

学校代码: 10286

分 类 号: TP311

密 级: 公开

UDC: 004.4

学 号: 131455



东南大学

硕士学位论文

基于可信度计算与 QoS 相结合的 Web 服务组合 优化问题研究

研究生姓名: 杨丹榕

导 师 姓 名: 王红兵

申请学位类别 工学硕士

学位授予单位 东南大学

一级学科名称 计算机科学与技术

论文答辩日期 2016 年 月 日

二级学科名称 计算机软件与理论

学位授予日期 2016 年 月 日

答辩委员会主席

评 阅 人

2016 年 月 日

OPTIMIZATION STUDY FOR WEB SERVICE COMPOSITION BY INTEGRATING QOS AND CREDIBILITY COMPUTING

A Thesis Submitted to
Southeast University
For the Academic Degree of Master of Engineering

BY

YANG Dan-rong

Supervised by

Prof. WANG Hong-bing

School of Computer Science and Engineering
Southeast University
May, 2016

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：_____日期：_____

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：_____导师签名：_____日期：_____

摘要

Web 技术与软件工程的结合使得面向服务的体系架构(Service Oriented Architecture, SOA)成为当前热门的新型计算模式。Web 服务是 SOA 中的一种重要形式且在多个领域广泛应用。该技术的快速发展与应用使得 Web 服务的数量与日俱增,单个 Web 服务由于功能有限已不能满足用户复杂的应用请求。因此,为充分利用服务资源、通过组合现存 Web 服务来开发新的系统逐渐受到关注,学术界也相继出现了大量的 Web 服务组合技术的相关研究。其中基于 QoS 的 Web 服务组合成为当前研究的热点。其目的是在系统的功能需求被满足的前提下,根据非功能属性即服务质量(Quality of Service, QoS)求得最优的 Web 服务组合方案。

然而,开放的互联网上提供的服务纷繁复杂,服务均处在一个动态变化的环境中,因缺少监管多少都会存在欺诈行为,传统服务选择方法得到的高 QoS 并不能保证服务的可信性,这将导致最终组合出的组合服务也无法确保用户可以满意。不可信的服务会给企业应用系统带来不可预知的后果,只有既满足服务的高质量又保证其可信性,才能确保服务最符合用户需要。因此,针对这一问题,本文提出了一种基于可信度计算与 QoS 相结合的 Web 服务组合优化方法。

具体地说,本文在已有研究的基础上引入新的可信度概念,从客观和主观两个方面对 Web 服务的 QoS 属性进行可信度评估和权重计算,提出融合了可信度的 QoS 度量模型。在局部优化阶段,根据评估出的可信 QoS 度量值对候选服务进行筛选,然后再根据挑选出的可信服务进行后续的全局组合优化。在全局优化阶段,将服务组合问题建立为多目标优化模型,并对当前最新的元启发式优化算法—布谷鸟搜索算法进行了改进,采用改进的布谷鸟多目标搜索算法对问题进行求解,以得到全局最优的服务组合。最后本文进行了一系列的仿真实验来评估验证这些方法的正确性、有效性。

关键词: Web 服务, 服务组合, QoS, 可信度评估, 多目标优化

Abstract

The combination of Web technology and software engineering makes the Service Oriented Architecture (SOA) become the new computing model. Web service is a typical technology of SOA, which has widely applied in various fields. The number of Web services is growing more and more with the rapid development of this technology and application. A single Web service has already cannot satisfy the user's complicated request because of the limited functionality. In order to make full use of service resources, developing new system through the combination of the existing Web services gets wide attention and a large number of technologies about Web service composition have emerged in academia. Among them, QoS-aware Web service composition is becoming the focus of current research. That is, on the premise of system's functional requirements have been met, according to the non-functional attributes (Quality of Service, QoS) to find the optimal combination scheme.

However, in the open Internet environment, a variety of offered services are all in a dynamic process, there will be fraud more or less due to the lack of regulation, high QoS derived from traditional service selection approach cannot guarantee the credibility of service, result in the final composite service cannot guarantee the satisfaction of user. Dishonest services will bring unpredictable consequences for enterprise application system, so it is necessary to not only guarantee service quality but also meet the demand of credibility, only in this way can ensure that services meet user's requirement most. Therefore, in order to solve above problems, this paper proposes a QoS-aware Web service combination optimization method combined with credibility computing.

Specifically, on the basis of the existing research, this paper raises the new concept of credibility, give credibility evaluation and weights calculation for Web service's QoS attributes from objective aspect and subjective aspect. A QoS measurement integrated of credibility model is put forward. In local optimization phase, filtrate candidate services according to the assessment of credible QoS metrics, and then the selected credible services are applied in subsequent global combinatorial optimization. In the stage of global optimization, the service composition problem is established by multiple objective optimization model, and is solved by an improved cuckoo multi-objective search algorithm, aiming to get the global optimal composite services. At the end of the paper a series of experiments are conducted to assess the correctness and validity of these methods.

Keywords: Web services, service composition, QoS, credibility evaluation, multi-objective optimization

目 录

摘要.....	I
Abstract.....	II
目 录.....	III
本论文专用术语的注释表.....	V
第一章 绪论.....	1
1.1 研究背景.....	1
1.2 问题提出.....	2
1.3 研究现状.....	3
1.3.1 基于 QoS 的 Web 服务组合研究.....	3
1.3.2 Web 服务可信度计算研究.....	3
1.3.3 Web 服务组合多目标全局优化研究.....	4
1.4 研究内容.....	5
1.5 论文组织安排.....	6
第二章 Web 服务及服务组合技术概述.....	8
2.1 什么是 Web 服务.....	8
2.2 Web 服务标准.....	8
2.2.1 Web 服务体系结构.....	8
2.2.2 Web 服务规范.....	9
2.2.3 Web 服务协议栈.....	10
2.3 Web 服务组合.....	12
2.3.1 Web 服务组合定义.....	12
2.3.2 Web 服务组合分类.....	12
2.3.3 基于 QoS 感知的 Web 服务组合.....	14
2.4 Web 服务选择与组合优化.....	16
2.4.1 优化策略.....	16
2.4.2 优化算法.....	16
2.4.3 存在的问题.....	18
2.5 本章小结.....	18
第三章 相关理论基础.....	19
3.1 贝叶斯学习理论.....	19
3.2 粗糙集理论.....	21
3.3 模糊层次分析法.....	22
3.4 布谷鸟搜索算法.....	23
3.5 本章小结.....	26
第四章 融合可信度的 Web 服务 QoS 度量模型及服务选择.....	27
4.1 Web 服务可信度计算模型.....	27
4.2.1 客观信誉评估方法.....	27
4.2.2 主观信任评估方法.....	29
4.2.3 可信度计算.....	30
4.2 QoS 属性权重计算模型.....	31
4.2.1 属性客观权重计算.....	31
4.2.2 属性主观权重计算.....	32

4.2.3 属性综合权重	34
4.3 QoS 计算与服务选择	34
4.4 本章小结	36
第五章 基于布谷鸟算法的 Web 服务组合全局优化	37
5.1 Web 服务组合优化流程	37
5.2 多目标优化模型	38
5.3 改进的多目标布谷鸟搜索算法	41
5.4 本章小结	44
第六章 实验与分析	45
6.1 实验环境与数据集	45
6.2 可信度评估实验验证	45
6.3 服务选择方法对比	47
6.4 改进的布谷鸟搜索算法在 Web 服务组合全局优化过程中的对比	48
6.5 本章小结	52
第七章 总结与展望	53
7.1 本文总结	53
7.1 未来工作	54
致谢	55
参考文献	56
附录 A 攻读硕士学位期间科研成果	61

本论文专用术语的注释表

SOA	Service-Oriented Architecture	面向服务的体系结构
QoS	Quality of Service	服务质量
SOAP	Simple Object Access Protocol	简单对象访问协议
WSDL	Web Service Description Language	网络服务描述语言
UDDI	Universal Description Discovery and Integration	统一描述、发现和集成协议
W3C	World Wide Web Consortium	万维网联盟
HTTP	HyperText Text Transfer Protocol	超文本传输协议
FTP	File Transfer Protocol	文件传输协议
SMTP	Simple Mail Transfer Protocol	简单邮件传输协议
JMS	Java Message Service	Java 消息服务
XML	eXtensible Markup Language	可扩展标记语言
AI	Artificial Intelligence	人工智能
HTN	Hierarchical Task Network	层次任务网
PDDL	Planning Domain Definition Language	规划域定义语言
GA	Genetic Algorithm	遗传算法
ACO	Ant Colony Optimization	蚁群优化
PSO	Particle Swarm Optimization	粒子群优化
RST	Rough Sets Theory	粗糙集理论
AHP	Analytic Hierarchy Process	层次分析法
FAHP	Fuzzy Analytic Hierarchy Process	模糊层次分析法
CS	Cuckoo Search	布谷鸟搜索算法
NSGA	Non-dominated Sorting Genetic Algorithm	非支配排序遗传算法
MOCS	Multi-Objective Cuckoo Search	多目标布谷鸟搜索算法
IMOCS	Improved Multi-Objective Cuckoo Search	改进的多目标布谷鸟搜索算法

第一章 绪论

1.1 研究背景

随着 Internet 信息技术的快速发展, 软件系统的设计与开发都产生了相当大的变化与进步。但是怎样将异构的、遗留的系统进行无缝集成, 仍然是业内存在的一个问题。这就迫切需要一个标准的 Web 协议以解决异构系统间互操作的问题, Web 服务(Web Services)技术便是在这种需要下产生的。Web 技术的进一步发展以及与软件工程的方法深入结合, 使得一种新型计算模型: 面向服务的体系结构(Service Oriented Architecture, SOA)成为当前技术研究的新热点。SOA 更加专注于服务, 汇合在一起的应用程序可以提供内容更充实、用途更明显的业务流程, 从而更真实地反映商业模式。Web 服务作为 SOA 架构的一种重要实现形式, 已得到了工业和学术界的一致认可并广泛应用在多个领域。作为一种新型的用于构建分布式、可互操作的应用程序平台, 它是基于网络、分布式、自描述、模块化的组件^[1], 它执行特定的任务且遵循一定的技术规范, 可以在线发布、调用和定位, 具有高度的灵活性、互操作性和松耦合性。Web 服务框架通常含有三个方面: 通信协议、服务描述和服务发现。它通过简单对象访问协议 (SOAP) 提供 Web 上的软件服务, 使用 Web 服务描述语言 (WSDL) 文件进行服务说明, 并通过通用的描述、发现和集成 (UDDI) 完成对 Web 服务的注册。

Web 服务相关技术的发展与应用, 使得日益增多的 Web 服务可以提供更多、更丰富的功能。SOA 应用领域含有各式各样的原子 Web 服务, 但一个简单的原子 Web 服务功能通常是有限的, 现有的服务可能无法满足用户复杂的需求。为解决此问题, 企业也越来越迫切需要集成整合的、支持业务应用的电子商务技术, 以提高自己在激烈多变的市场环境中的适应性和竞争力。为了充分利用现有的服务, 通过组合现有的 Web 服务以开发新的系统成为人们关注的焦点, 同时大量的 Web 服务组合技术研究^[2]开始在学术界涌现。具有不同单一功能的 Web 服务可以被灵活、动态地组合成能够满足各类用户复杂业务需求的增值服务。基于功能和非功能属性, 将这些单个 Web 服务集成整合为组合服务, 已成为当前 Web 服务技术研究的热点。

Web 服务组合是一个很好的可将资源重新利用的方式, 也是一种用于构建复杂软件或系统的新方法。服务组合的整个过程可以分为组合规划和优化选择两个阶段, 组合规划阶段主要是根据用户对 Web 服务的功能需求自动生成组合服务且保证组合过程的正确性, 当前关于此阶段的研究主要是基于图论方法和人工智能等方法^[3, 4]。优化选择的目的是要满足用户对组合服务的非功能需求, 通常是指服务质量(Quality of Service, QoS), 从而使最终组合出的组合服务在非功能属性上达到最优化。在这个阶段, 需要为一个规划好的服务组合流程, 从众多候选服务构成的抽象任务中选择一个具体的实际服务, 以达到组合服务全局的 QoS 最大化从而满足用户对服务质量的要求。随着具有同

样功能属性而在非功能属性上表现各异的 Web 服务的数量与日俱增,学术界中涌现的关于对 Web 服务组合问题的研究正逐渐集中在如何进行 Web 服务的优化选择这一问题上。

1.2 问题提出

目前,网络上已有巨大数量的 Web 服务提供相同或相似的功能,因此用户更加关注的通常是那些非功能属性,即服务质量(QoS)。在此背景下,基于 QoS 的 Web 服务组合问题成为了当前领域内研究的热点。基于 QoS 的服务组合模型的目的是给系统提供一个组合服务方案,即为服务组合 workflow 中的每个抽象任务(抽象服务类)具体实例化一个实体服务,使最终得到的组合服务在整体的 QoS 上尽最大力度的满足用户的需求^[5]。为达到这个目标,最为直观的办法就是将所有可能存在的组合方案结果一一列举出来并分别对它们进行 QoS 评估,此方法通过蛮力计算是可以找到 QoS 最优的组合方案的。但是在服务规模很大的情况下,遍历所有的服务组合方案往往是行不通的。首先调用服务是需要一定费用的,另外所需要的时间花费也是相当大的,两方面都会造成很大的成本。因此,当前的研究往往是采用各种最优化技术来避免这种蛮力式的穷举遍历,如通过线性规划^[6]、动态规划^[7]、遗传算法^[8]等优化算法来获取更为高效的服务组合结果。

然而,服务实际所处的网络环境是极其复杂多变的,只是通过单纯比较 Web 服务 QoS 的大小并无法彻底保证服务质量。比如,在动态、开放的网络环境中,服务的 QoS 属性会随着网络环境发生变化,此外,一些服务提供商也可能会发布不符合实际 QoS 水平的服务,这将导致服务质量的评估并无法确保其可信性。面对这种现状我们认为,在基于 QoS 的 Web 服务组合、优化选择等研究的基础上引入服务可信度计算的概念是很有必要的。当前,可信度计算在多个领域都逐渐被关注并成为相关研究的重点,SOA 领域中的针对服务可信度问题的研究也变得越来越热门。动态多变的互联网上提供着各式各样的 Web 服务,因缺乏监管多少都会存在一些不诚信的行为,被不诚信的服务商恶意提供的 Web 服务可能会给企业带来非常严重的后果,只有既满足服务的高质量又能确保其可信性,才能正确有效的选择出最能满足用户需求的服务。然而,当前对于如何衡量 Web 服务的可信度还没有一个确切、统一的说法,借鉴现有的研究本文将从新的角度合理的提出服务可信度的概念并给出相应的计算方法,在基于 QoS 的 Web 服务选择模型中加入对服务可信度的评估计算,然后基于此再根据选择出的可信任的 Web 服务进行后续组合优化的研究,这将会同时保证服务的高质量 and 可信性,促进 Web 服务优化组合的进一步发展与应用。

鉴于此,研究开放网络环境与复杂应用下的可信 Web 服务组合优化问题有着重要的理论意义和应用价值。本文的研究主要是从数学理论方面进行建模、分析、提出算法、实验验证等,对结合可信度计算的 Web 服务组合与优化问题进行相关研究。

1.3 研究现状

工业领域和学术领域已广泛关注于 Web 服务组合问题, 国、内外学者纷纷对此深入研究, 所得成果也是十分显著的。因为本文主要进行的是基于可信度与 QoS 相结合的 Web 服务组合优化问题的研究, 所以关注的文献多从以下几个方面去分析:

1.3.1 基于 QoS 的 Web 服务组合研究

QoS 作为一种非常重要的、对服务进行过滤的非功能性标准, 是判断服务是否被选择的关键因素^[9], 可以有效地提高服务发现和服务组合的满意度。基于 QoS 的 Web 服务组合, 是根据用户偏好及服务的整体 QoS 约束给组合流程中的所有抽象服务都绑定一个实际的具体服务, 从而为用户提供具有最佳 QoS 的组合服务。这个问题被认为是 Web 服务组合领域中的一个基本问题, 直到今日也仍然是一个研究的热点。

Zeng L 等人^[6]比较早的研究了此问题, 给出了对服务 QoS 的评价方法以及“服务群”、“执行方案”、“执行路径”等基本概念, 提出将服务组合问题转化成有向无环图的基本策略, 并根据线性规划的方法给出相应的优化算法, 以确保得到最佳的服务组合结果。这为以后的研究工作提供了充分的参考价值。在此基础上, Jaeger M C 等人^[10]对 QoS 的评价方法进行了扩展, 探讨了五种典型 QoS 属性在顺序、并行、选择和循环这四种流程下的聚合计算方法。在服务组合的局部优化阶段, 也就是基于 QoS 的服务选择过程中, 系统通常是通过提供者对服务的描述获得 QoS 信息, 并根据用户提供的 QoS 需求对候选服务进行排序从而选择 QoS 较优的服务^[11]。与此同时, 很多研究者提出了改善的服务选择框架^[12, 13], 对于传统的基于 UDDI 发布、发现、绑定的 Web 服务模型进行改进, 加入了 QoS 相关的模块, 如 QoS 评分模块、QoS 数据库、QoS 分级比较模块等, 以完善和提高基于 QoS 的 Web 服务选择功能。但也有研究人员更加注重 QoS 属性度量方法和 QoS 数据表示规范。Kalepu S 等人认为信誉和信任都是 QoS 属性, 且由用户使用经验决定, 关于这二者的取值上述研究者认为均是由历史用户对服务的主观评价得到的^[14]。可见大部分基于 QoS 的 Web 服务组合研究重点关注的是 QoS 属性, QoS 度量、数据更新和服务发现框架模型。本文研究的问题就是基于这个内容之上的。

1.3.2 Web 服务可信度计算研究

开放的环境是当前技术的发展趋势, 而资源的相互竞争、潜在的冲突、恶意的攻击等都是开放环境中比较常见的现象。所以计算系统不得不接受开放环境所带来的考验。近年来具有安全思想的可信计算技术受到了普遍关注。在 SOA 领域, 许多应用程序都需要 Web 服务的选择以及与其他 Web 服务间的协调, 尤其是在电子商务^[15]、语义网、点对点网络等活动中^[16]。相关的例子如 Amazon, Expdeia, eBay 等这些活动中涉及到很多有价值的资源信息, 如时间、费用、硬件使用等, 这就非常有必要在这些方面上对服务进行未来表现情况的预测和评估。由于业务流程和资源利用均取决于服务选择和组合任

