数，为每个请求者选择一个服务，建立全局优化选择模型：

$$\min = \sum_{i=1}^N \sum_{j=1}^M x_{ij} \text{dis}(ws_i, r_j) \quad (4.6)$$

$$s.t. \begin{cases} 0 \leq \sum_{j=1}^M x_{ij} \leq T(ws_i) - C(ws_i) & i = 1, \dots, N \\ \sum_{i=1}^N x_{ij} = 1 & j = 1, \dots, M \end{cases} \quad (4.7)$$

其中， $T(ws_i)$  表示服务  $ws_i$  至多可同时调度给  $T(ws_i)$  个请求， $C(ws_i)$  表示服务  $s_i$  当前正服务于  $C(ws_i)$  个请求。当  $\sum_{i=1}^N (T(ws_i) - C(ws_i)) \geq M$  时，则部分服务请求必须等待服务。算出  $x_{ij}$  的值，矩阵  $X$  即为所求得的最优 Web 服务选择方案。

根据上述 Web 服务环境下的全局优化选择模型，列出 Web 服务全局优化选择算法 GOSSMR，其算法描述如图 4.4 所示：

**Algorithm GOSSMR****Input:** QoS // QoS of Web service and QoS demand of request**Output:** X // Policy of global optimal Web service selection

1. rep=f(Rep); // quantization of reputation grades
2. normalization(QoS); // normalize the values of QoS
3. W=preferece (QoS\_r); // calculate preferences with (6)
4. for i=1:N, j=1:M // N Web services and M requests
5. if A<0
6. dis=Max;
7. else dis=distance(QoS\_ws\_i, QoS\_r\_j); // calculate distances with (7)
8. end if
9. end for
10. X=LP(DIS); // 0-1 integral linear programming model; DIS is the matrix of distance

图 4.4 算法 GOSSMR

Fig. 4.4 GOSSMR algorithm

该算法的时间复杂度为 $O(N!(N-M)!)$ 。然而, 0-1 整数规划问题被分类为 NP 难问题。对大规模的数据, 很难找到最优解, 故需研究提高算法效率的方法, 进一步提高算法的可行性。实际问题中, 针对  $N \gg M$  的情况, 可在求解前过滤掉一些服务质量较低的 Web 服务, 保留足够数量的且 QoS 较好的 Web 服务, 以减小 N 的值, 降低算法的时间复杂度, 例如, 可参照文献[27,43-44]的方法, 只在不被任何其他服务所支配的服务集中选择服务, 即只在 Skyline 服务集中选择服务, 可大大降低待选服务的规模, 关于支配的概念可见文献[27,43-44], 若一次求的 Skyline 中服务数太少, 则可以在剩下的服务中再求一次 Skyline 服务集合, 再在这两次所求得的 Skyline 服务集中进行服务选择, 如图 4.5 所示为二维情况下求两次 Skyline 服务集合的例子。

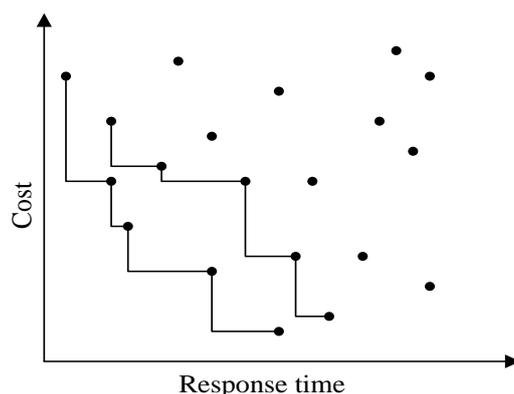


图 4.5 二维情况下求两次 Skyline 的例子

Fig. 4.5 An example of solving Skyline twice

依此类推, 可多次求 Skyline 服务集合, 直到满足服务请求的数量。这种改进算法只需在 GOSSMR 算法中加入求解 Skyline 服务集合的步骤即可, 本章记改进的算法为 Skyline GOSSMR, 其算法描述如图 4.6 所示。

**Algorithm Skyline GOSSMR**


---

**Input:** QoS // QoS of Web service and QoS demand of request  
**Output:** X // Policy of Web services global optimal selection

1. rep=f(Rep); // quantization of reputation grade
2. normalization(QoS); // normalize the value of QoS
3. if  $N \geq k * M$  // that is  $N \gg M$
4. skyline(QoS\_ws); //solve the skyline
5. update(ws); // update the set of ws
6. end if
7. W=preference (QoS\_r); // calculate preference with (6)
8. for  $i=1:N, j=1:M$  // N Web services and M requests
9. if  $A < 0$
10. dis=Max;
11. else dis=distance(QoS\_ws\_i, QoS\_r\_j); // calculate distances with (7)
12. end if
13. end for
14. X=LP(DIS); // 0-1 integral linear programming model; DIS is the matrix of distance

---

图 4.6 算法 Skyline GOSSMR

Fig. 4.6 Skyline GOSSMR algorithm

## 4.4 仿真实验及分析

本节将以一个计算实例说明该模型与算法的有效性。令  $N=5$ ,  $M=3$ ,  $N$  是 Web 服务的数量,  $M$  是服务请求的数量。Web 服务的 QoS 及信誉度如表 4.1 所示, 服务请求的 QoS 及信誉度需求如表 4.2 所示, 它们均是通过随机函数在一定范围内产生。

表 4.1 Web 服务的 QoS 和信誉度

Tab. 4.1 QoS and reputation of Web services

	<b>Time</b>	<b>Cost</b>	<b>Availability</b>	<b>Reliability</b>	<b>Reputation</b>
<b>ws<sub>1</sub></b>	10	500	1	0.9	very good
<b>ws<sub>2</sub></b>	15	100	0.8	0.7	general
<b>ws<sub>3</sub></b>	5	200	0.6	0.8	bad
<b>ws<sub>4</sub></b>	20	300	0.9	1	very good
<b>ws<sub>5</sub></b>	15	200	0.7	0.6	general

表 4.2 Web 服务请求的 QoS 和信誉度需求

Tab. 4.2 QoS and reputation demand of service requests

	<b>Time</b>	<b>Cost</b>	<b>Availability</b>	<b>Reliability</b>	<b>Reputation</b>
<b>r<sub>1</sub></b>	20	250	1	0.9	general
<b>r<sub>2</sub></b>	20	200	0.8	0.7	very good
<b>r<sub>3</sub></b>	15	300	0.7	0.8	good

QoS 值标准化后, 取  $\alpha=1/2$ ,  $\text{Max}=10$ 。为方便说明, 该实例中设每个服务在某时刻只能服务于一个服务请求, 为每个服务请求选择一个服务。根据 4.2 节中的模型及算法, 求解得出结果:  $x_{12}=x_{23}=x_{41}=1$ , 其余  $x_{ij}=0$ , 对应的匹配度为:  $\text{dis}(ws_1, r_2)=10$ ,  $\text{dis}(ws_2, r_3)=0.3133$ ,  $\text{dis}(ws_4, r_1)=0.2739$ , 说明在使全局匹配度最大化的条件下,  $r_2$  请求失败, 无满足需求的 Web 服务。从该实例中可知, 结果正是对  $r_1$ 、 $r_3$  请求者最佳匹配的 Web 服务, 说明该模型算法的有效性, 能满足请求者真实的 QoS 需求。

通过以上实例, 可知在多服务请求的情形下是如何制定全局最优 Web 服务选择的。接下来, 本节将给出大规模数据的仿真, 并同传统考虑单个服务请求的基于 QoS 加权和的 Web 服务选择算法进行比较, 以评价全局优化 Web 服务选择算法的性能。仿真实验的微机配置为 PentiumD 3400MHz 处理器, 1G 内存, 操作系统为 Windows XP, 算法用 Matlab7.1 实现。在传统的 Web 服务选择算法中, 为每个请求者选择 QoS 加权和最大的 Web 服务。另外, 本节还评价在不考虑 Web 服务负载的情形, 为每个服务请求选择最佳匹配的 Web 服务。分别列出三个表格评价三种 Web 服务选择算法的优劣。取  $N=50$  和 100,  $M=80$ , 其中,  $N$  是 Web 服务的数量,  $M$  是服务请求的数量。QoS 数据在一定范围内随机产生, 其中执行时间和执行价格的数据在一定概率内呈负相关的关系产生。表 4.3、4.4、4.5 分别给出三种 Web 服务选择算法的仿真计算结果。