务的结果，Web 服务的可信性便成为一个关键的问题^[17]。

文献^[18]提供了一个深入的有关 Web 服务社区中信任和信誉系统的调查研究综述。这篇调查研究表明了此领域的研究动机并强调了不同应用程序的可信度计算。在另一项调查研究^[19]中，作者将信任分成四大类，即基于策略的信任、基于声誉的信任，通用模型的信任，和对信息资源的信任。对于 Web 服务组合，为了能够提供满足服务请求者需求的高质量组合服务，需要考虑原子 Web 服务的可信性。文献^[20]为此设计了一个中介调解器，通过不同类型的原子服务的信任级别，推导出组合服务的可信任水平，从而提供给服务请求者最佳的执行方案。在文献^[21]中，Octavian 等人提出用主观概念计算原子 Web 服务的信任值，且给出了不同组合结构下的信任计算方法。文献^[22]中，Paradesi S 等人提出在服务选择时需同时考虑 Web 服务和服务提供商的可信性，创建了 Web 服务的信任值映射到服务提供商信任值的转换规则，且将服务属性分为客观和主观两种情况并认为二者对服务整体的信任具有不同的影响程度。文献^[14]指出仅根据用户评价得到的可信度并无法有效识别恶意 Web 服务，Kalepu S 等人认为可以将信誉作为评估服务可信度的指标，提出将用户评价和服务历史记录一同来评估服务的可信度，该方法提高了服务可信度的客观性，但其中对信誉的定义还不够清晰。文献^[23-27]将概率论统计知识用于 Web 服务的 QoS 度量，很多信任模型被提出，如文献^[23]提出服务信任值需用历史累积的经验数据进行不断更新，文献^[25]中将服务的质量水平分为好与坏两个等级，对一个给定服务的质量水平评估预测其在未来交互中表现为好质量的概率。这些模型都是通过先验与后验分布共轭的 Beta 分布这一统计知识来表示服务质量等级概率分布的。也有一些其他模型如文献^[17, 28, 29]可以表示多个服务质量等级，将二维的 Beta 分布扩展成多维的 Dirichlet 分布，从而对服务质量在未来的表现情况进行预测和评估。

本文在此研究现状下总结并提炼出新的适合的服务可信度计算方法，将信任与信誉进行重新定义并共同考虑为影响服务可信度的因素。

1.3.3 Web 服务组合多目标全局优化研究

很多关于 Web 服务选择与组合的研究通常注重的都只是局部优化^[30-33]，这些方法只是为组合中的每个局部的服务节点根据最佳的 QoS 参数进行服务选择，并不能解决最终的组合服务整体 QoS 最优化的问题。尽管一些研究^[34-36]涉及到了全局优化，但这些方法只是将 QoS 指标进行传统的加权并表示为一个单一的目标进行寻优。文献^[37]旨在加强服务组合方案的可信度问题，提出使用 Web 服务的历史 QoS 信息而不是服务提供商给出的 QoS 信息，然而其最后的服务组合优化问题却仍然转化成了整数规划模型。这些方法是存在不足的，一方面它们产生的优化结果是单一的，无法供用户挑选，当这个方案在实际中无法执行时并没有其他的方案可以替换；另一方面线性规划是一个 NP 完全问题，当服务规模增大时会对服务组合的性能造成很大的影响，且这类方法要求目标函数必须是线性的，具有很大的限制性。为避免 NP 完全的求解，文献^[38]使用混合的整数规划方法将全局约束分解为局部约束再进行分布式的寻优求解。文献^[39]提出通过 skyline 技术对候选服务的数量进行缩减从而高效的解决服务组合问题。文献^[40]提出了一个叫做

局部优化枚举法的启发式服务组合方法，旨在过滤每个服务类中的候选服务以缩减为一个数量较小的优质集合，再通过枚举所有组合方案求得最优解。然而，所有这些方法其实都无法合适的解决带有多个目标、多个约束的服务组合全局最优化问题，比如求一个使得全局服务费用最少、时间最短且在可靠性和可用性等指标上分别满足条件约束的可执行组合方案，是需要新的、合适的寻优策略和启发机制的。

Web 服务通常包含多个非功能属性评价指标，从全局角度考虑服务组合，很难找到一个在各个方面都达到最优的组合方案，而且有些属性又互相具有冲突性，如价格低与时间短往往并不同时出现在一个服务上，这些属性很难达到一致的最优，所以只能对各目标进行折中求解以尽可能地满足用户的需要。因此，将 Web 服务组合全局优化问题建立为一个多目标优化模型是很有必要的。多目标优化问题的特点就是各个目标之间的冲突性使得无法找到在所有目标上都达到最优的解，其目的是在多个目标间找到一个很好的折中，也就是说需要找到一种方案，使得对每个目标都优化到一个再优化下去就会影响其他目标的程度，这种现象被称为 Pareto 最优^[41]。Pareto 多目标优化无需再通过辅助方法对目标进行聚集或隔离，而是直接地对所有目标同时进行考虑。该方法提出对解的优劣进行“支配”判定，最终求得的解不仅包含少数在某些目标上极其优秀的解，也包含在各个目标上都不差的折衷解，求得的结果其实是一个非劣解集，可供系统或用户选择其最满意的解。Pareto 优化机制虽然比单目标优化求解要复杂，但对于问题求解的适应性很强，设计出的算法需改动的地方很少，且无需提供很繁杂的聚集函数，是真正意义上的多目标优化。基于此多目标优化机制，很多有影响力的优化算法都成功的运用到了实际应用中^[42-45]。此外，元启发式优化算法在多目标全局优化问题中起到了非常重要的作用^[46-48]，如遗传算法^[49]、模拟退火算法^[50]、粒子群算法^[51]、蚁群算法^[52]等，它们模仿自然界、生物系统中的特征，可以有效地求解 QoS 感知的服务组合全局优化问题。

在某种程度上，这些优化算法都可以成功解决 Web 服务组合优化问题。但其涉及到的参数普遍比较多，实现过程也比较复杂，且得到的最终组合结果受到算法参数选择的影响也比较大。因此本文在调研当前经典的原启发式优化算法的基础上，提出采用一种新颖的最新优化算法—改进的布谷鸟多目标优化算法来解决 Web 服务组合全局优化问题。

1.4 研究内容

基于以上对研究背景和研究现状的分析，本文旨在把可信度计算与基于 QoS 感知的 Web 服务组合进行整合，使具有可信度的 QoS 即可信 QoS 的服务作为后续参与全局组合优化的备选，并将服务全局优化组合构建为多目标优化模型进行求解。具体内容如下：

考虑在开放动态的网络环境中会存在恶意欺诈行为，一些虚假服务的存在、某些用户出于某种不正当目的对服务提供者进行不符合实际的评价等，这些都会导致系统很难得到真正优质的服务组合。此外，服务请求者从 UDDI 获取的候选服务的真实 QoS 可能与其发布出来的承诺值并不相符，也就是说某些 Web 服务没有履行其公告承诺的 QoS，这也是 Web 服务可信度低的表现。为可靠、有效地实现服务发现与服务调用，我们根据每个 Web 服务的历史行为预测和评估该服务的可信度，通过对劣质服务和可信度低的服务进行过滤，为服务

发现筛选出真正优质的 Web 服务, 从而保证后续服务优化组合的可靠性。本文根据已有对 Web 服务 QoS 的研究, 从可信性的角度来分析服务组合问题。将可信度与 QoS 相结合进行服务选择局部优化, 得到可信 QoS 的 Web 服务作为后续全局优化的备选。之后将全局组合优化构建为多目标优化模型, 并采用新颖的优化算法进行求解, 以提高服务组合优化效率且确保了组合服务的可信性。基于以上提出的研究内容, 概括如下:

(1) 针对 Web 服务的可信度(credibility), 从客观和主观两个方面, 对信誉(reputation)和信任(trust)进行清晰地定义并分别给出二者的评估计算方法, 使它们共同决定 Web 服务的可信度。

(2) 在局部优化阶段, 将可信度与 QoS 结合, 同时仍从客观和主观两方面, 给出一种 QoS 属性权重的计算方法, 提出融合了可信度的 QoS 度量模型, 并根据此度量值对候选服务进行排序筛选, 将选择出的具有可信性的 QoS 即可信 QoS 的 Web 服务作为后续全局优化组合的备选。

(3) 在全局优化阶段, 根据用户需求确定 QoS 指标及约束, 将 Web 服务全局优化组合构建为一个多目标优化模型。有了 Web 服务组合优化模型后, 本文选择采用一种当前最新的元启发式优化算法(布谷鸟算法)对模型进行求解。

(4) 实验验证本文方法的可行性及有效性。

1.5 论文组织安排

论文共分为七章, 具体安排如下:

- 第一章 绪论

本章首先介绍了本文研究所处的背景和课题的研究动机, 然后详细介绍了与本文工作高度相关的各领域的研究现状。在此基础上阐述了具体的论文研究内容。最后交代了论文的整体结构安排。

- 第二章 Web 服务及服务组合技术概述

本章首先对 Web 服务的概念和核心技术进行了总体阐述, 接着详细介绍了 Web 服务组合的有关概念及分类, 最后重点介绍了基于 QoS 的 Web 服务选择与组合优化的策略和常用算法。

- 第三章 相关理论基础

本章主要介绍了本文需要用到相关技术与理论知识, 为后文的内容阐述做好理论铺垫。

- 第四章 融合可信度的 Web 服务 QoS 度量模型及服务选择

本章提出在对 Web 服务进行 QoS 度量时将可信度考虑进去, 从客观和主观两个方面, 重新提出信誉和信任的概念, 并给出二者的评估计算模型。接着同样从主、客观两个方面给出了一种 QoS 属性权重的计算方法。最终得到融合了可信度的 QoS 度量模型, 并根据此度量值对候选服务排序从而进行服务选择。

- 第五章 基于布谷鸟算法的 Web 服务组合全局优化

本章在前述章节局部优化的基础上进行全局优化, 介绍了 Web 服务组合优化流程, 将得到的具有可信 QoS 的 Web 服务作为后续优化目标, 把 Web 服务组合全局最优化问

题建立为多目标优化模型,并提出采用改进的最新的元启发式优化算法——布谷鸟算法,对 Web 服务组合问题进行求解。

- 第六章 实验与分析

本章从实验的角度分析了所提出的 Web 服务组合优化方法的相关性能,评估和验证了本文所提出方法的正确性和有效性。

- 第七章 总结与展望

本章总结了论文的主要内容及工作贡献,并对未来的研究工作做了展望,提出一系列下一步将要研究的方向。

第二章 Web 服务及服务组合技术概述

2.1 什么是 Web 服务

Web 服务是在Internet和XML技术之上构建的分布式计算技术。从核心来看其实很简单,其实质就是利用 Internet 协议在分布节点间传送消息的 XML 文档,这种机制使得 Web 服务实现了其主要目标:使数据格式异构的软件应用、编程语言、系统平台之间的互操作成为可能^[53]。通常来说, Web 服务是由 URI 标识的软件系统,作为一种特殊服务它继承了服务的开放性、自治性、自描述性以及实现无关性,同时 Web 服务是通过 Internet 实现远程访问的。万维网联盟(World Wide Web Consortium, W3C)对此给出了一个较为详细的定义^[54]:

“Web 服务是通过统一资源定位符来进行标识的软件系统,其公共接口和绑定方法是通过 XML 进行描述和定义的,这些定义可以被其他软件系统发现与识别。这些软件系统根据 Web 服务描述的接口信息和绑定方法,在 Internet 协议栈的基础上借助基于 XML 技术的消息包来与 Web 服务进行互动。”

这个定义隐含地指出了 Web 服务的三大核心技术:

(1) 通信: 与服务器和浏览器之间通过 HTTP、HTML 协议相互通信类似, Web 服务之间也需要一种基于 Internet 协议(如 HTTP、FTP、SMTP 等)能够互相传递 XML 消息的通信协议规范;

(2) 描述: Web 服务的接口信息和绑定信息需要用一种基于 XML 的语言来描述和定义;

(3) 发布和发现: Web 服务请求者可以通过中介发现 Web 服务,且 Web 服务的元信息需要发布到中介上。

2.2 Web 服务标准

2.2.1 Web 服务体系结构

Web 服务体系结构基于三种角色之间的行为交互,这些角色和操作一起作用于 Web 服务软件模块及其描述。图 2.1 给出了 Web 服务体系架构的基本模型。

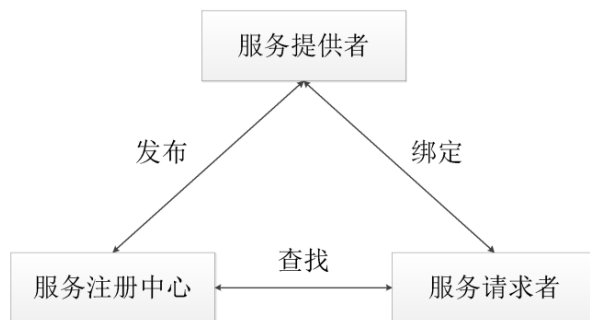


图2.1 Web 服务体系架构基本模型

该体系结构中包含三个独立的角色：

(1) 服务提供者(**Service Provider**)：指服务的所有者（企业开发商或个人），从软件体系结构的角度来说也可指提供服务的平台。

(2) 服务请求者(**Service Requester**)：指需要使用提供某些功能的 Web 服务的企业或个人，从软件体系结构的角度看也可指查找且调用 Web 服务的客户端。

(3) 服务注册中心(**Service Registry**)：指描述服务相关信息的注册机构，服务提供者在此发布他们的服务描述，服务请求者则在这里查找他们需要的服务且获取服务的绑定信息。

在面向服务的软件体系结构中，这三种角色之间存在着三种交互行为，即服务发布、服务查找和服务绑定，这些交互行为区别于传统的软件体系结构，具体含义如下：

(1) 服务发布(**Publish Service**)：为了使服务可访问，需要发布服务描述以使服务请求者可以查找它。服务可以发布到本地机构、服务注册中心等，发布服务描述的位置可以根据应用程序的要求而变化。

(2) 服务查找(**Find Service**)：在查找操作中，服务请求者直接对服务描述进行检索或者在服务注册中心中查询其所要求的服务类型。对于服务请求者，可能会在两个不同的阶段涉及到此操作：在设计时为了程序开发而检索服务的接口描述，另在运行时为了调用而检索服务的绑定和位置描述。

(3) 服务绑定(**Bind Service**)：最后对服务的调用需要执行绑定操作，在此操作中，服务请求者使用服务描述中的绑定细节来定位、联系和调用服务，从而在运行时调用或启动与服务的交互。此种绑定行为既可以于系统开发时静态的完成，也可于系统运行的过程中动态的完成。

2.2.2 Web 服务规范

上述所有行为都需要角色之间进行信息交换，因此需要用到几种核心 Web 服务规范：

1. 传输规范

Web 服务可以利用多种互联网协议来实现 XML 消息的传送，超文本传输协议(**HyperText Transfer Protocol, HTTP**)作为最广泛的 Web 通信协议成为 Web 服务技术的首选，其他传输协议还包括互联网文件传输协议(**File Transfer Protocol, FTP**)、简单邮件传输协议(**Simple Mail Transfer Protocol, SMTP**)、Java 消息服务(**Java Message Service, JMS**)协议等等，都可以供 Web 服务发送 XML 消息。

XML(eXtensible Markup Language, 可扩展标记语言)是一种用于解决数据描述与数据交换的标准技术，它是对普通 **HTML** 的扩展，用户可根据需要对其内容和结构的标记进行自定义和扩展。从而可以将文档的内容和逻辑信息反映的更为详细，具有良好的可扩展性和跨平台性，人和计算机都可以方便地进行数据类型的识别。目前 XML 已在各开源软件的模块当中得到了广泛的应用，可用于存储数据及创建配置文件等。具备这些优点的 XML 已成为了 Web 服务整个协议架构的基础。

2. 消息规范

Web 服务之间的通信是通过相互传送符合 SOAP 协议的 XML 消息实现的。SOAP 的全称是简单对象访问协议(Simple Object Access Protocol)，其主要目的是为了支持 Web 服务之间单向异步、松耦合的消息交互，为 Web 服务之间交换基于 XML 结构化信息提供可扩展的消息交互协议及消息处理框架。SOAP 消息通常包括两个部分，一部分是真正的 XML 数据负载，另一部分是额外的头信息，其中可附加各种控制信息，这些控制信息也可由一定的规范来约束从而完成特定的功能。SOAP 不针对任何一种编程语言或系统平台，无需考虑传输协议就能将序列化后的数据在 Web 上进行传输，可以将各平台、各应用之间数据交换的不兼容性降到最低。

3. 描述规范

Web 服务技术完全符合 SOA 的松耦合性，即只要通过服务描述就可以访问服务的功能。Web 服务是通过 Web 服务描述语言(Web Services Description Language, WSDL) 来进行服务描述的。WSDL 是 XML 格式的，它包括 Web 服务的两方面内容：抽象接口和抽象接口的绑定。通过抽象接口的描述定义可以知道 Web 服务提供的功能是什么；通过抽象接口的绑定描述可以知道如何对 Web 服务提供的功能进行访问；一个抽象接口可以同时具有多个绑定。具体来说，抽象接口信息包含了一系列的操作名称、输入输出参数及出错返回信息，抽象接口的绑定信息则指定了抽象接口应采用的通信协议和抽象接口的 Internet 访问地址，二者可分布在不同的描述文档中。

4. 发布和发现规范

服务注册中心是服务发布和发现的中间环节，不仅可以存储服务的元信息，而且还要提供发布和发现服务的功能接口。UDDI(Universal Discovery, Description, and Integration, 通用发现、描述和集成)是当前 Web 服务注册中心的规范，此规范为 Web 服务定义了以下元信息：

- (1) 白页信息：一般包含相关企业的地址、联系人和已知标识符(区别公司的唯一标识)；
- (2) 黄页信息：基于标准分类法的行业类别，把企业分成不同类别；
- (3) 绿页信息：包含有关业务公开的服务的的技术信息，可提供对服务的 WSDL 信息的访问。

总的来说，UDDI 定义了用于发布、发现、更新及查找其他相关 Web 服务应用程序的标准 SOAP 编程接口。它实际上是一个开放的通用信息描述框架，可以使得众多的在线企业商可以准确的发现查找到其需要的合作商，且为增加当前用户的访问以及对新用户的扩展带来便利，它能够很好的满足用户需求，从而对互联网经济的发展起到了重要作用。

2.2.3 Web 服务协议栈

当前，W3C、OASIS 都为 Web 服务技术的标准化展开了相关工作，微软、IBM、Oracle、HP、BEA 等公司也在积极得促进 Web 服务技术的发展与标准化。为正确有效地实施服务发布、服务查找、服务绑定等交互行为，就需要为每一种行为定义一个标准的协议，这些协议便构成了面向服务的软件体系结构的协议栈。图 2.2 详细呈现了 Web 服务的架构协

议栈。

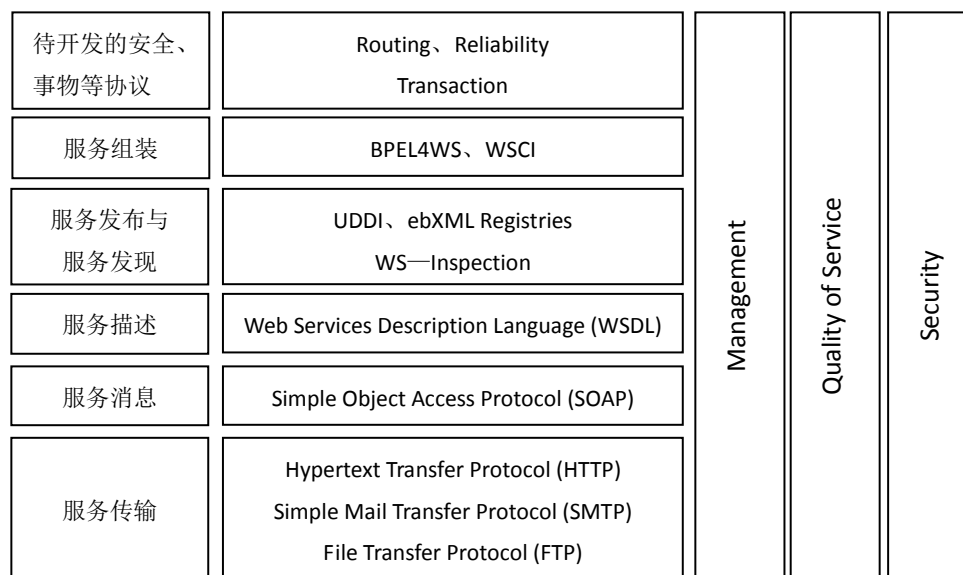


图 2.2 Web 服务架构协议栈

其中最底层的是服务传输层，从图中可以看出 Web 服务能够通过各种互联网协议传递消息，HTTP 是最广泛使用的一种，事实上已成为 Web 服务的标准协议。另外，包含 SMTP 及 FTP 在内的其他的传输协议也可以支持 Web 服务。

服务消息层中的协议主要是对消息格式的定义，该层中几乎都是以 SOAP 协议为标准的，SOAP 协议是以 XML 为基础的，Web 服务之间的交互是通过符合 SOAP 协议的 XML 消息实现的。

服务描述层的协议用于描述如何使用 Web 服务，描述信息通常为相关的数据类型、消息格式、方法名称及参数等。且 WSDL 也是以 XML 为基础的。

服务发布和发现层的协议是专为服务注册中心使用的，其中 UDDI 是当前最为广泛使用的注册中心，图中的其他方式也有应用。

服务组装层则用于将 Web 服务组装成体现特定业务流程的新服务，优点在于其中各个服务之间具有比较小的耦合性，很容易进行改造。当前在这一层中有不少互相竞争的协议，BPEL4WS 算是其中具有优势的一种。

整个 Web 服务周期都存在的协议是图中最右边的三个协议，分别为服务管理，服务质量以及服务安全。由于 Web 服务很多都是需要收费的，与金钱相关的事情就必须进行管理、追求质量和确保安全才可以。这些问题仍然是当前 Web 服务技术研究的难点，如果可以攻破这些非功能性方面的难题，Web 服务将获得更进一步的发展。

2.3 Web 服务组合

2.3.1 Web 服务组合定义

Web 服务组合是 SOA 实现业务流程和软件开发的核心技术方法。前述章节介绍的服务通信、服务描述、服务发布和查找等相关技术给面向服务的松耦合、分布式应用的开发提供了基础的平台，Web 服务组合便是在此平台上进行应用集成和开发的主要手段。

Pires 对 Web 服务组合的定义^[55]为“Web 服务组合是通过组合基本的 Web 服务以提供增值服务的能力，且这些基本的 Web 服务可由别的组织提供。”此定义提供了三点重要信息：首先，服务组合需要使用其他的服务；其次，组合后所得到的是一个具有新功能的服务；第三，服务组合所用的基本服务可以是组织外部的。此外，不同的组织机构也都有不同的说法，如 IBM 公司给出的定义^[56]为：Web 服务组合是一组支持业务流程逻辑的 Web 服务，它可以是一个新的 Web 服务，也可以是最终的应用，且组合过程是通过不同的 Web 服务的执行顺序以及它们之间复杂的交互而实现的。HPBPM 给出的定义^[57]为：Web 服务组合实际上是将现存已有的 Web 服务当作新的 Web 服务的组成模块进行重新利用，并且从总体来看各个模块都实现了增值，这种增值表现在新组成的 Web 服务不仅能够满足特定的需求，而且使服务具有更高的可用性和 QoS 保障。

虽然每个机构给出的定义都不尽相同，但大致意思都是一致的，我们可以将这些概念简要的概括为一句话，即：Web 服务组合是以特定的方式，按照给定的应用逻辑，将若干个 Web 服务组织成为一个逻辑整体的方法、过程和技术。

2.3.2 Web 服务组合分类

Web 服务的组合方法纷繁复杂。下面从四个不同的角度对当前 Web 服务组合方法进行分类，详述如下。

1. 按照使用的方法学分类

Web 服务组合方法很多都是对现有技术的继承或发展，按照不同的技术理论通常可以分为以下几类，

(1) 基于工作流的服务组合：这种方法出现的比较早，在工业界已是非常成熟的技术。该方法将服务分为两类，即抽象服务和实体服务，工作流则实际为一种建模方法，在工作流中设计人员为每一个任务都设置一个抽象服务集合，接着在执行过程开始之前通过用户需求对每个抽象服务集合进行实体服务选择和绑定调用，最后工作流中的所有抽象服务都被具体的实体服务替换，从而将抽象的工作流模型转换成实际的 Web 服务组合。具有代表性的基于工作流的方法有 BPEL4, Eflow^[58]等。

(2) 基于软件工程方法的服务组合：此类方法考虑到 Web 服务的组建是一个非常复

杂的过程，不仅需要大量比较服务是否满足匹配条件，而且搜索服务的空间也是非常巨大的，不是简简单单就能通过人工操作完成的。因此为实现自动化的服务组合，就需运用软件工程的思想，系统的分析用户需求并将需求所涉及的活动进行模型化，从而可以更好的实现和管理Web服务组合。也就是说Web服务被看作为独立的软件模块，并将软件工程中用到的思想、技术运用于真实的Web服务组合。具有代表性的基于软件工程的方法如Web构件(Web Component)组合方法、模型驱动型服务组合、程序自动生成等方法。

(3) 基于人工智能(Artificial Intelligence, AI)的服务组合：此类方法能够适应Web服务变化的动态性，可以动态链接到实体Web服务、动态生成Web服务的组合方案，从而满足不断变化的业务需求。由于传统的Web服务描述语言(WSDL)没有语义相关的描述，学术界通常用OWL-S、WSDL-S、WSML、Web PDDL等方式来增加对Web服务语义方面的描述，这些技术使基于AI规划的Web服务组合方法成为了可能。此类方法用前提、结果和IO参数等对Web服务进行描述，Web服务被看成是AI中的动作。在服务进行组合的时候，只需将这些描述映射成动作的形式，然后根据形式化推理获得Web服务的组合序列，从而动态地生成Web服务组合方案，并同时确保结果的完整性和正确性。基于AI规划的方法包括状态空间规划、层次任务网(HTN)和规划域定义语言(PDDL)等方法。

2. 按照动态程度分类

(1) 静态Web服务组合：不论是用于组合的Web服务还是组合流程都是在设计阶段就设定好的，这类方法适用于构建稳定的服务组合系统。

(2) 半动态Web服务组合：用于组合的高层模版是事先设置好的，但在实际运行的过程由系统通过给定需求和高层模版生成组合服务的步骤，然后再根据这些步骤进行服务选择从而确定参与组合的Web服务。

(3) 完全动态Web服务组合：即在运行的过程中根据用户需求生成若干符合要求的Web服务参与组合的系统。

显然，具有更高动态性的组合系统更加能够适应动态多变的互联网环境和不断变更的应用需求。动态性越高就越灵活，但是不确定因素也越多，服务组合的难度也越大。

3. 按照人工参与程度分类

根据设计人员在组合过程中的参与程度，可以分为以下几类，

(1) 完全人工组合：组合流程中的所有的步骤全部都是由设计人员计划定夺的，基于工作流的组合方法基本上都属于完全人工组合方法。

(2) 交互式组合：设计人员和组合工具交互式的完成服务组合过程。设计人员可于较高的抽象层上表达出组合的需求，然后组合工具生成部分或全部的组合流程，设计人员可以基于此进行调整，通过此种方式的不断反复，从而完成最终的服务组合。

(3) 自动组合：由设计人员于较高的抽象层次上表达服务组合的功能及非功能性需求，然后由组合工具自动地生成服务组合流程。

2.3.3 基于 QoS 感知的 Web 服务组合

QoS作为一个宽泛的概念，用来指明一个服务的非功能属性，它会影响到Web服务所提供的服务质量。目前网络上可以满足相同或相似功能的Web服务数量巨大，用户更加关注的便是服务的非功能属性（如QoS），QoS指标对于服务的过滤、选择和优化都起到了非常重要的作用，并且已成为Web服务和Web服务组合最常用的质量衡量指标^[59]。根据 W3C 标准，当前 Web服务常见的QoS属性有以下几种：

响应时间（Time）：从服务请求者发出调用服务的请求到得到响应的过程所花费的全部时间，包括服务执行的时间以及服务请求者与服务提供者之间通信往返的时间。

费用（Cost）：指服务请求者为了调用某服务需要向服务提供者支付的费用，具有相同功能的Web服务因服务质量的不同也会有不同的费用标准。

吞吐量（Throughput）：服务提供商在单位时间内处理Web服务调用请求的数量。

可用性（Availability）：表示服务可以被正常调用的概率，指一段考察时间内服务正常运行的时间比例。

可靠性（Reliability）：表示在一段时间内服务可以成功执行的次数与在该时间段内此服务总的执行次数之比，即在特定时间里服务被正确调用的概率。

声誉（Reputation）：表示基于用户的反馈评价的Web服务的可信程度。

由于QoS各属性所表示的含义及取值范围均不相同，如果没有一个统一的度量标准，就无法对多个QoS属性值进行直接的比较。为了使它们之间可以进行比较，在计算QoS之前需要对各个QoS属性值进行标准化处理，也就是将原始QoS属性值从其原来的值域线性的变换到[0,1]区间上。具体计算方法如下：

$$V' = \begin{cases} \frac{\max - v}{\max - \min}, & \max \neq \min \\ 1, & \max = \min \end{cases} \quad (2.1)$$

$$V' = \begin{cases} \frac{v - \min}{\max - \min}, & \max \neq \min \\ 1, & \max = \min \end{cases} \quad (2.2)$$

QoS属性分为两种，一种是值越小表示质量越好的（称为成本型属性），一种是值越大表示质量越好的（称为效益型属性）。如果是成本型属性，则按照公式(2.1)标准化；如果是效益型属性，则按照公式(2.2)标准化。服务的响应时间和费用按照公式(2.1)标准化，可用性和可靠性则按公式(2.2)标准化。其中v表示原始的QoS数据，v'表示归一化之后的值。

基于QoS感知的Web服务组合的目标是为系统选出一个组合方案，即给 workflow 中每一个抽象服务实例化具体的实体服务，并使得最终的组合服务的整体QoS尽最大可能满足用户的需求。因此需要对Web服务及Web服务组合进行QoS的度量计算。当前常用的Web服务QoS计算模型如下所述：

1. 原子Web服务的QoS计算模型

通过对各种类型的QoS属性的归一化预处理，并结合各属性相应的权重值，可以给出

原子Web服务的QoS值的计算公式:

$$QoS_i = \sum_{j=1}^n v_{ij} \times w_j \quad (2.3)$$

其中 QoS_i 表示第 i 个Web服务的QoS值, v_{ij} 表示第 i 个Web服务的第 j 个QoS属性标准化后的值, w_j 表示第 j 个QoS属性对应的权值, n 为第 i 个Web服务所设计的QoS属性的个数。

2. 组合Web服务的QoS计算模型

Web服务组合的QoS参数是多维的, 组合服务的QoS值主要依赖于组合流程中各原子Web服务的QoS, 且会因为Web服务之间不同的逻辑调用结构而具有不同的计算方法。常见的逻辑调用结构有顺序结构、并行结构、分支结构、和循环结构^[60]。如图2.3所示,

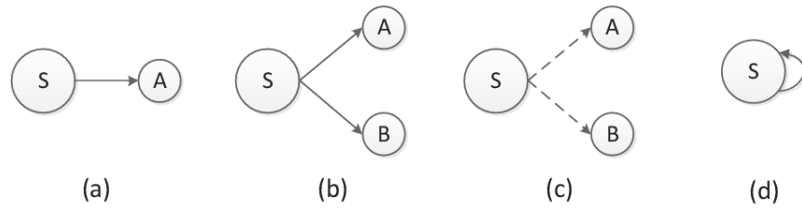


图 2.3 Web 服务基本逻辑控制结构

- (a) 顺序结构: 服务S执行完之后紧接着就执行后面的服务A;
- (b) 并行结构: 服务S执行完之后两个后续服务A和B同时执行, 且互相不依赖;
- (c) 分支结构: 服务S执行完之后服务A和B有且只有一个执行, 若执行A的概率为P则B的执行概率为1-P;
- (d) 循环结构: 服务S被重复调用执行。

设 S_i 为组合服务中的原子Web服务, 服务组合共有 m 个原子服务, P_{ai} 为选择结构的分支执行概率, 若存在 n 个分支选择结构则满足 $\sum_{i=1}^n P_{ai} = 1$, k 表示循环执行次数; 则组合服务的QoS计算模型如表2.1所示:

表2.1 组合服务QoS计算

QoS 属性	顺序	分支	并行	循环
服务费用	$\sum_{i=1}^m C(s_i)$	$\sum_{i=1}^m P_{ai} C(s_i)$	$Max\{C(s_i)\}$	$kC(s_i)$
响应时间	$\sum_{i=1}^m T(s_i)$	$\sum_{i=1}^m P_{ai} T(s_i)$	$Max\{T(s_i)\}$	$kT(s_i)$
可靠性	$\prod_{i=1}^m R(s_i)$	$\prod_{i=1}^m P_{ai} R(s_i)$	$Min\{R(s_i)\}$	$R(s_i)^k$
可用性	$\prod_{i=1}^m A(s_i)$	$\prod_{i=1}^m P_{ai} A(s_i)$	$Min\{A(s_i)\}$	$A(s_i)^k$

2.4 Web 服务选择与组合优化

2.4.1 优化策略

当前,服务选择与组合优化策略可分为三种,即局部最优策略、全局最优策略及混合策略。

1.局部最优策略

此策略的基本思想是,为服务组合中的所有抽象服务择优选择出满足局部约束条件的实体服务,即针对每组具有相同或相似功能的候选服务集,选择出能够使该抽象服务的服务质量表现最优的服务。具体来说,是对抽象服务中的各候选服务的各个参数进行加权求和得到一个服务质量的度量值,然后根据得到的度量值对每组候选服务进行排序,从而选择出既满足局部约束条件又保证服务质量最优的服务来构建组合服务。局部最优策略中各抽象服务的服务选择过程都是互相独立的,因为此策略并未涉及到对全局约束的考虑,虽然可以保证选出的服务在各单一的抽象服务中满足用户的需求,但并不能保证最后生成的组合服务是全局最优的,甚至有可能无法满足用户对全局约束的要求。局部最优策略的优点是灵活,通用;缺点则表现在没有考虑全局约束、无法保证全局最优,最终的结果只满足局部最优。

2.全局最优策略

与局部最优策略相比,全局最优策略的关注点不再只是各单一的抽象服务,而是整个服务组合过程,其目标是使得最终的组合服务在整体上满足给定的约束条件或达到预定的一些优化目标,从而在全局范围上满足用户对组合服务的质量要求。其中对各抽象服务的服务选择则需综合考虑各具体服务的聚合结果。全局最优策略考虑到了全局约束,且得到的最终解是满足全局约束条件的全局最优解,但是此类方法会增加计算的复杂性,需要高性能、高效率优化算法的支持。

3.混合策略

可以看出,两种策略都具有一定的优势同时也存在局限性。局部最优策略的计算量虽小却无法全满足用户在全局范围上的要求,全局最优策略可以得到全局最优解,但当服务规模变大时也会带来很大的计算量。混合策略则可以将局部最优策略和全局最优策略的优点融合在一起,比如可以先通过局部最优策略过滤、筛选每个抽象服务中的候选服务,然后再使用全局最优策略从没有被过滤掉的候选服务集中选择服务进行组合。利用混合策略的思想,通常可先通过局部最优策略缩小服务组合的空间,然后通过全局最优策略达到对整体的把握,最后得到的解则是既满足局部约束又满足全局约束的最优解。采用此策略需要对算法进行精心的设计,设计思路很重要。