表 4.3 传统 Web 服务选择算法的性能(请求数量=80)

Tab. 4.3 Performance in classical Web service selection algorithm (request number=80)

Classic web service selection algorithm based on weighted summation of QoS factors											
WS	Web Service Number=50					WS	Web Service Number=100				
	SA	SN	DN	TT	TC		SA	SN	DN	TT	TC
WS <sub>1</sub>	7	0	0	20	31	WS <sub>1</sub>	1	0	0	7	304
WS <sub>2</sub>	8	0	0	16	120	WS <sub>2</sub>	4	0	0	18	73
WS <sub>3</sub>	10	0	0	18	69	WS <sub>3</sub>	2	0	0	16	106
WS <sub>4</sub>	7	0	0	14	212	WS <sub>4</sub>	8	0	0	4	396
WS <sub>5</sub>	8	0	0	14	187	WS <sub>5</sub>	8	0	0	15	142
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>6</sub>	7	34	27	11	215	WS <sub>7</sub>	8	7	0	16	164
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>31</sub>	7	3	0	8	258	WS <sub>34</sub>	10	0	0	15	137
WS <sub>32</sub>	6	25	19	15	108	WS <sub>35</sub>	5	16	11	17	67
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>43</sub>	8	18	10	2	436	WS <sub>68</sub>	6	57	51	1	445
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>50</sub>	9	0	0	11	263	WS <sub>100</sub>	1	0	0	5	362
<b>Overall</b>	299	80	56	75	18632	<b>Overall</b>	561	80	62	68	27585

表 4.4 选择最佳匹配的 Web 服务选择算法的性能 (请求数量=80)

Tab. 4.4 Performance by selecting the most matched Web service for each request (request number=80)

Select the most matched Web service for each request											
Web Service Number=50						Web Service Number=100					
WS	SA	SN	DN	TT	TC	WS	SA	SN	DN	TT	TC
WS <sub>1</sub>	7	0	0	20	31	WS <sub>1</sub>	1	0	0	7	304
WS <sub>2</sub>	8	3	0	16	120	WS <sub>2</sub>	4	0	0	18	73
WS <sub>3</sub>	10	0	0	18	69	WS <sub>3</sub>	2	1	0	16	106
WS <sub>4</sub>	7	0	0	14	212	WS <sub>4</sub>	8	0	0	4	396
WS <sub>5</sub>	8	0	0	14	187	WS <sub>5</sub>	8	0	0	15	142
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>50</sub>	9	5	0	11	263	WS <sub>100</sub>	1	0	0	5	362
<b>Overall</b>	299	80	22	45	17904	<b>Overall</b>	561	80	28	64	18745

表 4.5 选择全局最优的 Web 服务选择算法的性能 (请求数量=80)

Tab. 4.5 Performance in global optimal Web service selection (request number=80)

Global optimal Web service selection algorithm for multiple service requests											
Web Service Number=50						Web Service Number=100					
WS	SA	SN	DN	TT	TC	WS	SA	SN	DN	TT	TC
WS <sub>1</sub>	7	4	0	20	31	WS <sub>1</sub>	1	0	0	7	304
WS <sub>2</sub>	8	6	0	16	120	WS <sub>2</sub>	4	1	0	18	73
WS <sub>3</sub>	10	2	0	18	69	WS <sub>3</sub>	2	2	0	16	106
WS <sub>4</sub>	7	2	0	14	212	WS <sub>4</sub>	8	0	0	4	396
WS <sub>5</sub>	8	4	0	14	187	WS <sub>5</sub>	8	1	0	15	142
...	...	...	...	...	...	...	...	...	...	...	...
WS <sub>50</sub>	9	5	0	11	263	WS <sub>100</sub>	1	0	0	5	362
<b>Overall</b>	299	80	0	20	18336	<b>Overall</b>	561	80	0	20	18340

在表格中, SA 是当前 Web 服务的能力, 例如: SA=5, 表示为保证服务不负载, 该服务最多还可以同时接受 5 个服务请求。Web 服务能力的数值在 [1,5] 范围内随机产生; SN 是当前该服务接受的请求数; DN 是延迟的服务请求数, 即这些服务请求需要等待服务。因此, 当  $SN \geq SA$  时, 有  $DN = SN - SA$ , 例如表 4.3 中, 对于  $WS_6$ , 有  $27 = 34 - 7$ 。当服务请求延迟时, 该服务请求者可能给相应的 Web 服务一个“差”的评价, 从而导致该 Web 服务的信誉度下降; TT 和 TC 分别表示使用某 Web 服务完成所有服务请求所花的时间和所有请求所需支付的费用。

从上面的表格可以看出, 在 GOSSMR 算法中, 服务请求不需要等待, 因此总的的时间开销最少。同时, 在整个过程中, 没有超过负载的 Web 服务, 因此 Web 服务的信誉将得到改善。在表 4.3 中, 只有 3 个 QoS 较好的 Web 服务被选择, 且所有被选择 Web 服务均超过负载。在表 4.5 中, 有较多的 Web 服务被选择。相比之下, 在表 4.4 中有较

少的 Web 服务被选择，也就意味着在被选的 Web 服务当中有部分 Web 服务超过负载。对于超过负载的 Web 服务，由于请求者需要等待，可能给服务一个“差”的评价，因此可能导致 Web 服务的信誉度下降。然而，表 4.5 中的总费用有时候可能会比表 4.4 稍微多一点，特别是当服务请求超过所有最佳匹配服务所能承受的服务能力时。但是，既然服务执行费用只是计算匹配度的一个因素，这种情况是极少发生的。在表 4.3 中，当 Web 服务的数量增加到 100 时，总的执行时间下降，这是由于请求者有更多的机会选择 QoS 加权和更大的 Web 服务。然而，由于 QoS 较高的 Web 服务通常服务的执行费用也相对较高，因此总的执行费用增加了。在表 4.4 和 4.5 中，请求者有更多机会找到更加匹配的 Web 服务。因此，部分 QoS 更好的 Web 服务可能反而未被选择，留给后面有更高 QoS 需求的服务请求，从而使得资源的合理利用，尽管总的执行时间和执行费用可能会增加。因此，使用 GOSSMR 算法，总体的性能能够被改善。特别是在 Web 服务在一个时刻只能服务于一个请求的情形下，改善的效果将更加明显。而且，从用户的角度看，总的执行时间相比传统的方法也会大大减少。

下面仿真实验比较 GOSSMR 和 Skyline GOSSMR 两个算法时间的执行效率。设  $M=10$ ， $N$  分别取 10, 50, 100, 150, 200, 250 和 300，比较两算法的 CPU 时间开销，如图 4.7 所示。当服务数为 10 时，两算法的时间开销一样，因  $N < k * M$  时，Skyline GOSSMR 中省略 Skyline 服务集合的计算，此时和 GOSSMR 一样，本实验中取  $k=5$ 。随着服务数量的增加，GOSSMR 的时间开销增加较快，且一直高于 Skyline GOSSMR，差距越来越大，而 Skyline GOSSMR 的时间开销增加缓慢，其时间开销增加的原因是随着 Web 服务数量的增加，求解 Skyline 服务集合的时间增加了，尤其是当需多次求解 Skyline 服务集合时，时间开销将增加更多。因此，相对 GOSSMR，Skyline GOSSMR 更具可行性。

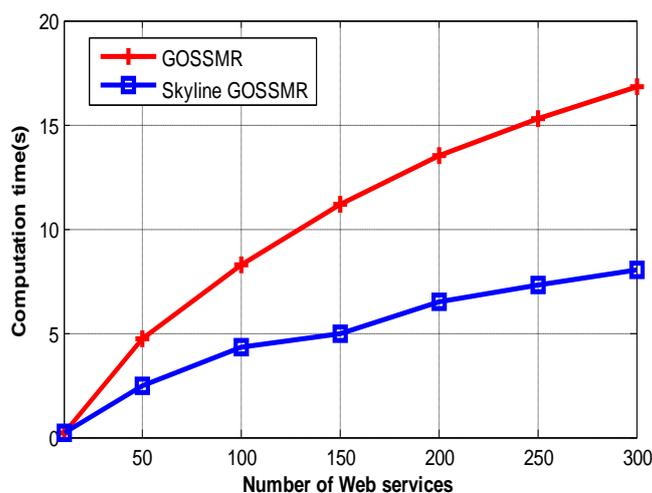


图 4.7 算法计算时间 (GOSSMR VS Skyline GOSSMR)