2.4.2 优化算法

上述服务组合优化问题实际上是NP难问题,最笨的解决方法便是传统的穷举算法和贪婪算法。由于服务组合涉及的求解空间和服务规模都比较大,这种蛮力求解的算法会

有很高的复杂度，并不适合此类问题。当前比较流行的用于服务组合优化的算法主要是元启发式优化算法，如遗传算法（Genetic Algorithm, GA）、蚁群优化（Ant Colony Optimization, ACO）算法、粒子群优化（Particle Swarm Optimization, PSO）算法和一些融合算法等。上述算法具体介绍如下：

1. 穷举算法

穷举算法是将全部的服务组合方案都列举出来，并通过一一比较从而找到满足用户需求的最佳方案。此方法简单易懂，具有较强的准确性，但是计算量很大，可扩展性也比较差。如果组合流程中有 n 个抽象服务，每个抽象服务都有 m 个候选服务，那么就会存在 m^n 个组合方案，算法的时间复杂度为 $O(m^n)$ ，可见，当服务组合中的抽象服务数量或抽象服务中的候选服务数量增多时，复杂度会呈指数级增长。因此此算法仅适用于规模较小的场景。

2. 贪婪算法

贪婪算法的基本思想是以局部最优一步一步的达到最终的全局解，通过对每一个抽象服务对应的候选服务集进行贪婪的选择，找到其中的最佳服务，逐步给出解的各部分，但并没有考虑对组合服务整体的影响，通常无法保证可以得到最优解，适应场景有限。此方法比较直观，时间复杂度为 $O(mn)$ ，缺点是没有从整体角度考虑，仅能确保在单个的服务上满足用户给出的约束条件，但无法保证组合服务在整体上满足全局的约束。

3. 遗传算法

遗传算法是一种模拟自然界生物优胜劣汰进化过程的启发式搜索算法，其求解过程本质上是通过种群的不迭代寻优的过程。通过选择、交叉、突变等操作生成下一代解，逐步加强适用度函数值高的解、淘汰适用度函数值低的解，经过种群的不迭代进化，最终得到最适应环境即适应度函数值很高的解。采用遗传算法来解决基于QoS的服务组合问题，是将每一组具体的服务组合方案抽象为一个染色体，每一个染色体都由若干个基因组成，所以组合中的每一个Web服务则被抽象为一个基因。适应度函数可以确定为各个具体服务的QoS效用函数。交叉操作是对两种不同的组合方案（两个染色体）以交叉互换的方式更变某些基因位的值，从而可以产生两个新的组合方案。变异操作可以理解为染色体中的某一位基因按一定的概率进行随机变化，也就是说用一个可用服务去代替另一个对应的服务。遗传算法通用、简单、实用性强，具有全局搜索能力，但缺点是收敛性差、运行结果不稳定，需要对算法性能进行改进。

4. 蚁群优化算法

蚁群优化算法模拟蚂蚁觅食过程中进行信息交流的机制，并通过正反馈的方式得到最终的最优路径：在觅食的过程中，蚂蚁会在自己经过的路径上留下具有不同浓度的信息素，且信息素会慢慢的挥发。如果某一条路径被蚂蚁走过的越多，那么该路径上的信息素浓度就会越高，所以后来的蚂蚁就会以更大的概率去选择这条路径。蚁群算法适合解决离散的组合优化问题，因此也非常适用于组合优化中的服务选择问题，具有很好的并行性和较强的鲁棒性，能够有效地找到较优解，且易于和其他算法进行结合。但同时具有搜索时间长、易陷入局部最优等缺点。

5. 粒子群优化算法

粒子群优化算法模拟的是鸟类的觅食行为：鸟类在觅食的过程中会不断改变飞行速度和位置，各个分散的鸟儿会逐渐聚成鸟群，这个鸟群位置会高低左右不停变换，直到最终找到食物。算法执行过程中，首先初始化一组随机的粒子，然后通过迭代的方式使得每一个粒子都向自己找到的最佳位置和群体中最优的粒子靠近，从而搜索得到最优解。每一次迭代都重新计算粒子新位置的适应值，据此对个体极值和全局极值进行更新，且运动方程更新粒子的速度和位置。此算法操作简单易于实现，具有较强的记忆力，收敛速度较快，但在后期容易陷入局部最优导致搜索精度不够高。

6. 融合算法

每一种算法都具有各自的优势和不足，单一的算法往往无法很好的满足现实优化问题对高效率高性能的要求，这就需要将各算法的优势合理的融合在一起，从而提高对复杂问题进行优化的性能和效率。

2.4.3 存在的问题

当前研究领域虽然已经存在大量的基于QoS的Web服务选择与组合优化的模型及算法，但仍然或多或少的存在一些问题。比如大部分针对服务选择的研究都是给候选服务的每一个QoS属性赋一个权重值，把服务请求者要求的多个参数指标加权求和，表示为一个目标函数来衡量服务的质量，然后再采用整数规划的方法或者通过单目标优化算法优化此函数来获得全局最优解。然而，在实际应用中服务的各QoS属性的权重到底该如何确定，并没有一个合理统一的方案，现存研究多为让用户直接给出权重值，有的甚至并没有提到如何确定权重。而且由整数规划或单目标优化算法求得的解并无法实质性的解决多目标优化问题，比如将组合服务的响应时间和服务费用等进行同时优化，这其实在一定程度上限制了算法的实用性。

另外，QoS仅代表了Web服务提供商给出的承诺，而实际在服务被调用的过程中，QoS各属性性能并不一定能够达到提供商承诺的那样好，这就涉及到服务的可信性问题，传统的Web服务选择方法并未考虑到这一点，这就会导致系统最终提供的组合服务会因为其中一些不诚信的Web服务无法真正满足用户的需求，从而使得用户对服务的满意度降低。

针对以上存在的问题，本文提出一种基于可信度计算与QoS相结合的Web服务组合优化方法。首先针对Web服务QoS的可信性问题提出从客观信誉和主观信任两方面进行评估；接着给出一种QoS属性权重的计算方法，提出融合了可信度的QoS度量模型，综合求得每个Web服务的可信QoS度量值，并根据此度量值对每个抽象服务的候选服务进行过滤、筛选；最后，对各抽象服务选择出的可信Web服务，采用最新的启发式优化算法（布谷鸟算法）进行全局服务的多目标组合优化。

2.5 本章小结

本章介绍了有关Web服务及Web服务组合的基础概念，并介绍了基于QoS感知的Web服务组合优化问题的相关策略及算法，最后提出了当前存在的问题及本文研究的切入点。

第三章 相关理论基础

3.1 贝叶斯学习理论

贝叶斯方法是一种利用概率规则来进行学习与推理的过程，是统计学中非常精细的一个模型。它的基本思想是，可以根据未知参数的先验分布以及所给样本求得其后验分布，从而得到总体分布。用以下三个观点来归纳：

假设 I 随机变量 X 有一个密度函数 $p(x; \theta)$ ，其中参数 θ 取值不同就会有不同的密度函数，以贝叶斯理论来看， $p(x; \theta)$ 在给出 θ 之后就是一个条件密度函数，所以记为 $p(x | \theta)$ 更恰当，它所给出的有关 θ 的信息就是总体信息。

假设 II 在给出特定的 θ 之后，从总体的 $p(x | \theta)$ 当中随机选出一个样本 X_1, \dots, X_n ，这个样本里所包含的有关 θ 的信息就是样本信息。

假设 III 当参数 θ 已经有了足够多的历史资料，通过对这些信息的整合和总结，可以得到有关 θ 的一些有价值的信息，也就是先验信息。

参数 θ 是无法预先定义的，它不是一个固定的确定量。贝叶斯理论把未知参数 θ 看作为一个随机变量，而描述 θ 的分布的相关概念定义如下：

定义 3.1 先验分布 把总体分布中的未知参数 θ 当作为某个随机变量 Θ 的一个可能的取值，将 $\theta \in \Theta$ 的概率分布记为 $\pi(\theta)$ ，并称之为参数 θ 的先验分布。

定义 3.2 后验分布 在贝叶斯理论中，将上述所有信息综合起来的最佳方式就是在总体分布基础上获得的样本以及参数的联合密度函数 $p(x_1, \dots, x_n, \theta) = p(x_1, \dots, x_n | \theta) \pi(\theta)$ 。

在这个式子中如果已经给出了样本 X_1, \dots, X_n ，那么未知量就只是参数 θ 了，所以在给定样本信息之后，可以通过密度计算公式得到 θ 的条件密度函数

$$\begin{aligned} \pi(\theta | x_1, \dots, x_n) &= \frac{p(x_1, \dots, x_n, \theta)}{p(x_1, \dots, x_n)} \\ &= \frac{p(x_1, \dots, x_n | \theta) \pi(\theta)}{\int p(x_1, \dots, x_n | \theta) \pi(\theta) d\theta} \end{aligned} \quad (3.1)$$

贝叶斯理论把 $\pi(\theta | x_1, \dots, x_n)$ 称为参数 θ 的后验密度函数，或后验分布。而

$p(x_1, \dots, x_n, \theta) = \int p(x_1, \dots, x_n | \theta) \pi(\theta) d\theta$ 是样本 X_1, \dots, X_n 的边际分布，其中的积分部分表示的是参数 θ 的取值范围，取决于实际情况。

综上所述可将此理论总结如下：人们根据先验信息对参数 θ 有了一个提前的认知，也就是先验分布 $\pi(\theta)$ 。在给定了一些样本信息之后，通过上述的贝叶斯公式对 θ 的先验分布进行调节，调节之后得到的结果就是后验分布 $\pi(\theta | x_1, \dots, x_n)$ 。后验分布可以看作是对以上3种信息的整合。得到后验分布之后，对 θ 的认知便更加深入了，同时可以发现，加入样本信息的作用就是把人们对 θ 的认知由先验分布调节整合到后验分布。此模型将人们对于参数的先验感觉与客观的样本信息进行了一致的考虑。其优点在于建模过程中

考虑了对求未知参数的已累积知识的先验认知，难点则在于怎样选用适当的先验分布。共轭先验分布则可以很好的解决这个问题。

共轭分布是一种极大简化贝叶斯分析的方法。其作用是，在贝叶斯公式包含多种概率分布的情况下，使这些分布的未知参数在试验前被赋予的物理意义，延续到试验后，便于分析。比如在对参数的分布情形缺乏先验认知的情况下，先验分布的取法就没有依据。统计学中一种广为使用的取法，限制先验分布与后验分布在同一种类型的分布中，这就是共轭先验分布的方法^[61]。定义如下：

定义 3.3 共轭分布 如果参数的先验分布与它关于模型分布 $p(x|\theta)$ 的后验分布是同一种类型的分布，则这种类型的分布就称为模型分布 $p(x|\theta)$ 的共轭(先验)分布。

使用共轭分布可以简化计算过程，原因是先验分布与后验分布是同属于一种类型的，只是参数不一样而已。这样每当新的样本加入时($X_{n+1}=x_{n+1}$)，就将之前得到的后验分布 $\pi(\theta|x_1, \dots, x_n)$ 看成为新的先验分布，当加入新的样本得到了新的后验分布时，只需要通过先验分布（原来的后验分布）中参数的更新来完成。这样就得到了简单的后验期望的更新公式。举例说明：

1. Beta分布与二项分布

Beta 概率函数如下：

$$Beta(p|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} p^{\alpha-1} (1-p)^{\beta-1} \quad (3.2)$$

其中， $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ 是一个常系数。除去常系数不看，Beta 函数与二项分布函数具

有相同的形式，即 $p^{\alpha-1}(1-p)^{\beta-1}$ 。如果把 Beta 分布当做先验分布，那么通过贝叶斯公式计算，关于二项分布得到的后验分布与先验分布具有相同的形式。所以说 Beta 分布与二项分布是共轭的。

2. Dirichlet 分布与多项分布

Dirichlet 分布（狄利克雷分布）实际上是 Beta 分布扩充到多维得到的，而二项分布扩充到多维就是多项分布。Dirichlet 分布函数如下：

$$Dir(\vec{p}|\alpha, K) = \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^K p_k^{\alpha-1} \quad (3.3)$$

而多项分布函数为： $Mult(\vec{n}|\vec{p}, N) = \binom{N}{\vec{n}} p_K^{n(K)}$ 。其中，K 是指 K 维。 $\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}$ 和 $\binom{N}{\vec{n}}$ 都是常

系数，不看常系数，Dirichlet 函数与多项分布函数具有相同的形式，因此计算得到的后验分布也与先验分布有相同的形式。即 Dirichlet 分布和多项分布是共轭的。

3.2 粗糙集理论

1982年由波兰学者Z.Pawlak 提出的粗糙集理论(Rough Set Theory, RST)^[62]受到了国内外研究学者的广泛重视,并逐渐发展成为一种重要的数据分析理论。这种理论从一种新的角度审视知识,提出了知识是与分类有关联的,并认为知识是具有粒度的,其方法从数据的结构分布出发,以知识的粒度为主要特征对知识进行处理,进而尽可能多的挖掘知识。利用该理论可以有效分析和处理信息,从而发现知识隐含的潜在规律。采用此方法理论对决策表进行分析,可以约简得出分类规则,评价出特定属性的重要程度,因此相关学者提出了根据粗糙集理论^[63]中属性的重要性概念来确定属性的客观权重。本文将借鉴于这个思想来合理地评价各属性在 QoS 度量中所起到的作用大小,从而确定具有实际意义的 QoS 属性客观权重。需要用到的理论基本概念如下:

定义 3.4 知识表达系统 称四元组 $I = (U, A, V, f)$ 为一个知识表达系统。其中, U 表示论域,是一个非空的、有限的对象集合; A 表示的是非空的、有限的属性集合; $V = \bigcup_{a \in A} V_a$, V_a 是属性 a 的值域; $f: U \times A \rightarrow V$ 称为信息函数,它为所有对象的所有属性都提供一个对应的信息值,即对每一 $a \in A$, $u \in U$, 都有 $f(u, a) \in V$ 。知识表达系统可以简记为 $I = (U, A)$, 它可以用信息表来展示, 拿一个简单的例子来说明, 如表 3.1 所示。

表 3.1 知识表达信息表举例

论域 U	属性集合A		
	a	b	c
u_1	2	2	4
u_2	2	2	3
u_3	1	3	2
u_4	3	3	2

定义 3.5 不可辨关系 给定一个知识表达系统 $I = (U, A)$, 每一个属性子集 $P \subseteq A$ 都决定了一个二元的、不可相互区分的关系, 即不可辨关系 $IND(P)$:

$$IND(P) = \{(x, y) \in U \times U \mid \forall a \in A, f(x, a) = f(y, a)\}. \quad (3.4)$$

也就是说如果 $(x, y) \in IND(P)$, 则在属性集 P 的范围里 x 和 y 具有相同的属性值, 它们之间是不可分辨的, 这种关系称之为不可辨关系。

定义 3.6 划分 关系 $IND(P)$, 属性子集 $P \subseteq A$ 构成了 U 的一个划分, 用 $U / IND(P)$ 表示, 或者简要的记为 U / P 。 $U / IND(P)$ 中的所有元素 $[x]_P$ 称为等价类。划分 U / P 即由 $IND(P)$ 决定的所有等价类的集合。

如从表3.1中可得到划分: $U/P_{\{a,b\}} = \{\{u_1, u_2\}, \{u_3\}, \{u_4\}\}$, $U/P_{\{b,c\}} = \{\{u_1\}, \{u_2\}, \{u_3, u_4\}\}$, $U/P_{\{a\}} = \{\{u_1, u_2\}, \{u_3\}, \{u_4\}\}$ 等。

定义 3.7 知识粒度 给定一个知识表达系统 $I = (U, A)$, $P \subseteq A$, $IND(P)$ 为定义在 P 上的不可辨关系, 也可把它称为知识, 则知识 $IND(P)$ 的粒度定义为

$$GR(P) = |IND(P)| / |U|^2 = |IND(P)| / |U|^2 \quad (3.5)$$

其中 $|X|$ 表示集合 X 的元素个数。通常, $1/|U| \leq GR(P) \leq 1$: 当 U 中所有元素都具有不可辨关系, 即只有唯一的等价类时, 粒度值达到最大值 1; 当 U 中所有元素都是可以区分的, 即划分出了 $|U|$ 个等价类时, 粒度达到最小 $|U| / |U|^2 = 1/|U|$ 。

定义 3.8 知识分辨率 定义知识 $IND(P)$ 的分辨度为

$$Dis(P) = 1 - GR(P) \quad (3.6)$$

同理, $0 \leq Dis(P) \leq 1 - 1/|U|$ 。知识粒度用以表示知识的分辨能力, 粒度越小, 分辨能力就越强。 $(x, y) \in IND(P)$ 时, 说明 x, y 在 P 中不可分辨, 同属于 P 的一个等价类, 不然它们就是可以被分辨的, 属于 P 的不同的等价类。所以知识粒度表示的是, 在论域中随机选取两个元素, 这两个元素具有不可辨关系的可能性, 这个可能性越大, 粒度越大, 也就表明分辨能力越弱。反之, 则越强。

3.3 模糊层次分析法

20世纪70年代由美国运筹学T.L.Saaty教授提出的层次分析法(AHP)是一种定性与定量相结合的系统分析方法。其基本思想是将目标按照层次细分为许多不同的指标, 并通过构建判断矩阵及层次模型用于方案中各指标的权重计算。而模糊层次分析法(FAHP)则是将模糊分析法和层次分析法结合起来的一种方法。该方法引入了模糊一致矩阵, 改进了传统层次分析法中诸如判断一致性与矩阵一致性的差异、一致性检验的困难及缺乏科学性等问题, 从而使得方法与人的思维更为一致。其确定方案指标权重大小的主要步骤为以下三步:

第一步, 建立优先关系矩阵 $R = (r_{ij})_{n \times n}$

假设存在元素 a_1, a_2, \dots, a_n , 则优先关系矩阵为

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix}$$

其中, 元素 r_{ij} 表示元素 a_i 和元素 a_j 进行比较时, 元素 a_i 和元素 a_j 具有的模糊关系的隶属度可用如表 3.2 所示 0.1–0.9 的标度给出定量描述。显然, 优先关系矩阵 $R = (r_{ij})_{n \times n}$ 是一个模糊互补矩阵。

表 3.2 评分标度

标度	定义	说明
0.5	同等重要	两个元素同等重要；
0.6	略微重要	一个元素与另一个元素相比略微重要；
0.7	重要	一个元素与另一个元素相比重要；
0.8	非常重要	一个元素与另一个元素相比非常重要；
0.9	绝对重要	一个元素与另一个元素相比绝对重要；
0.1, 0.2 0.3, 0.4	反过来比较	若元素 C_i 与元素 C_j 相比较得到判断 x_{ij} ，则元素 C_j 与元素 C_i 相比较得到判断 $x_{ji} = 1 - x_{ij}$

第二步，将模糊互补矩阵转换为模糊一致矩阵

若矩阵 $R = (r_{ij})_{n \times n}$ 满足 $r_{ij} = r_{ik} - r_{jk} + 0.5$ ，则称为模糊一致矩阵。首先对优先关系矩

阵按行求和： $r_i = \sum_{k=1}^m r_{ik}$ $i=1,2,\dots,m$ ；其中 a_i 表示第 i 行的和， r_{ik} 表示优先关系矩阵 R 中

第 i 行第 k 列的元素，然后作如下数学变换： $r_{ij} = \frac{r_i - r_j}{2m} + 0.5$ ；经过行变换，将优先关系矩

阵 R 转换为模糊一致矩阵 A 。式中 r_i 、 r_j 分别表示的是第 i 行元素的和、第 j 行元素的

和， m 是指模糊一致矩阵的行数。

第三步，求解指标权重

先将模糊一致矩阵按行求和的值减去 0.5： $l_i = \sum_{j=1}^m r_{ij} - 0.5$ $i=1,2,\dots,m$ ，再对剩余不

含对角线的元素进行求和： $\sum_i l_i = \frac{m(m-1)}{2}$ （ m 为模糊一致矩阵的行数）。最后进行归

一化求解指标权重，由于 l_i 表示指标 i 相对于上层目标的重要性，所以对 l_i 归一化可得到各指标权重：

$$w_i = \frac{l_i}{\sum_i l_i} = \frac{2l_i}{m(m-1)} \quad (3.7)$$

3.4 布谷鸟搜索算法

当前，各种各样的自然元启发式算法受到关注并变得越来越受欢迎，且广泛应用于各个领域。例如，粒子群优化 (PSO) 算法是受到鱼类和鸟类群体智慧的启发，萤火虫算

法的灵感则来源于热带萤火虫的闪光模式。几乎所有的现代元启发式算法都是来源于对大自然最好特征的模仿,尤其是由数百万年自然选择演变而来的生物系统^[64, 65]。元启发式算法的两个关键特征就是强化性和多样性^[66]。强化性是为了搜索当前的最佳解决方案,多样性则是为了确保算法可以有效地探索空间。近年来此类算法发展迅速,举例如表 3.3 所示:

表 3.3 元启发式算法举例

时间	名称	来源
1975	遗传算法	模拟自然选择、进化论
1990-1995	蚁群算法	模拟蚁群觅食过程
1995	粒子群算法	鱼类和鸟类群体运动行为
2000	蚁群算法	蜜蜂采蜜过程
2005	人工萤火虫优化算法	萤火虫通过荧光信息交流
2009	布谷鸟算法	布谷鸟孵育行为

本文打算采用目前最新的元启发式算法:布谷鸟算法对服务组合优化问题进行研究。布谷鸟搜索(Cuckoo Search, CS)算法是Yang和Deb于2009年提出的一种新型智能优化算法,该算法的灵感来源于自然界中布谷鸟奇特的孵育后代的行为,其主要思想来源介绍如下:

1. 布谷鸟的孵育寄生行为

布谷鸟是一种不会筑巢且不会哺育后代的鸟类,它孵育后代的行为十分奇特,采用的是寄生方法。它会挑选一些和自己孵育时间相近的其他鸟类作为宿主,趁其离巢外出时将自己的卵产到宿主的巢穴中,而孵化出来的小布谷鸟会本能地将宿主巢中其他的卵推出去,并通过响亮的叫声获得宿主养母的哺育、提高自己的存活率。当然,在某些情况下,宿主也会发现自己巢中有陌生的卵,这种情况下宿主则会丢弃这个巢穴再另外寻找其他的地方重新进行筑巢。

2. 莱维飞行

各种研究表明,自然界中许多动物和昆虫的飞行(觅食)行为都具有典型的莱维飞行特征,即它们的飞行路径符合一个随机游走的状态,下一步的行为取决于当前的位置状态和过渡到下一个位置的概率。莱维飞行的行走步长满足一个重尾(heavy-tailed)的稳定分布,其形式表现为短距离的飞行探索与偶尔较长距离的飞行行走交替相间(如图 3.1 所示)。

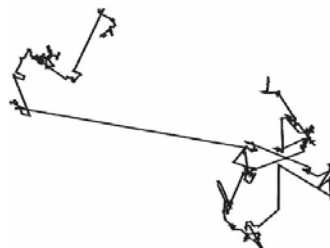


图 3.1 莱维飞行轨迹示意图

将莱维飞行运用到智能优化算法中有很大的好处:较大步长的搜索有利于扩大搜索

空间，增强种群的多样性，更容易跳出局部最优的局限；小距离步长的搜索有利于算法在小规模范围里收敛于全局最优。

3. 布谷鸟搜索

为简单起见，在描述算法之前使用以下三个理想化的规则：1) 每个布谷鸟每次只生产一个卵，并为它的卵随机选择一个宿主巢穴；2) 具有高质量卵的最佳巢穴将被留到下一代；3) 可以利用的宿主巢穴的数量是不变的，布谷鸟的卵被宿主发现的概率为 $p_a \in [0,1]$ ，这类情景下，宿主会扔掉布谷鸟的卵或者放弃此巢再建立一个新的巢。为简单起见，最后的这个假设可以近似的理解为第 n 个巢穴（方案）被新巢（新的随机解决方案）取代的概率为 p_a 。

例如对于最大化问题，一个解决方案的适应度是与目标函数的值成正比的。适应度的形式可以类似于遗传算法的适应度函数的方法。文献^[67]使用以下简单的表示：每个巢中的卵表示一个解决方案，布谷鸟的卵代表了一种新的解决方案，其目的是使用新的或潜在的更好的解决方案来代替一个不那么好的解决方案。且这个算法可以扩展为更复杂的情况，如每个巢有多个卵代表一组解决方案。算法流程如下所示：

Algorithm 1. Cuckoo Search (CS)

```

Initialise a population of n host nests  $x_i, i = 1, 2, \dots, n$ 
for all  $x_i$  do
  Calculate fitness  $F_i = f(x_i)$ 
end for
while NumberObjectiveEvaluations
  < MaxNumberEvaluations do
    Generate a cuckoo egg ( $x_j$ ) by taking a Lévy flight from random nest
     $F_j = f(x_j)$ 
    Choose a random nest  $i$ 
    if ( $F_j > F_i$ ) then
       $x_i \leftarrow x_j$ 
       $F_i \leftarrow F_j$ 
    end if
    Abandon a fraction  $p_a$  of the worst nests
    Build new nests at new locations via Lévy flights to replace nests lost
    Evaluate fitness of new nests and rank all solutions
  end while

```

产生新的方案时，以莱维飞行表现形式的布谷鸟位置更新公式为：

$$x_{t+1,i} = x_{t,i} + \alpha \oplus Levy(\lambda), \quad i=1,2,\dots,n \quad (3.8)$$

其中 $\alpha > 0$ 代表步长控制量，与具体问题的实际规模有关，大多数情况下可以使用 $\alpha = 1$ ；

$x_{t,i}$ 表示第 i 个鸟巢在第 t 代的鸟巢位置； \oplus 表示点对点乘法； $Levy(\lambda)$ 表示莱维随机搜索路径。莱维飞行实质上是随机游走的过程，其随机步长服从莱维分布：

$$Levy \sim u=t^{-\beta}, \quad 1 < \beta \leq 3 \quad (3.9)$$

布谷鸟算法具有的一些显著的优势：1) 它是一种群体智能算法，与粒子群算法和遗传算法类似，同时引入了生物学进化论；2) 莱维飞行的步长服从重尾稳定分布，因此随机搜索会更有效；3) 与其他算法相比，参数很少，便于操作，实质上只有 p_a 这一个参数。

3.5 本章小结

本章主要介绍了本文需要用到的相关技术及理论，具体用法会在后续章节中详细介绍。

第四章 融合可信度的 Web 服务 QoS 度量模型及服务选择

QoS 中的每一个属性都反映了 Web 服务在某一个方面的质量信息，每个属性对应一个属性值。由于供用户选择的候选服务数量众多，不可能做到对每一个服务都进行试用评价，因此需要对候选服务在各属性上的性能表现进行 QoS 度量从而根据度量结果做服务选择。当前研究通常采用效用函数来进行 QoS 度量，此方法最初是由 Zeng 等人^[6, 68]提出的，是服务选择与组合领域最具影响力的研究成果之一，其基本思想是在功能相同的众多候选服务中选取 QoS 度量结果最优的服务，每个服务 QoS 采用的效用函数形式为： $\sum q_i \cdot w_i$ ，其中 q_i 表示第 i 个属性标准化后的取值， w_i 表示第 i 个属性的权重。但是 Zeng 等人的研究中并没有给出属性权重的确定方法，且计算属性值的时候也没有考虑到是否值得用户信赖的问题，即服务质量的可信性问题。针对以上问题，本文提出新的 QoS 度量方法包括两个模块：QoS 可信度评估及属性权重计算。

4.1 Web 服务可信度计算模型

由前述章节可知 Web 服务有很多的 QoS 属性，其中有一个涉及到服务可信度的属性，即信誉属性，已存在的 QoS 模型是将这个属性和其他的属性视为同等的地位，并且他们所用的信誉属性表示的含义是历史用户对服务的主观评价。这种方法其实是不太合理的，首先，对于服务的可信度仅仅考虑了用户的主观评价，这样并不能识别出某些用户出于某种目的给出的恶意评价；其次，信誉值实际上是应该取决于其他属性的，而并不是与其他属性值直接进行加权求和。本文认为服务的可信度不仅需要考虑用户的主观评价，也需要对服务本身是否能够完成其所承诺的 QoS 进行客观上信誉的衡量，本文不再把信誉或信任视为与其他属性具有同等地位的指标，而是作为单独的侧面对服务的可信度进行评估。这两个指标在本文模型中的定义如下：

定义 4.1 信誉 (Reputation) Web 服务完成其承诺的好坏程度，是客观层面的指标。此部分涉及的历史数据信息由系统 UDDI 的 QoS 扩展模块根据客观反馈记录自动获取。服务具有连续性，之前的评估会为后来者提供参考价值。本章 4.2.1 小节将对此指标的评估方法进行详细介绍。

定义 4.2 信任 (Trust) 指用户对 Web 服务的使用评价，是主观层面的指标。本文从用户角度考虑，系统中的信任是单方向的，即用户对服务的信任，每个信任值是由使用过的所有用户的主观评价得到的。本章 4.2.2 小节将对此指标的评估方法进行详细介绍。

4.2.1 客观信誉评估方法

每个注册的 Web 服务都有服务提供者所承诺的质量属性值，并由服务提供者发布出来，但是具体在实际的服务调用过程中，Web 服务的 QoS 各属性的性能并不一定会

和提供商所承诺的情况相同，这就涉及到服务是否实现了承诺及完成承诺的好坏程度问题，也就是所谓的信誉问题。因此，对各服务进行信誉评估是十分必要的。

假设一个 Web 服务发布出来所承诺的 QoS 属性值为 $\{q_{p1}, q_{p2}, \dots, q_{pi}, \dots, q_{pn}\}$ ，此服务被服务请求调用后，由系统中 UDDI 的 QoS 监控模块根据其客观的执行过程反馈该服务在实际传输中的各属性值为 $\{q_{d1}, q_{d2}, \dots, q_{di}, \dots, q_{dn}\}$ ，设 $X_i \in [-1, 1]$ 用来衡量服务的实际执行完成服务提供者承诺值的好坏程度， X_i 越大表示服务完成承诺的程度越好，信誉越高。QoS 各属性指标概念不同，有的指标越大表示服务质量越好(如可靠性，可用性等)，有的指标越小表示服务质量越好(如响应时间，服务价格等)，对于前者我们更希望 $q_{di} \geq q_{pi}$ ，此时 $X_i = (q_{di} - q_{pi}) / q_{pi}$ ；对于后者，我们更希望 $q_{di} \leq q_{pi}$ ，此时 $X_i = (q_{pi} - q_{di}) / q_{pi}$ 。根据 X_i 的值我们对每一个 QoS 属性 q_i 预定义一个信誉等级：

$$C_i = \begin{cases} 1 \text{ (excellent)} & \text{if } 0.5 \leq X_i < 1 \\ 2 \text{ (good)} & \text{if } 0 < X_i < 0.5 \\ 3 \text{ (satisfactory)} & \text{if } X_i = 0 \\ 4 \text{ (bad)} & \text{if } -0.5 \leq X_i < 0 \\ 5 \text{ (very bad)} & \text{if } -1 \leq X_i < -0.5 \end{cases} \quad (4.1)$$

服务被调用后 QoS 中的每一个属性都可以通过客观执行信息的反馈得到一个信誉等级，可以看出 QoS 中的每一个属性在连续 N 次服务中履行承诺的信誉等级 X 是服从多项分布的。评估方法步骤如下：

对于 QoS 中的每一个属性，假设在信誉库中已有历史观察记录，系统中的信誉管理模块根据服务实际执行结果的反馈信息与初始发布的承诺信息进行比较，得到信誉等级反馈信息，设随机变量 X 表示 QoS 中的某一个属性在连续 N 次服务中履行承诺的信誉等级， $\bar{X}_i = \{X_{i1}, X_{i2}, \dots, X_{iK}\}$ 表示第 i 个属性的信誉信息集，K 代表信誉等级数，如我们的方法中有 5 个等级，K=5。X_{ik} 代表第 i 个属性的等级反馈值出现在第 K 个等级中的次数，我们假设 \bar{X}_i 是由带参数 $\vec{\theta}$ 的多项分布产生的：

$$p(\bar{X}_i | \vec{\theta}) = \frac{|\bar{X}_i|!}{\prod_{k=1}^{K+1} X_{ik}!} \prod_{k=1}^{K+1} \theta_k^{X_{ik}}, \quad (4.2)$$

其中 $|\bar{X}_i| = \sum_{k=1}^{K+1} X_{ik}$ ， $\theta_{K+1} = 1 - \sum_{k=1}^K \theta_k$ 。根据贝叶斯学习理论，我们知道 Dirichlet 分布可作为多项分布的共轭先验分布，带参数 $(\alpha_1, \dots, \alpha_{K+1})$ 的 Dirichlet 先验分布表示如下^[69]：

$$p(\bar{X}_i | \theta) = \frac{\Gamma(\sum_{k=1}^{K+1} \alpha_k) \Gamma(\sum_{k=1}^{K+1} X_{ik} + 1)}{\Gamma(\sum_{k=1}^{K+1} X_{ik} + \sum_{k=1}^{K+1} \alpha_k)} \prod_{k=1}^{K+1} \frac{\Gamma(X_{ik} + \alpha_k)}{\Gamma(\alpha_k) \Gamma(X_{ik} + 1)} \quad (4.3)$$

新加入执行结果（样本）后，对应参数为 $(\alpha_1 + X_{i1}, \dots, \alpha_{K+1} + X_{iK})$ 的后验分布可以给出：

$$\begin{aligned}
p(\vec{\theta} | \vec{X}_i, \alpha_1, \dots, \alpha_{K+1}) &= \frac{p(\vec{X}_i, \vec{\theta} | \alpha_1, \dots, \alpha_{K+1})}{p(\vec{X}_i | \alpha_1, \dots, \alpha_{K+1})} \\
&= \frac{\Gamma(\sum_{k=1}^{K+1} X_{ik} + \sum_{k=1}^{K+1} \alpha_k)}{\prod_{k=1}^{K+1} \alpha_k + X_{ik}} \\
&\quad \prod_{k=1}^{K+1} \theta_k^{\alpha_k + X_{ik} - 1}
\end{aligned} \tag{4.4}$$

根据已有的历史统计数据，可以计算出随机变量 $\vec{\theta}$ 的后验评估值为：

$$\hat{\theta}_k = \frac{X_{ik} + \alpha_k}{\sum_{k=1}^{K+1} \alpha_k + \sum_{k=1}^{K+1} X_{ik}}, \tag{4.5}$$

即可以理解为评估出的Qos某一属性出现在第k个信誉等级中的概率。我们把属性 q_i 的信誉值表示为其后验评估值在 $k \geq 3$ 等级中(满意, 好, 非常好)的概率和, 因此对属性 q_i 评估出的信誉值为:

$$R_{qi} = \sum_{k=3}^3 \hat{\theta}_k. \tag{4.6}$$

4.2.2 主观信任评估方法

信誉体现了 Web 服务可信度的客观层面信息，但对服务可信度的评价如果没有用户参与显然是不合理的。本文引入用户评价问题，将用户对服务的信任评价作为服务可信度的主观层面信息。相对于客观信誉来说，信任是一个直觉、主观的概念，反应的是交互双方的信赖程度。在本文的系统中信任是从用户角度单方向考虑的，即只涉及用户对服务的信任，每个服务涉及到的信任值是由以往使用过该服务的用户进行主观评价得到的。当一个服务被请求调用的时候，可以根据以往用户的历史评价记录来评估此次服务可能得到的用户信任度，历史评价是由不同用户主观评价所得到的。根据历史调用信息，Web 服务 QoS 的每一个属性都有一个用户评价价值： $T = \{t_{q1}, t_{q2}, \dots, t_{qn}\}$ ，其中每一个属性获得的信任度由以下公式计算得出：

$$T_{qi} = \frac{\sum_{k=1}^N t_{ik}(s)}{N} \tag{4.7}$$

其中， $t_{ik}(s)$ 表示第 k 次用户对服务 s 的第 i 个 QoS 属性的主观评价值，在[0,1]范围里。N 为用户与服务的交互次数。

当然，用户对服务的信任评价数据也有可能存在不可信不真实的信息，如某些用户可能会对竞争对手提供的服务进行恶意评价或对自家服务给出不符合实际情况的好评，所以可信度的评估需要综合客观和主观两个层面的信息，避免评估片面性造成后续服务选择和组合的不准确。