Fig. 4.7 Computation time of algorithm (GOSSMR VS Skyline GOSSMR)

## 4.5 本章小结

上一章中针对 Web 服务选择的实际情况对现有服务选择方法的有效性进行了分析, 得出了 Web 服务和请求双方应有需求准则。进一步地, 本章从长时间的角度分析 Web 服务全局优化选择问题, 得出了基于匹配度的服务选择策略。针对同时存在多个冲突服务请求的 Web 服务优化选择问题, 基于本实验室开发的 Web 服务超市平台<sup>[69]</sup>, 提出了基于 QoS 的 Web 服务选择框架。给出了 QoS 的计算模型, 并提出了将信誉等级进行量化的方法。利用带权的 Euclidean 距离计算 Web 服务与各服务请求需求之间的匹配度, 基于 0-1 整数规划建立了面向 QoS 的 Web 服务全局优化选择模型, 使最多的服务请求需求得到满足但又尽可能不浪费资源, 保证资源的合理利用, 避免负载问题的发生, 提高系统的性能。提出了相应的 Web 服务优化选择算法 GOSSMR 及其改进的算法 Skyline GOSSMR。实验结果表明该方法具有较好的实用性和有效性。



## 第五章 组合服务中 QoS 全局最优动态服务选择方法

QoS 全局最优动态 Web 服务选择是服务组合中的一个难题<sup>[71-72]</sup>。本章基于粒子群进化算法,设计一种用于解决该问题的 PSO-GODSS(Global optimal of dynamic Web service selection based on PSO)算法。算法的主要思想是将问题表示为一个带 QoS 约束的多目标服务组合优化问题,通过理想点的方法将多目标向单目标转化,然后利用粒子群算法的智能优化原理进行算法设计及求解,最终产生一组满足约束条件的优化服务组合流程集。理论分析和实验结果表明该算法的可行性和有效性,且算法执行效率优于以往的多目标遗传算法。相对多目标遗传算法,本章提出的方法规则简单,易于编程实现,收敛速度快,搜索的全局性更好,能够快速找到全局最佳服务组合决策,在实际应用中是可行的。一般情况下,对服务请求者来说只要求找出满足条件约束的组合服务,而并非是全球最优服务,这可通过控制迭代次数来实现。因此,进化算法更适合解决该类型的组合优化问题。

### 5.1 服务组合问题描述

**定义 5.1 (多约束条件下多目标最优路径(multi-constrained, multi-objective optimal path- MCOOP))<sup>[31]</sup>:** 对于图  $G=(N,E,W)$ ,  $N$  为顶点集,  $E$  为链路集,  $W$  为链路权重, 设存在  $k(k \geq 2)$  个约束  $C_i(i=1, \dots, k)$ , 从源点  $S$  到目的点  $T$  的全部路径集合为  $\Omega$ , 对  $\forall P \in \Omega$ , 具有  $m(m \geq 2)$  个非负的性能度量指标  $f_1, \dots, f_m$ 。若  $P^* \in \Omega, \forall P \in \Omega(P \neq P^*)$ , 在  $P$  和  $P^*$  均满足约束  $C_i$  的条件下, 对于所有的度量准则均使得  $f_i(P^*) \succ= f_i(P)(i=1, \dots, m)$ , 至少存在一个严格不等  $f_i(P^*) \succ f_i(P)(i=1, \dots, m)$ , 则称  $P^*$  为多约束条件下多目标问题的 Pareto 优化解, 其中  $\succ=$  和  $\succ$  分别表示不劣于和优于关系。

服务结点(service node, SN)是一个抽象的概念,各服务结点只包含对 Web 服务的功能描述和接口信息,而不指向具体的 Web 服务<sup>[73]</sup>。在动态服务组合中,服务组合流程模型由多个抽象的服务结点组成,每个结点对应一个 Web 服务群(service group, SG)。同一服务群中的多个 Web 服务具有相同或相似的功能,而 QoS 不尽相同。如图 5.1 所示,为一个包含  $m$  个服务结点的串行服务组合流程,第  $i$  个服务结点  $SN_i$  包含有  $n_i$  服务,即  $SG_i=(s_{i,1}, \dots, s_{i,n_i})(i=1, \dots, m)$ 。



图 5.1 串行服务组合流程

Fig. 5.1 Sequence Web service composition process

动态 Web 服务选择 QoS 全局最优化问题,即是在组合流程模型执行过程中,从各个服务结点对应的服务群中选择出具体的服务组成一个可执行的服务链,使得服务链在

满足 QoS 约束的前提下，多个目标(QoS 参数)达到最优。由定义 5.1，可把服务组合的服务动态选择 QoS 全局最优化问题转化为一个求解服务组合流程图(service composition graph-SCG)中带 QoS 约束条件的多目标最优路径问题，亦即 SCG 中的 MCOOP 求解问题。每个服务结点所对应的服务群中的服务可看作图的顶点。为方便分析服务选择，在图中添加两个虚结点 S (start vertex)和 T (end vertex)，则图 5.1 对应的串行服务组合流程图如图 5.2 所示。本章将研究并提出求解该问题的 PSO-GODSS 算法。

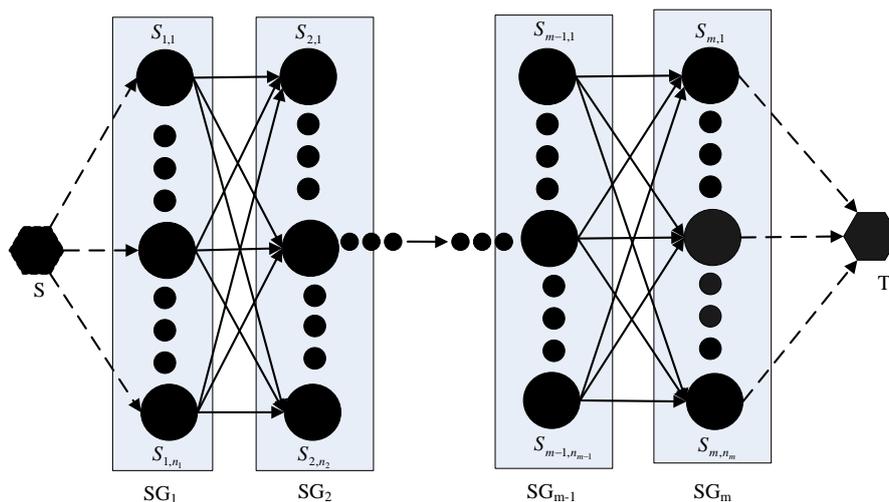


图 5.2 串行服务组合流程图

Fig. 5.2 SCG of sequence service composition process

服务组合流程实际上是一个基于 Web 服务的工作流<sup>[74-76]</sup>，与工作流管理联盟 (WfMC)<sup>[25, 77-78]</sup>定义的 4 种基本模型(即串联模型、并联模型、选择模型和循环模型)相对应，大部分服务组合流程均可由这 4 种基本模型组合而成。基于 4 种基本模型的组合服务 QoS 参数及约简规则和 QoS 计算方法，已有一些研究<sup>[4, 11, 13, 18]</sup>。大多是先计算基本模型的 QoS，然后通过恰当的聚合函数将各服务结点的 QoS 合起来构成组合服务的 QoS。本章参考文献[31]中的方法对 4 种模型进行表示和 QoS 参数的计算。基于 Petri 网的流程建模方法，Web 服务组合四种基本模型及其约简规则表示如图 5.3 所示，其中变迁代表服务；变迁的前置库所和后置库所分别表示服务的输入和输出。考虑 Web 服务的 4 种 QoS 参数，即执行时间 T、执行费用 C、信誉等级 Rep 和可靠性 R。设 cs 为组合服务， $s_i$  为组合服务中的单个服务， $s_i$  和 cs 的服务质量分别为  $Q(s_i) = (T_i, C_i, Rep_i, R_i)$ 、 $Q(cs) = (T_{cs}, C_{cs}, Rep_{cs}, R_{cs})$ ，4 种基本模型的 QoS 参数计算方法分别如下：

(1)串联模型：

$$T_{cs} = \sum_{i=1}^n T_i, \quad C_{cs} = \sum_{i=1}^n C_i,$$

$$Rep_{cs} = \sum_{i=1}^n Rep_i / n, \quad R_{cs} = \prod_{i=1}^n R_i;$$

(2) 并联模型:

$$T_{cs} = \text{Max}\{T_i\}, \quad C_{cs} = \sum_{i=1}^n C_i,$$

$$\text{Re } p_{cs} = \sum_{i=1}^n \text{Re } p_i / n, \quad R_{cs} = \text{Min}\{R_i\};$$