4.2.3 可信度计算

综上, 本文认为对于 Web 服务给出承诺的 QoS 信息不给予彻底的相信, 可信程度取决于该服务的历史行为, 包括该服务客观上完成其承诺的好坏程度以及用户对它的主观评价。在度量 Web 服务 QoS 的计算公式中, 可信度信息也应该包括在内。本文设定 QoS 属性的可信度(Credibility):

$$C_{qi} = \lambda R_{qi} + (1 - \lambda) T_{qi}, \quad 0 \leq \lambda \leq 1 \quad (4.8)$$

其中 C_{qi} 代表 QoS 中第 i 个属性的可信度, R_{qi} 为第 i 个属性的信誉评估值, T_{qi} 为第 i 个属性的信任评估值。式中 λ 取值情况如下:

- 1) $\lambda = 1$: 此时可信度完全依赖于客观信誉评估, 适用于没有用户评价的情况;
- 2) $\lambda = 0$: 此时可信度完全依赖于用户主观评价, 通常是不科学的, 因为用户的主观评价也有可能存在不真实的信息。
- 3) $0 < \lambda < 1$: 可信度取决于客观信誉与主观信任的综合评估, 可根据实际情况调整二者对可信度决定的重要程度比例, 通常将客观信誉所占比例设置的高些, 比如令 $\lambda = 0.7$ 。

对于 QoS 中的每一个属性, 我们不再使用其承诺值, 而是使用其进行可信度评估后的可信信息作为质量衡量指标, 第 i 个可信的属性值计算方法为:

$$q_{ci} = q_{pi} \times C_{qi} \quad (4.9)$$

其中, q_{ci} 表示 QoS 中第 i 个可信属性值, q_{pi} 对应服务提供者发布的第 i 个属性的承诺值 (计算时需对此属性值做归一化处理), C_{qi} 是第 i 个属性的可信度。以 4 个 QoS 属性为例, 图 4.1 简要展示了 Web 服务的可信度评估。

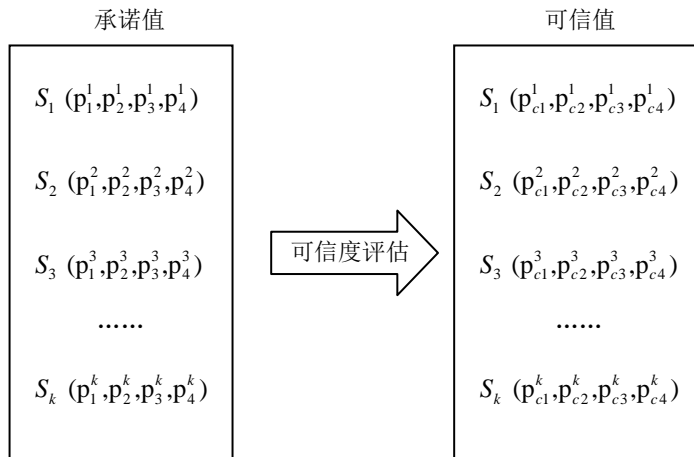


图 4.1 QoS 属性可信度评估示意图

4.2 QoS 属性权重计算模型

属性权重反应的是各QoS属性在Web服务质量中的重要程度，它的取值对Web服务的综合评价起着非常重要的作用。目前，Web服务QoS属性权重的计算方法大致分为两类：主观权重计算方法和客观权重计算方法。前者具有很强的主观性，大多侧重对用户偏好的处理并将其作为QoS度量的主要依据。后者则通过对候选服务的QoS属性进行客观的数据分析和计算，可以在不需要人为经验的干预下自动地计算出属性的权重，具有很好的客观性，但此类方法没有考虑用户的实际偏好。事实上QoS的衡量是候选服务集合与用户偏好两者共同作用的过程，若只注重其中一方面却忽略二者间关系，就可能出现主客观偏差，导致QoS度量结果不够准确。

因此，本文既考虑数据的客观分布，也考虑实际的用户偏好，给出一种从客观和主观两方面考虑的QoS属性权重计算方法：首先利用粗糙集理论通过客观权重的计算方法确定服务内在的QoS属性权重；然后利用模糊层次分析法根据用户提出的对各QoS属性的偏好来计算主观属性权重；最后综合两种方法最终得到Web服务QoS的属性权重。

4.2.1 属性客观权重计算

对Web服务进行QoS度量的目的就是为了区别各服务，找出候选服务之间存在的差异，从而为服务选择提供依据。而这种差异实际上是来源于QoS各属性指标的，所以，QoS属性对区分服务间差异的能力强弱体现了它在QoS度量中起到的作用大小。本文将这个区分服务差异的能力定义为相应QoS属性的属性分辨度，通过粗糙集理论可以给出属性分辨度的计算方法，从而进行属性客观权重的计算。为了便于理解，拿一个比较直观的例子进行说明，如表4.1所示。

表4.1 候选服务属性举例

候选 服务	属性			
	p_1	p_2	p_3	p_4
WS ₁	2	3	6	1
WS ₂	3	5	6	3
WS ₃	5	3	6	4
WS ₄	4	3	6	2
WS ₅	2	2	5	5

表中所示的候选服务集中有5个服务，每个服务对应4个QoS属性。我们可以看出，5个服务中有4个服务（即80%的服务）在 p_3 属性上的取值都是相同的，假如用户给出的属性偏好是考虑 p_2 、 p_3 作为指标进行度量，若只依据此用户偏好进行度量就会导致其度量结果几乎仅仅取决于属性 p_2 ，这样会存在很大的片面性，因此在进行QoS度量时需要考虑各属性客观的数据分布特征，以保障各属性对服务质量整体性能体现的准确性。

表4.1可以看作是一个知识表达系统，从表中可以看出，每一个QoS属性对

候选服务的区分能力是不同的，根据粗糙集理论可以得知每个属性对候选服务的划分：

$WS/p_1 = \{\{ws_1, ws_5\}, \{ws_2\}, \{ws_3\}, \{ws_4\}\}$, p_1 将候选服务划分为4个等价类；

$WS/p_2 = \{\{ws_1, ws_3, ws_4\}, \{ws_2\}, \{ws_5\}\}$, p_2 将候选服务划分为3个等价类；

$WS/p_3 = \{\{ws_1, ws_2, ws_3, ws_4\}, \{ws_5\}\}$, p_3 将候选服务划分为2个等价类；

$WS/p_4 = \{\{ws_1\}, \{ws_2\}, \{ws_3\}, \{ws_4\}, \{ws_5\}\}$, p_4 将候选服务划分为5个等价类。

为了反映各属性对于候选服务的区分能力，由粗糙集理论中知识分辨度的概念，定义每个属性的分辨度： $Dis(p_i) = 1 - |IND\{p_i\}|/|WS|^2$ 。其中 $Dis(p_i)$ 表示第 i 个QoS属性 p_i 的分辨度， $IND\{p_i\}$ 表示属性 p_i 上的不可辨关系， WS 表示 p_i 对候选服务的划分。最后，各属性对应的客观权重计算方法为：

$$w_{pi} = Dis(p_i) / \sum_{i=1}^n Dis(p_i) \quad (4.10)$$

其中 n 为属性个数。从表4.1可得：

$$Dis(p_1) = 1 - 4/16 = 3/4;$$

$$Dis(p_2) = 1 - 3/9 = 2/3;$$

$$Dis(p_3) = 1 - 2/4 = 1/2;$$

$$Dis(p_4) = 1 - 5/25 = 4/5.$$

$$\sum_{i=1}^n Dis(p_i) = 3/4 + 2/3 + 1/2 + 4/5 = 163/60$$

所以各属性的客观权重为：

$$w_1 = Dis(p_1) / \sum_{i=1}^n Dis(p_i) = 45/163 = 0.276; w_2 = Dis(p_2) / \sum_{i=1}^n Dis(p_i) = 40/163 = 0.245;$$

$$w_3 = Dis(p_3) / \sum_{i=1}^n Dis(p_i) = 30/163 = 0.184; w_4 = Dis(p_4) / \sum_{i=1}^n Dis(p_i) = 48/163 = 0.295.$$

4.2.2 属性主观权重计算

用户在进行服务选择时会对不同的属性指标有不同的偏好要求，比如有的用户比较注重服务价格这个指标，而有的用户，相比服务价格更看重服务响应时间的长短，也就是认为响应时间这个指标比服务价格这个指标更重要。这种对不同属性的不同要求是由用户的主观判断体现的。我们同样以表 4.1 为例，通过运用模糊层次分析法，对候选服务的各 QoS 属性进行主观权重的计算。

表中服务共有 4 个 QoS 属性，由用户主观判断给出两两指标之间的相对重要性，即在 0.1-0.9 模糊尺度中选择确定相对重要性，假设一个用户给出的偏好信息如表 4.2 所示，主观权重的计算步骤如下所述：

表 4.2 用户偏好信息

QoS	p ₁	p ₂	p ₃	p ₄
p ₁	0.5	0.6	0.4	0.7
p ₂	0.4	0.5	0.8	0.6
p ₃	0.6	0.2	0.5	0.4
p ₄	0.3	0.4	0.6	0.5

首先，建立两两指标比较的模糊互补优先关系矩阵：

$$R = \begin{bmatrix} 0.5 & 0.6 & 0.4 & 0.7 \\ 0.4 & 0.5 & 0.8 & 0.6 \\ 0.6 & 0.2 & 0.5 & 0.4 \\ 0.3 & 0.4 & 0.6 & 0.5 \end{bmatrix}$$

其次，将优先关系矩阵转换为模糊一致矩阵，转换规则为

$$r_i = \sum_{k=1}^m r_{ik} ; \quad a_{ij} = \frac{r_i - r_j}{2m} + 0.5 ;$$

(4.11)

$$r_1 = 0.5 + 0.6 + 0.4 + 0.7 = 2.2, \quad r_2 = 0.4 + 0.5 + 0.8 + 0.6 = 2.3,$$

$$r_3 = 0.6 + 0.2 + 0.5 + 0.4 = 1.7, \quad r_4 = 0.3 + 0.4 + 0.6 + 0.5 = 1.8;$$

$$a_{11} = 0.5, \quad a_{12} = 0.4875, \quad a_{13} = 0.5625, \quad a_{14} = 0.55;$$

$$a_{21} = 0.5125, \quad a_{22} = 0.5, \quad a_{23} = 0.575, \quad a_{24} = 0.5625;$$

$$a_{31} = 0.4375, \quad a_{32} = 0.425, \quad a_{33} = 0.5, \quad a_{34} = 0.4875;$$

$$a_{41} = 0.45, \quad a_{42} = 0.4375, \quad a_{43} = 0.5125, \quad a_{44} = 0.5.$$

得到模糊一致矩阵为

$$A = \begin{bmatrix} 0.5 & 0.4875 & 0.5625 & 0.55 \\ 0.5125 & 0.5 & 0.575 & 0.5625 \\ 0.4375 & 0.425 & 0.5 & 0.4875 \\ 0.45 & 0.4375 & 0.5125 & 0.5 \end{bmatrix}$$

下一步，对模糊一致矩阵的每一行分别求和，并减去 0.5 得到

$$a_1 = 1.6, \quad a_2 = 1.65, \quad a_3 = 1.35, \quad a_4 = 1.4;$$

接着对矩阵中去除对角线的其他元素进行求和得到 $\sum_i a_i = 6$;

最后，根据公式 $w'_i = \frac{a_i}{\sum_i a_i}$ 求得各属性的主观权重：

$$w'_1 = 0.270, \quad w'_2 = 0.275, \quad w'_3 = 0.225, \quad w'_4 = 0.230.$$

4.2.3 属性综合权重

通过分析可知，为衡量候选服务在服务质量上的表现分别，各属性的客观分布特征和用户的主观偏好都会对 QoS 的度量产生影响，因此在计算属性权重时要综合考虑这两方面的信息。本文将求得的客观权重和主观权重进行综合计算作为最终的属性权重，定义属性综合权重：

$$w_{qi} = \frac{w_i \times w'_i}{\sum_{i=1}^n w_i \times w'_i}, \quad (4.12)$$

其中 w_i 为属性客观权重，为 w'_i 属性主观权重。例子中 4 个属性的客观权重分别为 0.276, 0.245, 0.184, 0.295；主观权重分别为：0.270, 0.275, 0.225, 0.230。综合计算得出最终的属性权重为 0.297, 0.268, 0.165, 0.270。

4.3 QoS 计算与服务选择

综上所述，本文已对 QoS 各属性进行了可信度评估，且合理的确定了各属性权重，基于此，提出使用融合可信度的属性即可信属性进行 QoS 的度量计算，计算公式为：

$$QoS = \sum_{i=1}^n q_{ci} \cdot w_{qi} \quad (4.13)$$

其中 q_{ci} 是融合了可信度的属性值。这样，使用该度量方法作为服务选择的依据，可满足质量需求又保证服务的可信性，同时也考虑到了用户对各属性的偏好。该度量方法的整体框架可用图 4.2 来概括展示。

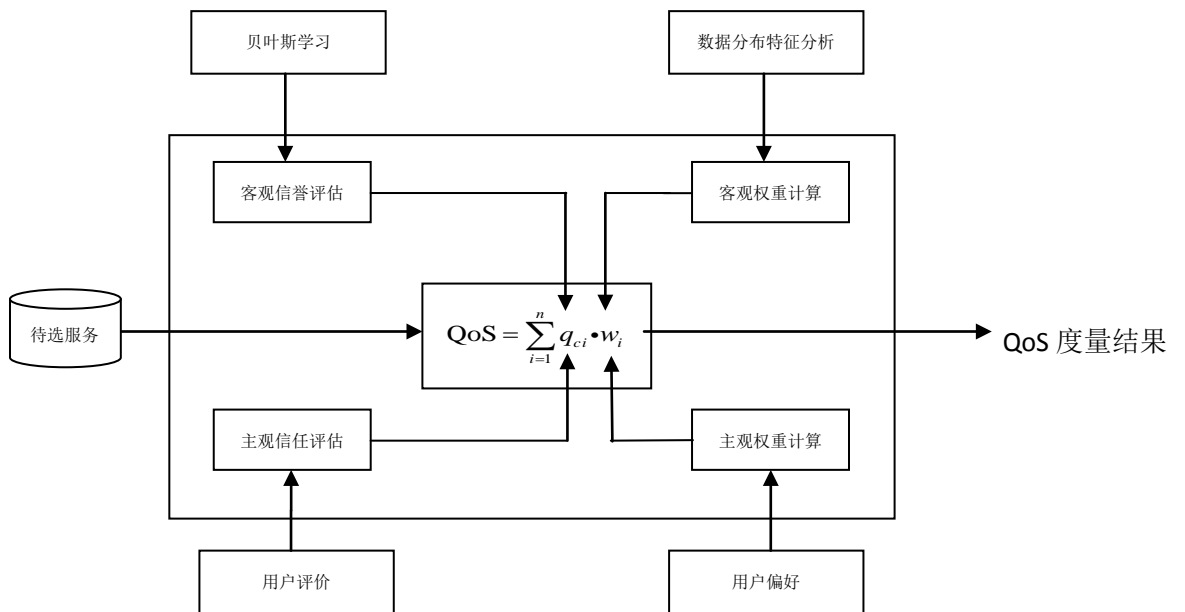


图 4.2: QoS 度量框架

最后，每一个候选服务都可以通过此 QoS 度量方法获得一个效用值，依据该效用值对候选服务进行排序及筛选，从而选择出候选服务集中效用值比较高的一些 Web 服务，作为后续全局优化组合的备选服务。此阶段的工作为后续的全局优化缩小了服务组合的规模，节省了时间和资源。

另外,针对此阶段的局部优化工作,图 4.3 简要地展示了我们在传统 SOA 标准下提出的服务选择框架,其中包含了两个新的模块:监测模块和可信度评估模块。现模拟用户针对一个抽象服务的候选服务集进行服务选择的场景,则整个过程的步骤如下所述:

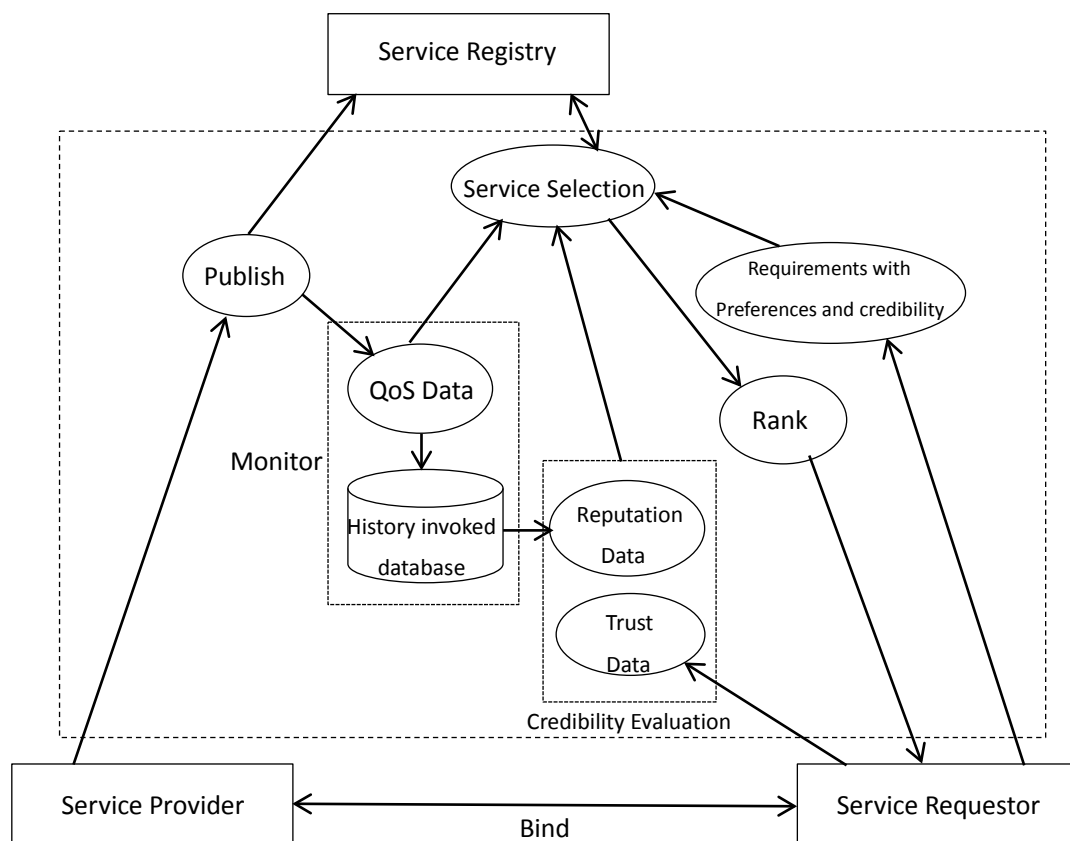


图 4.3 服务选择框架

- 1) 当服务提供者发布**Web**服务时，我们在“Qos Data”节点中收集并存储服务的 QoS 属性信息，然后将服务的功能性信息发布到“Service Registry”上。
- 2) 当服务请求者请求**Web**服务时，可以同时给出 QoS 偏好需求和可信度需求。这里的偏好是指服务请求者对 QoS 属性重要性（权重）的决策和对 QoS 属性的区间约束。
- 3) 进行服务选择时，首先在“Service Registry”上根据服务的功能属性找到满足条件的候选服务，然后再根据“Qos Data”节点中存储的 QoS 属性信息以及可信度评估模块评估出的可信度信息，对候选服务进行效用值度量、排序，将排名较为靠前的 **Web** 服务选择出来提供给服务请求者。

- 4) 当服务每完成一次调用，都由监测模块将服务在实际调用中的 QoS 信息存储在历史调用信息库中，并将由此评估出的信誉信息存储在“Reputation Data”中；同样，服务请求者在调用完该服务之后，将其对该服务各 QoS 属性的信任评价信息存放在“Trust Data”中，以供下一次服务选择时可信度评估使用。

由上述信息可知，图中虚线框中的部分可以看作是服务请求者与服务提供者之间的中介。

4.4 本章小结

本章详细介绍了如何对 Web 服务 QoS 各属性进行可信度的评估，从客观和主观两个方面，全面的分析 Web 服务的可信性问题，重新提出信誉和信任的概念，给出二者的评估计算模型。接着同样从客观和主观两个方面，给出了 QoS 属性权重的计算方法。最终得到融合了可信度的 QoS 度量公式，并根据此度量值对候选服务排序从而进行服务选择。

第五章 基于布谷鸟算法的 Web 服务组合全局优化

5.1 Web 服务组合优化流程

在服务组合领域, workflow 模型是一种广泛使用的业务流程模型, 是包括抽象任务(本文指抽象服务)节点并定义潜在数据依赖关系的有向无环图。如图 5.1 就是一个 workflow, 每个抽象服务都有一个候选服务集。涉及到的几个基本概念介绍如下:

定义 5.1 抽象服务 (Abstract Service) AS_i 一个抽象服务是抽象流程模型中描述相应任务所需功能的一个节点。

定义 5.2 具体服务 (Concrete Service) $CS_{(i,j)}$ 一个具体服务 $CS_{(i,j)}$ 是带有 n 个 QoS 属性参数的、完成了指定抽象服务 AS_i 功能的真实的服务。

定义 5.3 候选服务集 (Service Candidate Set) C_i 一个候选服务集 C_i 是一个具有相同功能但不同非功能属性 (QoS) 的所有具体服务的集合。

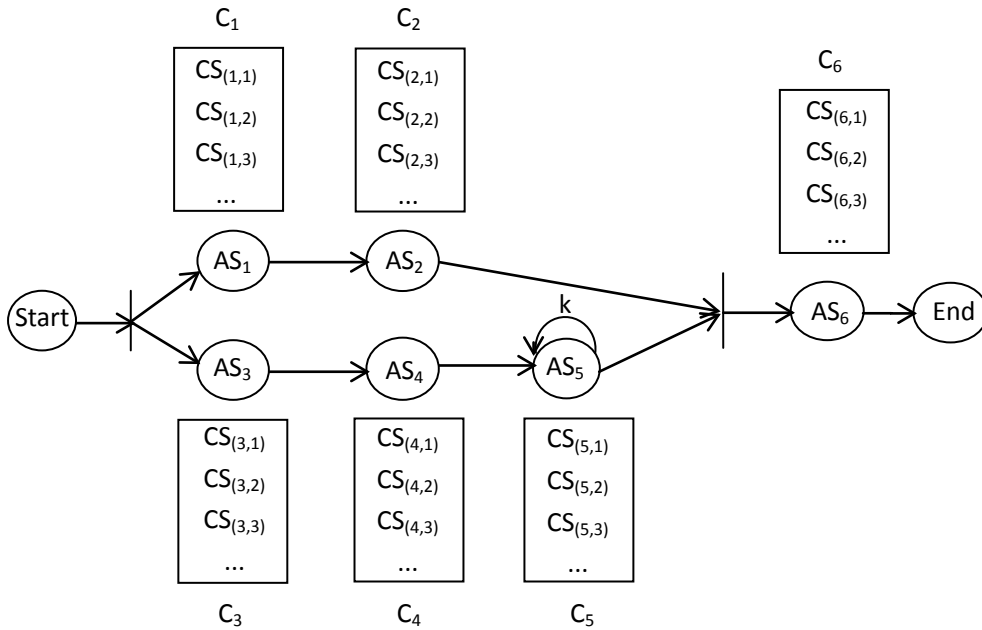


图 5.1 服务组合 workflow 示意图

根据一个 workflow, 适当的服务被选择出来, 形成一个实际的服务组合方案。在基于 QoS 的 Web 服务组合中, 如何从相应的候选服务集中选择具体服务直接决定了组合服务的 QoS 水平。 AS_i 代表抽象服务 i , $CS_{(i,j)}$ 代表抽象服务 AS_i 相应的候选服务集 C_i 中的第 j 个具体服务。根据图 5.1, 将抽象服务用其对应的候选服务集进行替换, 则构造出一个完整的具有两两候选服务集相连接的服务组合, 如图 5.2 所示。

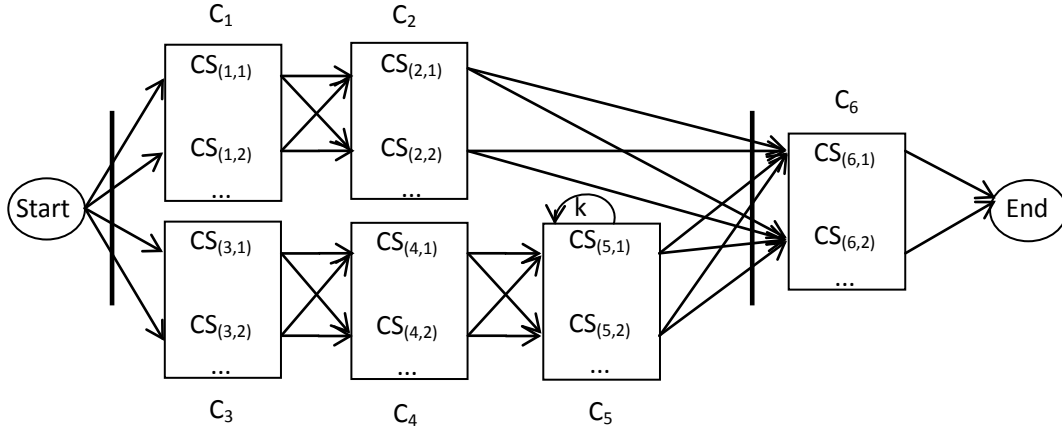


图 5.2 候选服务连接图

此示意图展示了所有可能的服务组合方案，每一个组合实际上都是一个遍历了包含每一个候选服务集在内的服务执行路径。图 5.1 和图 5.2 只是一个简单的服务组合例子，图中包含的节点和边都比较少，这样通过穷举法就可以比较容易的找到最优的组合方案，然而在实际的服务组合过程中情况往往复杂的多，涉及的服务数量很多，流程图的节点和边都会相应增多，采用穷举法寻优就变得不太可能，这就需要借用合适的优化算法进行优化求解。

上一章解决的是如何在候选服务集中根据可信 QoS 效用值选择优质服务的问题，这是单纯的服务选择问题，也可以叫做局部优化问题。整个流程是对多个抽象服务进行筛选而得到各种组合方案的，而我们的最终目标是为了得到具有全局 QoS 最优的组合服务，所以对服务选择过程之后的这些组合方案还需要继续选择。也就是说在进行了局部优化之后，还需要进行全局优化。

5.2 多目标优化模型

基于 QoS 的 Web 服务组合全局最优化是一个典型的多目标优化问题，组合服务的多个 QoS 属性指标往往无法做到同时最优，如响应时间短和服务费用低通常不会一并得到满足。多目标优化是一种特殊的优化问题，它是在满足某些约束的情况下同时优化两个或两个以上的目标，且这些目标之间可能存在冲突，因此不能找到一个使全部的目标都满足最优情况的一个方案，只能对各个目标进行折中，使得最终得到的解在各个目标上尽可能的满足用户的需求。所以它比单目标优化情况更复杂，需要更为合适的评优策略和启发机制。首先我们先介绍几个相关定义和概念：

定义 5.4 多目标优化 通常，多目标优化的目的是找到一个向量 $\vec{x} = [x_1, x_2, x_3, \dots, x_n]$ ，使得相应的向量函数 $f(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})]$ 达到最优。其中 $x_i (i=1, 2, \dots, n)$ 称为决策变量（参数）， $f_i(\vec{x}) (i=1, 2, \dots, m)$ 表示相应的 m 个目标的目标函数^[41]。

定义 5.5 理想向量 假设向量 $\vec{x}^i = [\vec{x}_1^i, \vec{x}_2^i, \dots, \vec{x}_n^i]$ 使得第 i 个目标函数 $f_i(\vec{x})$ 达到最优（最大化或最小化），得到向量 $\vec{f}^0 = [f_1^0, f_2^0, \dots, f_m^0]$ ，称这个向量为理想向量。其中 f_i^0 表

示第 i 个目标函数的最优值，这在多目标优化模型中是最理想的情况^[41]。

理想向量中包含了每个目标单独考虑时的最优值，这个向量并不一定是真实存在的，可能只是一个假想的理想最优，但却可以作为各种优化算法所尽可能去达到的一个目标。也就是说，实际执行方案对应的向量与理想向量之间的差距越小就表示该方案越好。一旦从一个组合方案中得到一个向量，就可以用一个全局规则的测量方法来测量该向量与理想向量 \vec{f}^0 之间的相近程度。本文参考文献^[41]中一种权威且广泛应用的衡量标准： L_p 测量方法，来测量所得方案与理想向量间的差异，公式如下：

$$L_p(f(\vec{x})) = \left[\sum_{i=1}^k \left| \frac{f_i^0 - f_i(\vec{x})}{f_i^0} \right|^p + \sum_{i=k+1}^m \left| \frac{f_i(\vec{x}) - f_i^0}{f_i(\vec{x})} \right|^p \right]^{\frac{1}{p}} \quad 1 \leq p \leq \infty \quad (5.1)$$

其中 p 通常可以取 2, $f_i(\vec{x})$ ($1 \leq i \leq k$) 表示需要最小化的目标函数的值, $f_i(\vec{x})$ ($k+1 \leq i \leq m$) 表示需要最大化的目标函数的值。 L_p 的值越小，表示执行方案与理想向量越接近，方案越优。

下面举个简单的例子：三个抽象服务的候选服务集中各有 3 个具体服务，每个服务包含 4 个 QoS 属性 (response time, cost, availability, reliability)，具体取值如表 5.1 所示。

表 5.1 服务组合示例

AS ₁	AS ₂	AS ₃
CS _(1,1) : {3,1,0.7,0.8}	CS _(2,1) : {8,3,0.8,0.9}	CS _(3,1) : {4,2,0.7,0.6}
CS _(1,2) : {2,3,0.6,0.7}	CS _(2,2) : {6,4,0.6,0.7}	CS _(3,2) : {3,5,0.5,0.8}
CS _(1,3) : {4,2,0.5,0.9}	CS _(2,3) : {5,7,0.7,0.5}	CS _(3,3) : {5,6,0.8,0.9}

多目标优化模型为：全局最小化响应时间和费用，全局最大化可用性和可靠性，即

$$\begin{cases} \min q_{rt}(S) = \sum_{i=1}^j q_{rt}(CS_{(i,j)}) \\ \min q_c(S) = \sum_{i=1}^j q_c(CS_{(i,j)}) \\ \max q_a(S) = \prod_{i=1}^j q_a(CS_{(i,j)}) \\ \max q_r(S) = \prod_{i=1}^j q_r(CS_{(i,j)}) \end{cases} \quad (5.2)$$

其中 $q_{rt}(S)$ 、 $q_c(S)$ 、 $q_a(S)$ 、 $q_r(S)$ 对应表 2.1 中组合服务的 QoS 计算公式。

那么理想的最优情况是这样的：

$$f_{rt}^0 = \min\{3,2,4\} + \min\{8,6,5\} + \min\{4,3,5\} = 2 + 5 + 3 = 10;$$

$$f_c^0 = \min\{1,3,2\} + \min\{3,4,7\} + \min\{2,5,6\} = 1 + 3 + 2 = 6;$$

$$f_a^0 = \max\{0.7,0.6,0.5\} \times \max\{0.8,0.6,0.7\} \times \max\{0.7,0.5,0.8\} = 0.7 \times 0.8 \times 0.8 = 0.448;$$

$$f_r^0 = \max\{0.8, 0.7, 0.9\} \times \max\{0.9, 0.7, 0.5\} \times \max\{0.6, 0.8, 0.9\} = 0.9 \times 0.9 \times 0.9 = 0.729.$$

得到理想向量 $\vec{f}^0 = [10, 6, 0.448, 0.729]$ 。

然而，在真实情况下，理想向量中对应的值不一定会同时出现在一个实际的执行方案中。假设实际执行的组合方案为 $CS_{(1,1)}$, $CS_{(2,2)}$, $CS_{(3,3)}$ ，则该方案对应的属性向量为

$([3, 1, 0.7, 0.8], [6, 4, 0.6, 0.7], [5, 6, 0.8, 0.9])$ ，可以得到：

$$f_n(\vec{x}) = 3 + 6 + 5 = 14; \quad f_c(\vec{x}) = 1 + 4 + 6 = 11;$$

$$f_a(\vec{x}) = 0.7 \times 0.6 \times 0.8 = 0.336; \quad f_r(\vec{x}) = 0.8 \times 0.7 \times 0.9 = 0.504;$$

因此实际执行方案对应的向量为 $f(\vec{x}) = [14, 11, 0.336]$ ，与理想向量差距测量为：

$$\begin{aligned} L_p(f(\vec{x})) &= \left[\sum_{i=1}^k \left| \frac{f_i^0 - f_i(\vec{x})}{f_i^0} \right|^2 + \sum_{i=k+1}^m \left| \frac{f_i(\vec{x}) - f_i^0}{f_i(\vec{x})} \right|^2 \right]^{\frac{1}{2}} \\ &= \left[\left| \frac{f_n^0 - f_n(\vec{x})}{f_n^0} \right|^2 + \left| \frac{f_c^0 - f_c(\vec{x})}{f_c^0} \right|^2 + \left| \frac{f_a(\vec{x}) - f_a^0}{f_a^0} \right|^2 + \left| \frac{f_r(\vec{x}) - f_r^0}{f_r^0} \right|^2 \right]^{\frac{1}{2}} \\ &= 0.9579 \end{aligned} \quad (5.3)$$

如上所述，全局优化组合的难点在于在服务组合的过程中需要同时最优化4个目标，而目标之间又存在交叉或冲突，因此实际上可能并不存在满足所有目标都最优的服务组合。然而在多目标优化方法中是允许多个相互冲突的目标共同存在的，即可以降低某些目标的性能为代价来改善另一些目标。如此，最优解的概念就转换成了“非支配解集”的概念。也就是说，多目标优化问题虽不能得到各目标同时最优的解，但是可以得到 Pareto 最优解或称为非支配解，即在一个目标或几个目标上无法进一步优化，但对于其他目标也不至于劣化。相关定义介绍如下：

定义 5.6 Pareto 支配 假设两个解向量 \vec{x}_1 和 \vec{x}_2 ，如果满足：1) 在所有的各个目标上， \vec{x}_1 都不劣于 \vec{x}_2 ；2) 至少存在一个分目标，使得 \vec{x}_1 在这个分目标上严格优于 \vec{x}_2 ；则称 \vec{x}_1 Pareto 支配 \vec{x}_2 。

定义 5.7 Pareto 最优解 如果在解空间中不存在 Pareto 支配 \vec{x} 的解，则称 \vec{x} 是非支配解，也称为 Pareto 最优解。

定义 5.8 Pareto 最优解集 所有 Pareto 最优解构成的集合称为多目标优化问题的 Pareto 最优解集。

我们利用上述 Pareto 语义来解决本文提出的问题，对于给定的 Web 服务组合集合，各组合方案间的优劣关系可以根据以上语义通过两两比较得到。比如假设存在两个服务

组合 S_1 和 S_2 ，如果它们满足定义 5.6 中的两个条件，则称服务组合 S_1 Pareto 支配服务组合 S_2 。

5.3 改进的多目标布谷鸟搜索算法

本节的目的在于为前文建立的多目标优化模型提出高效的解决方案。近年来，启发式多目标进化算法被普遍认为是针对多目标优化问题的最强有力的解决方法^[41]，遗传算法、蚁群算法、粒子群算法、模拟退火算法等等已被成功应用在该类问题的解决中，这些算法在执行过程中无需额外信息，不管目标函数与其限制特性如何，都能获得较为高效的结果，而且这些算法具有很好的鲁棒性和令人满意的执行效率^[70]。本文采用近年来最新的启发式优化算法——布谷鸟算法作为本文模型的解决方案，目的是为了获得第 5.2 小节所建立的多目标优化模型中满足 4 个 QoS 属性优化目标的 Web 服务组合。