(3) 选择模型: 设第  $i$  个分支被选的概率是  $\alpha_i$ ,  $\sum_{i=1}^n \alpha_i = 1$ ,

$$T_{cs} = \sum_{i=1}^n T_i \alpha_i, \quad C_{cs} = \sum_{i=1}^n C_i \alpha_i,$$

$$\text{Re } p_{cs} = \sum_{i=1}^n \text{Re } p_i \alpha_i, \quad R_{cs} = \prod_{i=1}^n R_i \alpha_i;$$

(4) 循环模型: 设循环次数为  $k$

$$T_{cs} = k * \sum_{i=1}^n T_i, \quad C_{cs} = k * \sum_{i=1}^n C_i,$$

$$\text{Re } p_{cs} = \sum_{i=1}^n \text{Re } p_i / n, \quad R_{cs} = \prod_{i=1}^n R_i.$$

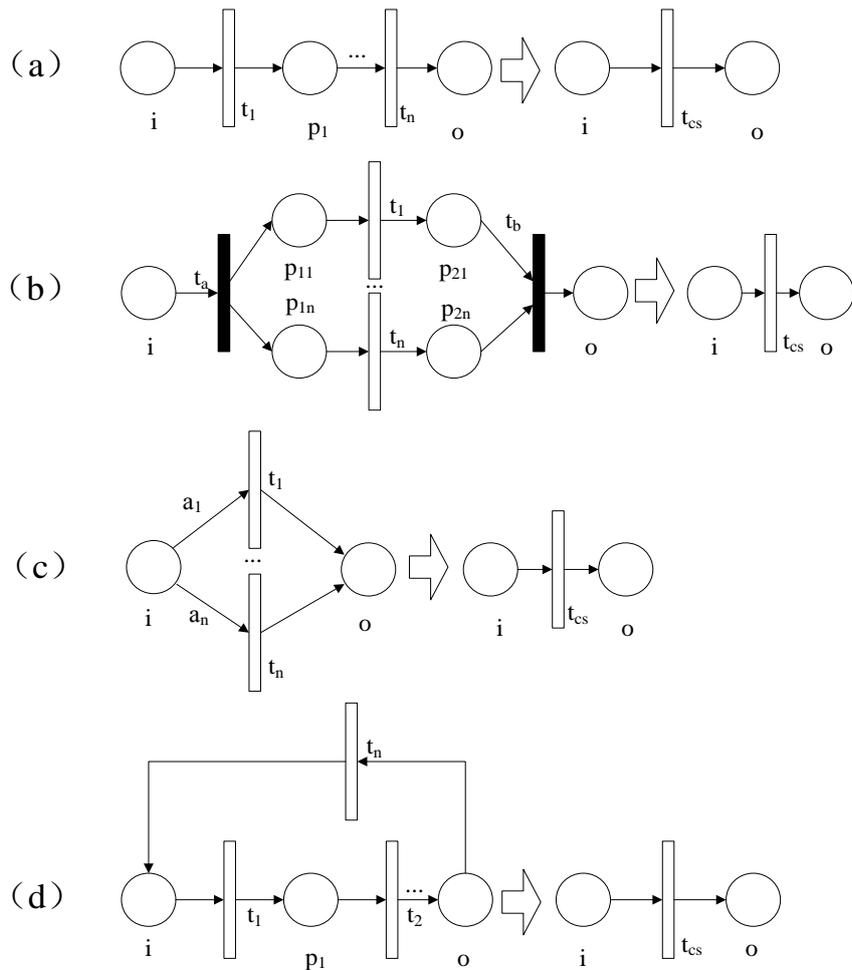


图 5.3 服务组合流程基本模型及约简规则

Fig. 5.3 Basic services composition process models and its reduction rules

## 5.2 PSO-GODSS 算法描述

### 5.2.1 MCOOP 形式化描述

为说明方便,本章将组合服务的执行费用和执行时间作为两目标准则,信誉等级和可靠性作为两约束条件,  $Re p_0$  和  $R_0$  分别表示组合服务所要求的最低信誉等级和最低可靠性,则一个带约束的多目标服务组合优化模型的形式化描述如下:

$$\text{Min } F(P) = \{[T(P), C(P)]\} \quad (5.1)$$

$$\text{s.t.} \begin{cases} Re p(P) \geq Re p_0 \\ R(P) \geq R_0 \end{cases} \quad (5.2)$$

其中,  $T(P)$ ,  $C(P)$ ,  $Re p(P)$  和  $R(P)$  对应服务组合流程的 QoS 参数计算式,根据服务组合模型 QoS 的聚合方法获得;式(5.1)表示取向量的极小化,即使目标  $F(P)$  中的目标函数值同时极小化。实际上 QoS 参数的种类很多,主要思想是将它们看作优化模型的目标函数或约束条件,故本模型可推广到任意多个目标函数和约束条件。

多目标优化问题的一个特点是多个目标函数之间总是相互冲突,即不存在这样一个解,使得所有的目标函数同时达到最优,亦即每个目标函数的最优解是不同的,这成为多目标优化问题求解的一个难点。因此,需要在各目标函数之间取折中,得到满足约束条件的 Pareto 解集,取相对较优的解作为多目标优化问题的解。

### 5.2.2 MCOOP 问题转化为单目标问题

求解多目标规划的最基本方法为评价函数。它的基本思想是借助几何或应用中的直观背景,构造评价函数,将多目标问题转化为单目标优化问题,然后利用成熟的单目标求解方法求出最优解,并把这种最优解当作多目标的最优解<sup>[79]</sup>。本章采用理想点法来构造 MCOOP 问题的评价函数。

$$\text{Min } F(P) = [(T(P) - T^*)^2 + (C(P) - C^*)^2]^{\frac{1}{2}} \quad (5.3)$$

$$\text{s.t.} \begin{cases} Re p(P) \geq Re p_0 \\ R(P) \geq R_0 \end{cases} \quad (5.4)$$

其中,  $(T^*, C^*)$  为一个理想点。理想点是对  $T(P)$  和  $C(P)$  分别求单目标最优值得到的:

$$T^* = \min\{T(P) \mid Re p(p) \geq Re p_0, R(P) \geq P_0\}$$

$$C^* = \min\{C(P) \mid Re p(p) \geq Re p_0, R(P) \geq P_0\}$$

对该多目标服务组合优化问题,根据组合服务 QoS 的聚合方法可知,在各服务群中均选择执行时间最少的服务形成的组合服务的执行时间为  $T^*$ ;同理,在各服务群中均

选择执行费用最低的服务形成的组合服务的执行费用为  $c^*$ 。

### 5.2.3 PSO-GODOSS 算法设计

粒子群优化算法(PSO)是由 Kennedy 和 Eberhart 于 1995 年提出的一种优化技术<sup>[48-49]</sup>。算法最初受到飞鸟和鱼类集群活动的规律性启发,用组织社会行为代替进化算法的自然选择机制,通过种群间个体协作来实现对问题最优解的搜索。具体的进化方程、参数含义及参数设置可参考文献[48-49]。文献[80]证明 MCOOP 问题是一个 NP-Complete 问题。粒子群算法作为一种智能优化方法,具有并行计算和群体寻优的特点。因此,广泛应用于各种 NP-Complete 问题的求解。

对 3.2 节的优化模型,基于粒子群算法,设计相应的参数。设有  $m$  个服务结点,第  $i$  个服务群中有  $n_i$  个服务,  $SG_i = (s_{i,1}, \dots, s_{i,n_i}) (i=1, \dots, m)$ ,  $s_{ij}$  为某个具体服务在第  $i$  个服务群中的编号。为第  $i$  个服务节点选择一个具体的服务  $x_k = s_{ik}$ , 则组合服务方案构成一个  $m$  维向量  $X_i = (x_i^1, x_i^2, \dots, x_i^m)$ ,  $x_i^k \in [1, n_k]$ 。

在基本粒子群算法中速度位移模型针对的是连续型的数值求解<sup>[50, 52-53]</sup>, 而从  $x_i$  的编码可看出,本章的组合优化问题是离散型问题,要对粒子的速度位移模型进行改进,保证微粒在整数空间内飞行。因此,为保证进化搜索在整数空间中进行,将速度位移模型进行改进如下:

$$v_{id}(t+1) = \text{int}(\omega v_{id}(t)) + \phi_1 + \phi_2 \quad (5.5)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (5.6)$$

其中:  $\phi_1 \in [a_1, b_1]$  等概率分布的整数, 即  $p\{\phi_1\} = 1/m_1$ ,  $m_1 = [b_1 - a_1 + 1]$ ;

$\phi_2 \in [a_2, b_2]$  等概率分布的整数, 即  $p\{\phi_2\} = 1/m_2$ ,  $m_2 = [b_2 - a_2 + 1]$ 。

$$a_1 = \begin{cases} 0 & p_{id} > x_{id}(t) \\ c_1(p_{id} - x_{id}(t)) & p_{id} \leq x_{id}(t) \end{cases} \quad b_1 = \begin{cases} c_1(p_{id} - x_{id}(t)) & p_{id} \leq x_{id}(t) \\ 0 & p_{id} > x_{id}(t) \end{cases}$$

$$a_2 = \begin{cases} 0 & p_{gd} > x_{id}(t) \\ c_2(p_{gd} - x_{gd}(t)) & p_{gd} \leq x_{id}(t) \end{cases} \quad b_2 = \begin{cases} c_2(p_{gd} - x_{gd}(t)) & p_{gd} \leq x_{id}(t) \\ 0 & p_{gd} > x_{id}(t) \end{cases}$$

式(5.5)和式(5.6)反映了基本 PSO 算法进化计算的基本思想,同时将进化控制在整数空间内。根据组合服务 QoS 聚合方法和位置向量  $x_i$  的编码以及速度位移模型的改进,结合粒子群优化思想,设计出求解 MCOOP 问题的 PSO-GODOSS 算法,算法描述如下:

**Step1:** 将 MCOOP 问题转化为单目标优化问题;

**Step2:** 初始化  $N$  个粒子, 设置相应的参数;

**Step3:** 将粒子的位置解码成具体的组合服务方案解, 判断该方案是否满足约束条件。

若满足，则按式(5.3)计算该粒子的适应值；否则，赋予该粒子最差的一个适应值；

Step4: 对粒子的当前位置  $x_i$ ，将其适应值与其经历过的局部最佳位置  $P_i$  的适应值作比较，如果较好，则将  $x_i$  作为该粒子当前的局部最佳位置  $P_i$ ；

Step5: 对粒子局部最佳位置  $P_i$ ，将其适应值与所有粒子经历过的全局最佳位置  $P_g$  的适应值作比较，如果较好，则将  $x_i$  作为当前的局部最佳位置  $P_g$ ；

Step6: 根据式(5.5)和式(5.6)更新粒子速度和位置，对新产生位置  $x_i$  和速度  $v_i$  进行范围检查，限制粒子的搜索范围和最大速度；

Step7: 重复 Step3 到 Step6，直到种群中所有粒子都执行完毕；

Step8:  $t=t+1$ ；

Step9: 判断是否满足终止条件(迭代次数)，不满足则返回 Step3。

### 5.2.4 算法时间复杂度分析

算法的时间复杂度主要由粒子群算法进行迭代的时间复杂度组成。假设迭代次数为  $T$ ，粒子群规模为  $N$ ，服务结点数为  $m$ ，则算法总的时间复杂度为  $O(mN^2T)$ ，而多目标遗传算法 GODSS 的总时间复杂度为  $O((mN^2 + NN^*)T)$ ，其中  $N^*$  为辅助种群规模。可知多目标遗传算法的时间复杂度要高。GODSS 算法在每次迭代赋予染色体适应值时，需对个体之间的优于关系进行排序，增加了算法的复杂度，而本章采用理想点的方法很好地将多目标转换为单目标，只需将个体适应值同自身的局部最佳位置的适应值和全局最佳位置的适应值比较即可。因此，从理论分析可知，PSO-GODSS 算法规则简单，易于实现，收敛速度快，在相同迭代次数下，算法的执行时间更少，证明该算法的可行性。

### 5.3 仿真实验及分析

本章针对文献[31]的例子进行仿真实验，实验的组合服务流程图如下：

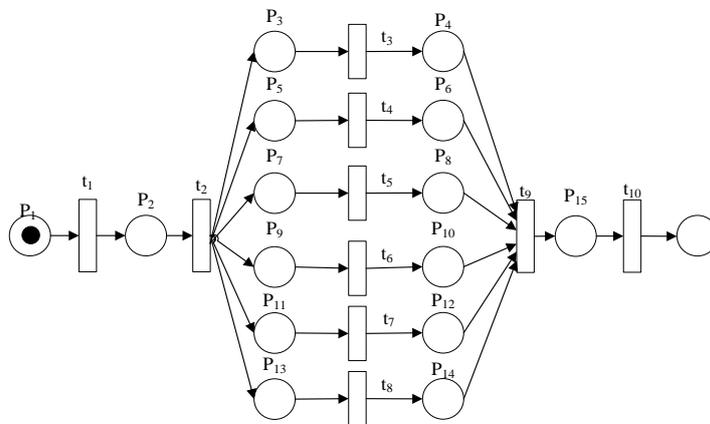


图 5.4 服务组合流程仿真实例

Fig. 5.4 Example of Web service composition process

流程中的每个服务结点对应一个服务群，各服务群中服务的 QoS 参数采用随机的方法在一定范围内生成，并使用极差变换方法对其进行标准化(见 3.3 节)。对于信誉等级，利用模糊数学中隶属度的方法进行量化(见 4.2 节)。根据服务组合流程基本模型以及 QoS 聚合方法，得到目标函数和约束函数。由基本模型组合服务的 QoS 聚合方法可知，组合服务的可靠性约束不宜设太大，故本章中设  $Rep_0 = 0.8$ ， $R_0 = 0.5$ 。该实验通过算法的执行时间和收敛速度来验证 PSO-GODSS 的有效性。仿真实验的微机配置为 PentiumD 3400MHz 处理器，1G 内存，操作系统为 Windows XP，算法用 Matlab7.1 实现。实验考虑服务群规模为 10，迭代次数分别为 100，200，300，400 的情况下与多目标遗传算法 GODSS 进行对比，如图 5.5 所示，发现其执行时间优于 GODSS，从执行时间上说明了 PSO-GODSS 算法的有效性。进一步地，与多目标遗传算法 GODSS 比较每一代种群的平均适应度。如图 5.6 所示，发现 PSO-GODSS 算法收敛速度优于 GODSS 算法。GODSS 算法在第 83 代收敛到最优值，而 PSO-GODSS 算法在第 51 代就已经收敛到了最优值，从收敛速度上说明了 PSO-GODSS 算法的有效性。

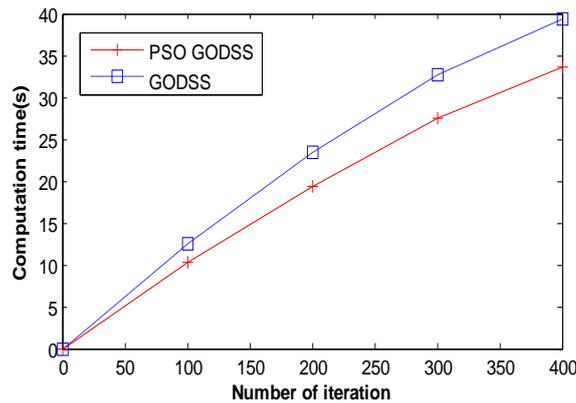


图 5.5 算法执行时间比较(PSO-GODSS VS GODSS)

Fig. 5.5 Computation time of PSO-GODSS VS GODSS

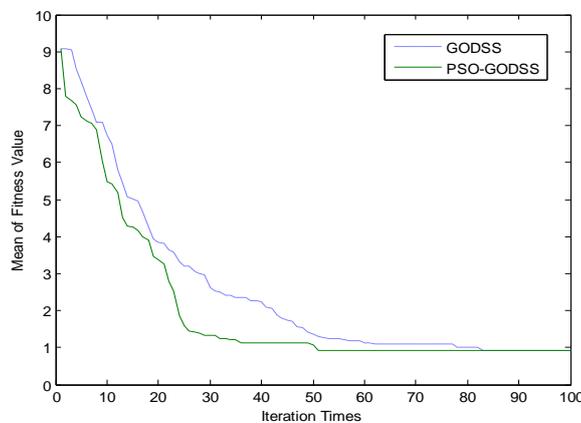


图 5.6 收敛速度的比较(PSO-GODSS VS GODSS)

Fig. 5.6 Convergence rate of PSO-GODSS VS GODSS

另外, 在文献[27]中, Alrifai 等人提出在 Skyline 服务集合中进行服务选择的方法, 通过淘汰掉 QoS 较差的候选原子服务, 从而降低候选服务的个数, 提高服务选择的效率。具体关于 Skyline 服务集合的概念详见文献[27,44]。同理, 在本文中, 对于每个服务结点, 我们考虑只在其相对应的服务群的 Skyline 服务集合中进行服务选择, 可进一步提高 PSO-GODSS 算法的执行效率。

## 5.4 本章小结

本章针对组合服务中 QoS 全局最优动态 Web 服务选择问题, 基于粒子群算法, 提出并设计了求解该问题的 PSO-GODSS 算法。将服务动态选择全局优化问题转化为一个带 QoS 约束的多目标服务组合优化问题, 通过理想点的方法将多目标向单目标转化, 利用粒子群算法的智能优化原理进行优化, 最终产生一组满足约束条件的优化服务组合流程集, 能够更好地满足用户的需求。实验结果表明该算法的可行性和有效性, 且算法执行效率优于多目标遗传算法。

## 第六章 结论与展望

### 6.1 结论

Web 服务是新形式的 Internet 软件, 统一使用 Internet 协议布置和调用, 来自不同服务提供商的服务可被整合以提供组合服务。随着 Web 服务技术的快速发展, 具有相同或相似功能的 Web 服务逐渐增多, 在 Web 服务合成中的一个主要问题不再是能否找到满足用户功能需求的服务, 而是如何从满足用户功能需求但服务质量各异的服务集中选择出最符合用户需求的服务。因此, 基于 QoS 的 Web 服务选择问题成为当前的研究热点。本文主要研究了同一时刻存在多个请求同种功能服务的全局优化 Web 服务选择和组合服务中 QoS 全局优化动态 Web 服务选择这两个问题, 主要研究成果和结论如下:

(1) 针对同一时刻存在多个请求同种功能服务情形的 Web 服务选择问题, 以网格服务环境为例, 从单一时刻的优化角度出发, 研究了全局优化 Web 服务选择模型。利用层次分析法计算出服务请求者对各服务的 QoS 满意度与服务信誉满意度, 将二者结合起来, 计算出综合满意度。基于综合满意度进行 0-1 整数规划, 使服务请求对 Web 服务的综合满意度之和最大, 建立了全局优化服务选择模型。仿真实验, 验证了该方法的有效性。

(2) 进一步地, 考虑资源的充分利用, 保证服务负载平衡。本文从服务请求者和服务提供者双方的角度以及长时间的角度考虑多服务请求情形的全局最优 Web 服务选择。基于带权的 Euclidean 距离计算 Web 服务对各服务请求的距离, 即匹配度。基于匹配度进行 0-1 整数规划, 使服务请求对 Web 服务的匹配度之和最大, 建立全局优化服务选择模型, 使得从长时间的角度看服务资源能够得到合理的利用。结合实际提出通用可行的解决多服务请求的 Web 服务全局优化服务选择算法 GOSSMR, 使最多的 Web 服务请求需求得到满足但又尽可能不浪费资源, 保证资源的合理分配和利用, 避免负载问题的发生, 提高系统的性能。为提高算法的效率, 基于 Skyline 的方法提出 Skyline GOSSMR 改进算法。通过降低候选服务的规模, 大大提高算法的执行效率。同时, 分析了算法的时间复杂度。仿真实验比较两种算法的执行效率, 验证了模型算法的可行性和有效性。

(3) 研究了组合服务中 QoS 全局优化动态 Web 服务选择模型及算法。基于粒子群进化算法, 设计了一种用于解决该问题的 PSO-GODSS 算法。算法的主要思想是将问题表示为一个带 QoS 约束的多目标服务组合优化问题, 通过理想点的方法将多目标向单目标转化, 然后利用粒子群算法的智能优化原理进行算法设计及求解, 最终产生一组满足约束条件的优化服务组合流程集。理论分析和实验结果表明了该算法的可行性和有效性, 且算法的执行效率和收敛速度优于以往的多目标遗传算法。相对多目标遗传算法,

本文提出的方法规则简单,易于编程实现,收敛速度快,搜索的全局性更好,能够快速找到全局最佳服务组合决策,在实际应用中是可行的。

## 6.2 展望

本文在 Web 服务选择相关研究的基础上做了进一步的研究,分析了现有的 Web 服务选择方法的不足,既提出了新的问题,也给出了相应的模型及算法。基于本文的研究工作基础,关于 Web 服务选择的研究,我们将从以下几个方面开展下一步的研究工作:

(1) 本文虽研究了同一时刻存在多个请求同种功能服务的全局优化 Web 服务选择问题,但未考虑一个时段内的全局优化服务选择问题。对于该问题,还存在一些难题亟待解决,例如:如何确定时段的大小,使得所有请求者能够接受又能使资源利用率达到最大化等。这需要研究用户的心理,研究用户对时间的等待限度。从而考虑更多的服务请求,在用户的时间等待限度内考虑全局优化服务选择问题。

(2) 随着服务数量的增加,在求得的 Skyline 服务集中的服务数量也会逐渐增加。这个时候,可能相对服务请求数量,候选服务的数量还是较多。因此,需要进一步提高算法的执行效率,满足系统的实时性。对于该问题,可以考虑采用智能进化算法求解整数规划问题。如遗传算法、蚁群算法、粒子群算法、差异演化算法、细菌觅食算法等优化算法,进一步满足算法的可行性要求。

(3) 由于服务 QoS 属性之间存在相关性,并且由于偏好的主观性和模糊性,用户无法用具体数字表达清楚自己的偏好,现有的利用用户对各维 QoS 的偏好对服务 QoS 进行加权的 Web 服务选择方法在实际应用中尚存在问题。一方面,可以研究如何将模糊偏好向权重转化的方法,从而使得基于 QoS 加权的 Web 服务选择方法在实际中有效,但该方法的准确性还有待验证;另一方面,可以利用主成分分析方法,克服 QoS 各维属性之间可能存在的相关性,有效地评价 Web 服务的综合质量,为 Web 服务选择提供依据。

(4) 在 Web 服务环境下,Web 服务具有动态的特征,例如:新服务的加入、原有服务的实效、服务 QoS 的改变等。应该考虑实时地对 Web 服务进行测试,尽量在服务选择之前或至少在服务调用之前监测出服务的有效性。而不是在服务调用之后,发现服务失效,然后再重新进行选择,那样就增加了服务选择的时间。因此,研究 Web 服务的实时自动化测试成为必要。

(5) 由于动态网络环境的不确定性,服务提供者对 QoS 的发布数据以及请求者对 QoS 的反馈数据的真实可信性在实际中难以保证。因此,对数据的可信性评价及评价的准确性有待进一步研究。应考察服务调用的历史记录,与服务的 QoS 进行比对,对服务的 QoS 不断进行修正,从而获得更加准确的服务 QoS 数据。另一方面,对服务的调用进行监控,尽量防止用户的虚假评价。

## 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] L. Zeng, B. Benatallah, H. Lei, A. Ngu, D. Flaxer, H. Chang. Flexible composition of enterprise web services[J]. *Electronic Markets*, 2001, 13(2): 141-152.
- [3] H. Wang, J. Huang, Y. Qu, J. Xie. Web services: problems and future directions[J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2004, 1(3): 309-320.
- [4] S. Ran. A model for web services discovery with QoS[J]. *ACM SIGecom Exchanges*, 2003, 4(1): 1-10.
- [5] D.A. Menasce. Composing web services: A QoS view[J]. *Internet Computing, IEEE*, 2004, 8(6): 88-90.
- [6] K. Sycara, M. Paolucci, A. Ankolekar, N. Srinivasan. Automated discovery, interaction and composition of semantic web services[J]. *Journal of Web Semantics*, 2003, 1(1): 27-46.
- [7] P. Traverso, M. Pistore. Automated composition of semantic web services into executable processes[J]. *The Semantic Web CISWC 2004*, 2004: 380-394.
- [8] E. Sirin, B. Parsia, J. Hendler. Filtering and selecting semantic web services with interactive composition techniques[J]. *IEEE intelligent systems*, 2004: 42-49.
- [9] N. Milanovic, M. Malek. Current solutions for web service composition[J]. *Internet Computing, IEEE*, 2004, 8(6): 51-59.
- [10] 牟玉洁, 曹健, 张申生, 张建宏. 面向服务质量的交互式 Web 服务选择[J]. *计算机集成制造系统*, 2006, 12(5): 789-794.
- [11] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q.Z. Sheng. Quality driven web services composition[C]. *Proceedings of In Proceedings of the International World Wide Web Conference. ACM*, 2003: 411-421.
- [12] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, H. Chang. QoS-aware middleware for web services composition[J]. *IEEE Transactions on software engineering*, 2004, 30(5): 311-327.
- [13] Y. Liu, A.H. Ngu, L.Z. Zeng. QoS computation and policing in dynamic web service selection[C]. *Proceedings of In Proceedings of the International World Wide Web Conference. ACM*, 2004: 66-73.
- [14] E.M. Maximilien, M.P. Singh. Agent-based architecture for autonomic web service selection[J]. *Journal on Web Semantics*, 2003: 261-279.
- [15] E.M. Maximilien, M.P. Singh. Toward autonomic web services trust and selection[C]. *Proceedings of, ACM*, 2004: 212-221.
- [16] E.M. Maximilien, M.P. Singh. A framework and ontology for dynamic web services selection[J]. *Internet Computing, IEEE*, 2004, 8(5): 84-93.
- [17] T. Yu, K.J. Lin. Service selection algorithms for Web services with end-to-end QoS constraints[J]. *Information Systems and E-Business Management*, 2005, 3(2): 103-126.
- [18] T. Yu, Y. Zhang, K.J. Lin. Efficient algorithms for Web services selection with end-to-end QoS constraints[J]. *ACM Transactions on the Web (TWEB)*, 2007, 1(1): 6-32.
- [19] D. Ardagna, B. Pernici. Global and local QoS guarantee in web service selection[C]. *Proceedings of Third International Conference on Business Process Management. Springer*, 2006: 32-46.
- [20] M. Alrifai, T. Risse. Combining global optimization with local selection for efficient QoS-aware service composition[C]. *Proceedings of, ACM*, 2009: 881-890.
- [21] L. Qi, Y. Tang, W. Dou, J. Chen. Combining Local Optimization and Enumeration for QoS-aware Web Service Composition[C]. *Proceedings of International Conference on Web Service. IEEE*, 2010: 34-41.
- [22] X. Chen, X. Liu, Z. Huang, H. Sun. RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation[C]. *Proceedings of International Conference on Web Services. IEEE*, 2010: 9-16.
- [23] Z. Zheng, H. Ma, M.R. Lyu, I. King. Wsrec: A collaborative filtering based web service recommender system[C].

- Proceedings of International Conference on Web Services. IEEE, 2009: 437-444.
- [24] D. Dyachuk, R. Deters. Scheduling of composite web services[C]. Proceedings of On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Springer, 2006: 19-20.
- [25] S. Shahand, S.J. Turner, W. Cai, H. Khademi. DynaSched: a dynamic Web service scheduling and deployment framework for data-intensive Grid workflows[J]. Procedia Computer Science, 2010, 1(1): 593-602.
- [26] B. Benatallah, M. Dumas, Q.Z. Sheng, A.H.H. Ngu. Declarative composition and peer-to-peer provisioning of dynamic web services[C]. Proceedings of Proceedings of the 18th International Conference on Data Engineering. IEEE, 2002: 297-308.
- [27] M. Alrifai, D. Skoutas, T. Risse. Selecting skyline services for QoS-based web service composition[C]. Proceedings of In Proceedings of the International World Wide Web Conference. ACM, 2010: 11-20.
- [28] F. Casati, S. Ilnicki, L.J. Jin, V. Krishnamoorthy, M.C. Shan. Adaptive and dynamic service composition in eFlow[C]. Proceedings of, Springer, 2000: 13-31.
- [29] 邢庆秀. 支持 QoS 全局优化的动态 Web 服务组合问题研究[D]. 青岛: 中国海洋大学, 2008.
- [30] 范小芹, 蒋昌俊, 方贤文, 丁志军. 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, (001): 147-156.
- [31] 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.
- [32] 吴朝晖, 邓水光, 吴健. 服务计算与技术[M]. 杭州: 浙江大学出版社, 2009.
- [33] M.P. Papazoglou, D. Georgakopoulos. Service-oriented computing[J]. Communications of the ACM, 2003, 46(10): 25-28.
- [34] A. Tsalgaidou, T. Pilioura. An overview of standards and related technology in Web services[J]. Distributed and Parallel Databases, 2002, 12(2): 135-162.
- [35] D. Fensel, C. Bussler. The web service modeling framework WSMF[J]. Electronic Commerce Research and Applications, 2002, 1(2): 113-137.
- [36] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI[J]. Internet Computing, IEEE, 2002, 6(2): 86-93.
- [37] R. Li, Y. Li, K. Kristiansen, J. Wang. SOAP: short oligonucleotide alignment program[J]. Bioinformatics, 2008, 24(5): 713.
- [38] P. Rompothong, T. Senivongse. A query federation of UDDI registries[C]. Proceedings of, Trinity College Dublin, 2003: 561-566.
- [39] 袁震东, 蒋鲁敏, 束金龙. 数学建模简明教程[M]. 上海: 华东师范大学出版社, 2002.
- [40] T.L. Saaty. How to handle dependence with the analytic hierarchy process[J]. Mathematical modelling, 1987, 9(3-5): 369-376.
- [41] J.S. Dyer. Remarks on the analytic hierarchy process[J]. Management science, 1990, 36(3): 249-258.
- [42] 梅长林, 范金城. 数据分析方法[M]. 北京: 高等教育出版社, 2007.
- [43] S. B Rzs Nyi, D. Kossmann, K. Stocker. The skyline operator[C]. Proceedings of Proceedings of International Conference on Data Engineering (ICDC). Citeseer, 2001: 421-430.
- [44] 吴健, 陈亮, 邓水光, 李莹, 邝砾. 基于 Skyline 的 QoS 感知的动态服务选择[J]. 计算机学报, 2010, 33(11): 2136-2146.
- [45] M. Morse, J.M. Patel, W.I. Grosky. Efficient continuous skyline computation[J]. Information Sciences, 2007, 177(17): 3411-3437.
- [46] D. Papadias, Y. Tao, G. Fu, B. Seeger. An optimal and progressive algorithm for skyline queries[C]. Proceedings of Proceedings of the 2003 ACM SIGMOD international conference on Management of data. ACM, 2003: 467-478.

- [47] D. Kossmann, F. Ramsak, S. Rost. Shooting stars in the sky: An online algorithm for skyline queries[C]. Proceedings of VLDB Endowment, 2002: 275-286.
- [48] J. Kennedy, R. Eberhart. Particle swarm optimization[C]. Proceedings of International Conference on Neural Networks. IEEE, 1995: 1942-1948.
- [49] Y. Shi. Particle swarm optimization[J]. IEEE Neural network society, 2004, 2: 8-13.
- [50] 曾建潮, 介婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004.
- [51] R.C. Eberhart, Y. Shi. Particle swarm optimization: developments, applications and resources[C]. Proceedings of, Piscataway, NJ, USA: IEEE, 2001: 81-86.
- [52] 廖宁. 基于微粒群算法的服务网格资源优化调度问题的研究[D]. 湘潭: 湖南科技大学, 2009.
- [53] 刘兰霞. 多目标粒子群优化算法研究[D]. 湘潭: 湖南科技大学, 2010.
- [54] P.Y. Yin, F. Glover, M. Laguna, J.X. Zhu. Cyber swarm algorithms-improving particle swarm optimization using adaptive memory strategies[J]. European Journal of Operational Research, 2010, 201(2): 377-389.
- [55] W.N. Chen, J. Zhang, H.S.H. Chung, W.L. Zhong, W.G. Wu, Y.H. Shi. A novel set-based particle swarm optimization method for discrete optimization problems[J]. Evolutionary Computation, IEEE Transactions on, 2010, 14(2): 278-300.
- [56] J. Sun, W. Xu, J. Liu. Parameter selection of quantum-behaved particle swarm optimization[J]. Advances in Natural Computation, 2005: 543-552.
- [57] I.C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 85(6): 317-325.
- [58] S. Liu, Y. Liu, N. Jing, G. Tang, Y. Tang. A dynamic web service selection strategy with QoS global optimization based on multi-objective genetic algorithm[J]. Grid and Cooperative Computing-GCC 2005, 2005: 84-89.
- [59] S. Galizia, A. Gugliotta, J. Domingue. A trust based methodology for web service selection[C]. Proceedings of International Conference on Web Service. IEEE Computer Society, 2009: 881-890.
- [60] A. De Moor, W.J. Van Den Heuvel. Web service selection in virtual communities[C]. Proceedings of Proceedings of the 37th Hawaii International Conference on System Sciences. IEEE Computer Society, 2004.
- [61] 康国胜, 刘建勋, 胡蓉. 网络服务环境下的全局最优服务选择模型研究[J]. 小型微型计算机系统, 2010(已录用).
- [62] 刘扬, 何华灿. 一种基于 QoS 的网格服务选择机制[J]. 计算机工程, 2007, 33(7): 33-35.
- [63] O. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, C. Goble. An overview of S-OGSA: A reference semantic grid architecture[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2006, 4(2): 102-115.
- [64] J. Jiang, B.Q. Xia, Z.K. Pan, X.Z. Yin. Dynamic Grid Model for Service Integration and Schedule Based on SOA[J]. Jisuanji Gongcheng/ Computer Engineering, 2009, 35(22): 68-70.
- [65] 李研, 周明辉, 李瑞超, 曹东刚, 梅宏. 一种考虑 QoS 数据可信性的服务选择方法[J]. 软件学报, 2008, 19(10): 2620-2627.
- [66] 王小英, 赵海, 林涛, 张文波, 尹震宇. 基于信任的普适计算服务选择模型[J]. 通信学报, 2005, 26(5): 1-8.
- [67] 姜启源, 谢金星, 叶俊. 数学模型[M]. 北京: 高等教育出版社, 2003.
- [68] G. Kang, J. Liu, M. Tang, X. Liu, K. Fletcher. Web Service Selection for Resolving Conflicting Service Requests[C]. Proceedings of International Conference on Web Service. 2011(accepted).
- [69] Web 服务超市. 2010 知识处理与网络化制造湖南省普通高校重点实验室 [EB/OL]. <http://125.221.225.2:8080/WSSM/>, 2010. 4.
- [70] 杨纶标, 高英仪. 模糊数学原理及应用(第四版)[M]. 广州: 华南理工大学出版社, 2006.
- [71] W.M.P. Van Der Aalst. Don't go with the flow: Web services composition standards exposed[J]. IEEE intelligent systems, 2003, 18(1): 72-76.
- [72] J. Rao, X. Su. A survey of automated web service composition methods[J]. Semantic Web Services and Web Process Composition, 2005: 43-54.

- [73] P. Pires, M. Benevides, M. Mattoso. Building reliable web services compositions[J]. *Web, Web-Services, and Database Systems*, 2009: 59-72.
- [74] J. Cardoso, A. Sheth, J. Miller, J. Arnold, K. Kochut. Quality of service for workflows and web service processes[J]. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2004, 1(3): 281-308.
- [75] J. Han, Y. Cho, E. Kim, J. Choi. A ubiquitous workflow service framework[J]. *Computational Science and its Applications-ICCSA 2006*, 2006: 30-39.
- [76] R. Hamadi, B. Benatallah. A Petri Net-based Model for Web Service Composition[C]. *Proceedings of 14th Australasian Database Conference (ADC 2003)*. Australian Computer Society, 2003: 191-200.
- [77] P. Senkul, I.H. Toroslu. An architecture for workflow scheduling under resource allocation constraints[J]. *Information Systems*, 2005, 30(5): 399-422.
- [78] T. Glatard, J. Montagnat, X. Pennec. Grid-enabled workflows for data intensive medical applications[J]. 2005.
- [79] 刘国平, 曾强. 多目标最优化的粒子群算法[J]. *杭州师范学院学报(自然科学版)*, 2005, 4(1): 30-33.
- [80] M.R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*[M]: WH Freeman & Co. New York, NY, USA, 1979.

## 致 谢

首先，我衷心地感谢我的导师刘建勋教授，感谢他三年对我无微不至的关怀和孜孜不倦的教诲。刘老师渊博的知识、严谨的治学态度、敏锐的学术洞察力、对事物深刻的理解力、开阔灵活的思路和简洁有力的表达都让我受益匪浅，让我知道如何做人、做事、做学问。刘老师给我提供参与众多科研项目和学术会议的机会，拓展我的知识和眼界。还为我提供优良的科研环境，帮助我进行论文的选题，分析论文写作中遇到的问题及解决方法等等，使我能顺利地完成论文的撰写工作。在刘老师的悉心培养和指导下，我对做科学研究有了更深的认识，同时也积累了丰富的经验，这些都将让我受益终生。

其次，感谢知识处理与网络化制造湖南省普通高校重点实验室的各位老师和同学，在他们帮助下，使得我的研究生学习得以顺利的开展。和他们讨论问题，寻求解决方法，是我最大的快乐。他们分别是：唐明董、胡蓉、文一凭、曹步青、赵肆江、刘毅志、廖祝华、廖宁、陈平、冯细光、杨树人、姜叶春、侯林娇、罗叶飞、金永平、钟新利、李欢欢、徐宇等老师和同学。

特别感谢两位国外的教授。一个是加拿大渥太华大学信息科学与工程学院 **Gregor v. Bochmann** 教授，在他来实验室指导的 3 个月期间，给我诸多的指导和鼓励，让我对学术产生了兴趣；另一个是美国密苏里科技大学计算机系刘小青教授，他为我的论文提出很多建设性的修改意见，并亲自为我做了一些修改，使我更懂得做事要认真真、脚踏实地的道理。让我再次致以他们两位最崇高的谢意。

另外，感谢几位外籍教师，给予我在外语学习上的帮助。他们是湖南科技大学外国语学院 **Pam**、**Arlene**、**Marla** 和湘潭大学外国语学院的 **Jason**。同时，感谢恩溢之家的众多弟兄姊妹，是他们给我心灵上的安慰和快乐，带给我生命的成长。他们是 **Jasmine**、**Portia**、**Mary**、**Daniel**、**Peter** 等。

最后，要感谢我的家人，给予我无怨无悔的付出和毫无保留的支持鼓励，我才能坚定的走完这段特殊的心路历程。他们是我的爸爸妈妈、我的妹妹、我的哥哥嫂嫂和我那乖巧的侄子。感谢你们的陪伴，我的世界因你们而充满爱。

人生是一个不断自我完善的过程，在此过程中凡是给过我关心和帮助的人我都记在心上，也在此表示感谢，无论是上面提到的还是没有提到的。

限于水平有限，论文存在错误和不足之处，欢迎批评和指正。

康国胜

2011.4.25



## 附录 A（攻读学位期间发表论文目录）

- [1] 康国胜, 刘建勋, 胡蓉. 网格服务环境下的全局最优服务选择模型研究[J]. 小型微型计算机系统, 2010(已录用).
- [2] Guosheng Kang, Jianxun Liu, Mingdong Tang, Xiaoqing(Frank) Liu, Kenneth K Fletcher. Web Service Selection for Resolving Conflicting Service Requests[C]. International Conference on Web Service (ICWS 2011), 2011(Regular Paper, Accepted).
- [3] 康国胜, 刘建勋, 唐明董, 徐宇. QoS 全局最优动态 Web 服务选择算法[C]. 2011 年中国计算机学会服务计算学术会议 (CCF NCSC 2011), 2011(已录用).
- [4] 康国胜, 刘建勋, 唐明董, 徐宇. 基于差异演化算法的 QoS 全局最优动态 Web 服务选择[C]. 第二届中国云计算与 SaaS 学术会议 (C<sup>4</sup>S<sup>2</sup> 2011), 2011(已录用).



## 附录 B（攻读学位期间所获的奖项）

- [1] 第六届全国研究生数学建模竞赛国家三等奖，2009.12
- [2] 湖南省第三届研究生创新论坛优秀论文一等奖，2010.10
- [3] 湖南省第三届研究生创新论坛优秀论文二等奖，2010.10



## 附录 C（攻读学位期间参加研究项目）

1. 国家自然科学基金项目“基于行为及其可信扩展的可信服务组合方法研究”，2009.1-2011.12(项目编号：90818004)
2. 教育部新世纪人才项目“基于 SOA 的企业间协同服务技术研究”，2009.12-2012.12(项目编号：NCET-10-0140)
3. 湖南省教育厅资助项目“基于信誉策略的可信 Web 服务合成模型与方法研究”，2010.9-2012.12(项目编号：09k085)