在 3.4 节中已经介绍了布谷鸟算法的基本概念及流程，但其涉及的只是单目标优化，针对前述章节建立的多目标优化模型，还需要做进一步的改进。在原始的单目标布谷鸟优化算法中，每个布谷鸟每次只生产一个卵，并为它的卵随机选择一个宿主巢穴，而对于多个目标优化的情况，可将此规则改为：每个布谷鸟一次生产 k 个卵并为其随机选择宿主巢穴，第 k 个卵就代表方案中的第 k 个目标。2011 年文献^[71]提出了多目标布谷鸟搜索 (Multi-Objective Cuckoo Search, MOCS) 算法，且在近几年受到了广泛的关注，该算法流程如 Algorithm 2 所示。

Algorithm 2. Multi-Objective Cuckoo Search (MOCS)

Initialize an initial population of n nests x_i ($i=1,2,\dots,n$) and each with K eggs

for all x_i **do**

Calculate objective functions $f_1(x), \dots, f_K(x)$ $x = (x_1, \dots, x_d)^T$

end for

while NumberObjectiveEvaluations

< MaxNumberEvaluations **do**

Generate $nest'_j$ ($j=1,2,\dots,n$) based on $nest$ by Lévy flight

for all x_j **do**

Evaluate the objective values of each objective functions

end for

Choose $nest'_i$ (say i) in $nest'$ randomly and choose $nest_j$ in $nest$

if $nest'_i$ dominate $nest_j$ **then**

$nest_j \leftarrow nest'_i$

end if

Abandon a fraction p_a of worse nests

Keep the best solutions (or nests with non-dominated sets)

Sort and find the current Pareto solutions
end while

本文即是于此基础上，在莱维飞行位置更新时提出采用可动态调整的步长，并结合非支配排序思想在种群更新方面做了改进，从而提出一种适合本文模型求解的改进的多目标布谷鸟搜索 (Improved Multi-Objective Cuckoo Search, IMOCS) 算法。具体改进之处详述如下：

1. 莱维飞行步长动态更新

布谷鸟算法采用莱维飞行的形式探索空间，当产生新解 x_{t+1} 时位置更新公式为：
 $x_{t+1,i} = x_{t,i} + \alpha \oplus \text{Levy}(\lambda)$, $i=1,2,\dots,n$ 。其中 $\text{Levy}(\lambda)$ 服从莱维概率分布，用下列公式产生莱维随机数：

$$\text{Levy}(\lambda) = \frac{u}{|v|^{\frac{1}{\beta}}}, \quad \lambda = 1 + \beta, \beta \in (0, 2] \quad (5.4)$$

编程过程中通常令 $\beta=1.5$ 。 u, v 服从正态分布： $u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2)$ ，标准差 σ_u, σ_v 满足：

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \times \sin(\pi \times \frac{\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}}, \quad \sigma_v = 1, \text{ 其中 } \Gamma(a) = \int_0^\infty e^{-t} t^{a-1} dt. \quad (5.5)$$

而步长控制量 α 则是一个很重要的参数，它控制着算法局部与全局的寻优探索能力，当步长太小时，会降低全局搜索速度，当步长比太大时，又不利于局部搜索精度，因此对步长进行动态调整是非常有必要的。原始的布谷鸟算法及 MOCS 算法^[71]中的 α 均采取的是常量的形式，本文则采用随着种群的迭代进行动态更新的步长控制量。通过分析可以知道，在迭代初期 α 应足够大以增加解向量的多样性，有利于全局搜索，而随着迭代的进行在迭代后期 α 应相应降低从而有利于局部搜索，提高搜索精度。参考文献^[72]的方法，改进的动态自适应步长控制量公式为：

$$\alpha = A / \sqrt{G} \quad (5.6)$$

其中 A 是莱维飞行步长初始值 1， G 是当前的迭代数。

2. 非支配等级排序

在算法寻优的过程中，有些目标函数的值是非支配的，有些则是受支配的，也就是说存在这样的解在每个目标上都优于它。而受支配的强弱程度也的存在差异的，也就是说能够支配它的解有多有少，那么怎样区分不同支配程度的解，以及如何保证非支配解的进化，对多目标优化问题十分重要。1994 年 Srinivas 等人于文献^[73]中提出一种对种群中的个体进行非支配排序与小生境技术相结合的非支配遗传算法 (NSGA)，后于 2002 年被 Deb 等人改进为 NSGA-II 算法^[49]，本文将其非支配排序思想融入到布谷鸟优化算

法中：将目标函数值根据其受支配的程度进行排序，对目标空间进行等级划分。首先将当前空间中的所有非支配解标记为级别 1，然后把它们移出，在剩下的目标值里继续寻找非支配解并将它们标记为级别 2，以此类推，直到空间中所有解都进行了非支配排序为止。序号越小的解，表示支配它的个体数越少，非支配能力越强，所以应优先得到进化。

3. 种群更新与适应度函数

在文献^[71]的 MOCS 算法中，原始种群通过莱维飞行位置更新后得到新的种群，其更新迭代机制为：在新的种群中随机选择某一个个体与原始种群中的某一个个体进行非支配关系的比较，然后根据其比较结果进行个体替换。而本文借鉴 NSGA-II 算法思想^[49]，在种群更新迭代时，将原始种群的 n 个体与新生种群的 n 个体结合形成一个组合，然后对组合中的 $2n$ 个个体进行非支配关系排序，从组合中选出 n 个个体进行下一轮的迭代进化。选择规则为：优先选择非支配排序中级别序号小的个体，对于级别相同的则按照个体适应度函数进行选择。本文选用 5.2 小节中提到的解向量与理想向量间的差距函数作为等级相同的个体适应度函数：

$$L_p(f(\vec{x})) = \left[\sum_{i=1}^k \left| \frac{f_i^0 - f_i(\vec{x})}{f_i^0} \right|^2 + \sum_{i=k+1}^m \left| \frac{f_i(\vec{x}) - f_i^0}{f_i(\vec{x})} \right|^2 \right]^{\frac{1}{2}} \quad (5.7)$$

由 5.2 小节可知该适应度函数值越小的个体越好。

以上详细介绍了本文提出的改进的多目标布谷鸟搜索算法的重要改进环节，具体的算法步骤描述如下：

Algorithm 3. Improved Multi-Objective Cuckoo Search (IMOCS)

Initialize an initial population of n nests x_i ($i=1,2,\dots,n$) and each with K eggs

for all x_i **do**

Calculate objective functions $f_1(x), \dots, f_K(x)$ $x = (x_1, \dots, x_d)^T$

end for

Generation number $G \leftarrow 1$

while NumberObjectiveEvaluations

< MaxNumberEvaluations **do**

Generate nest' x_j ($j=1,2,\dots,n$) based on nest by Lévy flight

for all x_j **do**

Evaluate the objective values of each objective functions

end for

Combine nest with nest' to form $2n$ tempnest

for all individuals in tempnest **do**

non-dominated sorting

end for

Choose best n individuals as the next population based on the level and fitness function

Abandon a fraction p_a of worse nests

Keep the best solutions (or nests with non-dominated sets)

Sort and find the current Pareto solutions

$\alpha \leftarrow A/\sqrt{G}$

$G \leftarrow G+1$

end while

将此改进后的布谷鸟多目标搜索算法用于对前述章节建立的模型求解，算法中的鸟巢对应模型中的 Web 服务组合方案，各个鸟巢中的卵对应服务组合方案中的全局 QoS 优化目标。下一章将涉及对算法的实现以及对实验结果进行的对比分析。

5.4 本章小结

本章首先介绍了 Web 服务组合优化流程，交代了本章进行全局优化的工作流模型；接着将基于 QoS 的 Web 服务全局最优化问题构建为多目标优化模型；并对原始布谷鸟算法进行改进，提出改进的布谷鸟多目标优化算法对问题进行求解。

第六章 实验与分析

本章就本文提出的基于可信度计算与 QoS 相结合的服务组合优化方法进行一系列的实验验证。实验分为三大部分：第一部分是本文提出的可信度评估方法进行有效性验证。第二部分是针对局部优化阶段，将基于 QoS 传统的服务选择方法、依据用户评价的服务选择方法和本文提出的服务选择方法这三种方法进行查准率(precision)对比。第三部分是针对全局组合阶段多目标优化问题中，本文采用的改进的布谷鸟搜索算法与其他几种经典算法之间的性能对比。

6.1 实验环境与数据集

本文所有实验均用 Matlab 7.1 编写实现，运行环境为 Intel i3-2120 3.30GHz CPU，4GB RAM,64位，Windows 7操作系统。数据集方面既采用了真实数据集也采用了模拟数据集。其中可信度评估实验因缺乏真实企业服务的实际历史调用信息，采用的是模拟数据。其他实验中真实数据集采用的是服务计算领域经典的数据集 QWS，和较为广泛使用的 MSR-Bing 数据集。具体地说，QWS 数据集包含 2507 个真实的 Web 服务记录，这些服务都是可以公开获得的，大多来源于服务网站、UDDI 注册中心和搜索引擎等，每个服务都有 9 个 QoS 属性，本文选取其中 4 个常用的属性用于多目标优化模型中。MSR-Bing 数据集来源于 Microsoft Research Bing Image Retrieval Challenge project2，一共 7,174,869 条记录，本文随机选取其中 5000 条记录用于实验，将每条记录模拟为用户提供图片服务的 QoS 信息，仍选取其中 4 个属性。另外，我们还随机生成了一组模拟数据，称为 Synthetic 数据集，该数据集随机生成 5000 个 Web 服务记录，每个服务包含 4 个属性用于实验，属性规格如表 6.1 所示。

表 6.1 Synthetic 数据集属性设置

属性	取值范围
价格 (\$)	(0, 2000)
响应时间 (ms)	(0, 400)
可用性 (%)	(0, 100)
可靠性 (%)	(0, 100)

6.2 可信度评估实验验证

第四章提出，对于 Web 服务给出承诺的 QoS 信息不给予彻底的相信，其可信程度取决于该服务的历史行为，包括该服务客观上完成其承诺的好坏程度以及用户对它的主观评价。由于在现实生活中，不同用户对服务的评价不尽相同，也会存在有些用户出于

某种不正当的目的对服务进行不真实的评价，从而影响对服务可信度的判断。当前存在的涉及到 Web 服务可信性的研究中几乎都只是把用户评价作为对服务可信度评估的指标，本文认为这会造成很大的评估片面性，因此提出了引入对服务进行客观信誉评估的方法，从而降低不真实的用户评价对可信度评估造成的不良影响。下面就针对本文提出的客观信誉评估方法进行实验验证。

由于现实中无法获得企业真实的服务调用数据，本文采用几组随机生成的模拟数据来模拟服务各 QoS 属性在实际调用交互中的数值，具体的数据分布如图 6.1 所示。

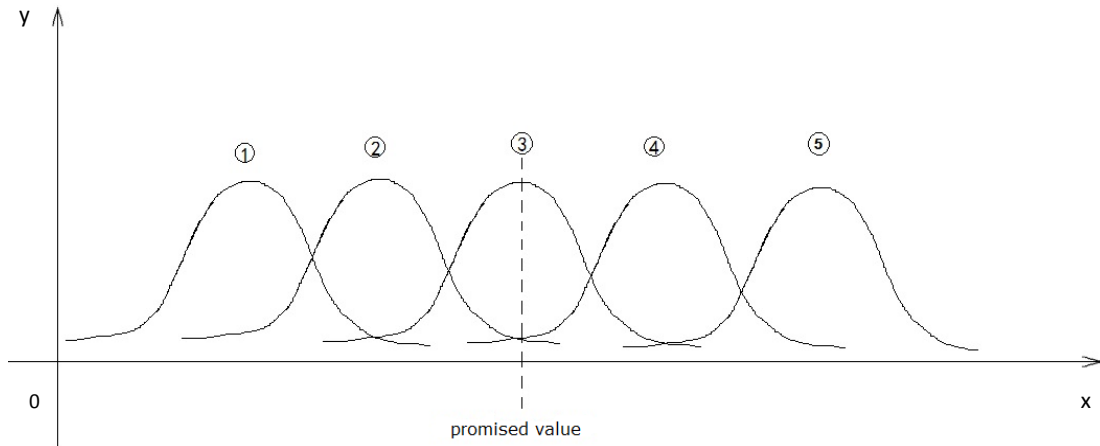


图 6.1 实验数据分布

实验产生 5 组不同的数据来模拟服务 QoS 属性在实际调用中的数值，作为历史记录，通过第四章提出的贝叶斯学习方法对属性进行客观信誉评估。

各组数据均服从正态分布： $D \sim Normal(M, \sigma^2)$ ，其中 M 为该组数据的均值， σ^2 为方差。第 3 组数据的均值为服务发布出来的承诺值，第 1 组和第 2 组数据均值均小于承诺值，第 4 组和第 5 组数据均值均大于承诺值，方差采用的是相同的值。按照随机生成的这 5 组分布特征的数据对客观信誉评估方法进行验证。服务 QoS 存在两种属性，一种是成本型属性（值越小表示质量越好），一种是效益型属性（值越大表示质量越好），下面利用这 5 组数据，对 4.2.1 小节中的信誉评估方法进行实现。实验结果如图 6.2 所示。

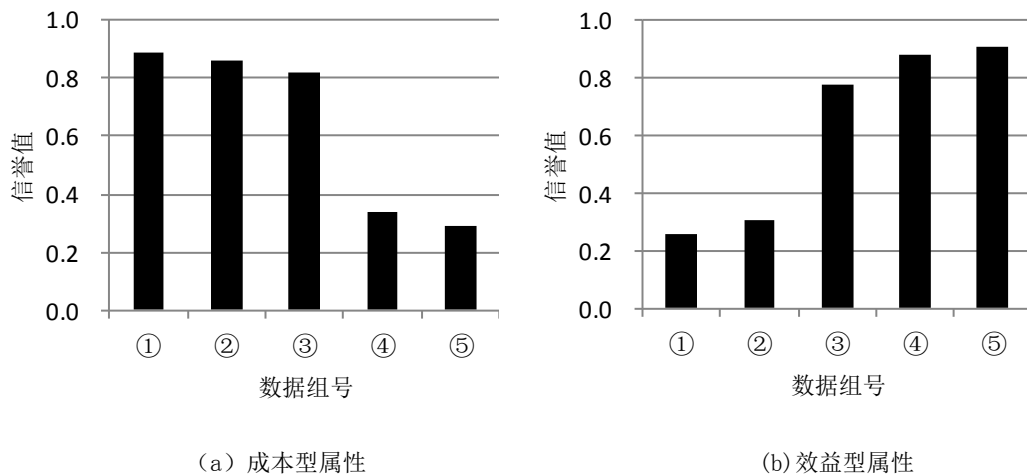


图 6.2 客观信誉评估结果

从图 6.2 中可以看到, 对于成本型属性, 第 1、2、3 组数据得到的信誉评估值都比较高, 第 4、5 组数据得到评估值则很低。分析原因可知, 前三组数据基本分布在承诺值左侧, 小于或等于承诺值, 代表服务实际调用中该属性值完成承诺的程度很高, 如第 3 组数据基本完成了承诺, 第 1、2 组数据则比承诺值还要好, 通过贝叶斯原理预测得到落入较优信誉等级的概率就比较大, 得到的信誉评估值就比较高。同理, 对于效益型属性, 前两组数据基本小于承诺值, 代表服务实际调用中该属性完成承诺的程度非常低, 后三组数据则基本完成了承诺或比承诺值更好, 因此得到的信誉评估值就比较高。可见, 此方法能够合理地反映各属性在服务实际调用中完成承诺的真实情况, 具有很高的客观正确性。因此, 引入客观信誉评估可以规避掉仅仅通过用户主观评价进行可信度评估带来的片面性, 提高服务可信度评估准确性。

6.3 服务选择方法对比

针对局部优化阶段, 即在某一个抽象服务类所存在的众多候选服务中进行服务选择, 其方法是对各候选服务进行 QoS 度量, 每个候选服务得到一个 QoS 效用值, 根据此效用值进行排序, 从而将效用值比较高的服务选择出来。传统的服务选择方法所依据的效用度量值, 仅仅是通过提供者发布的各属性的承诺值与权重加权求和得到的; 本文的服务选择方法所依据的 QoS 度量值则考虑到了服务的可信性, 是融合了各属性的可信度而得到的 QoS 效用值; 另外一种依据用户评价对服务进行选择的方法, 是仅将用户对各属性的主观评价加权求和得到的。其中属性权重的确定可由本文 4.2 节给出的方法获得。

为了将本文的服务选择方法与其他两种服务选择方法在查准率 (precision) 上进行对比, 我们将 QWS 数据集和 MSR-Bing 数据集分别模拟为一个抽象服务类, 并随机生成了 100 个用户需求, 然后为每个用户需求在数据集中人为地指定服务选择结果作为基准 (Baseline)。基准服务的指定原则为: 选择那些在 QoS 偏好和可信度方面同时满足用户需求的最优 (或近优) 的服务。对于每个服务选择请求, 我们先将不满足用户 QoS 属性约束的服务剔除, 然后对剩余的候选服务根据三种不同的方法分别进行 QoS 度量并依据度量效用值对候选服务排序, 最后为用户选出排名最靠前的一些服务。每个服务选择过程我们都分别选出 Top-1%, Top-3%, Top-5%, Top-7%, Top-9% 的候选服务, 并将不同方法选择出的服务结果与基准进行对比, 我们采用查准率 (precision) 来评估表现用户对不同服务选择方法的满意程度, 查准率 (precision) 的计算公式如下所示:

$$precision = \left(\frac{N_{contains \text{ Baseline}}}{N} \right) \times 100\% \quad (6.1)$$

其中 $N_{contains \text{ Baseline}}$ 代表服务选择结果中含有基准服务的次数, $N=100$ 代表所有服务选择的次数。实验结果如图 6.3 所示。

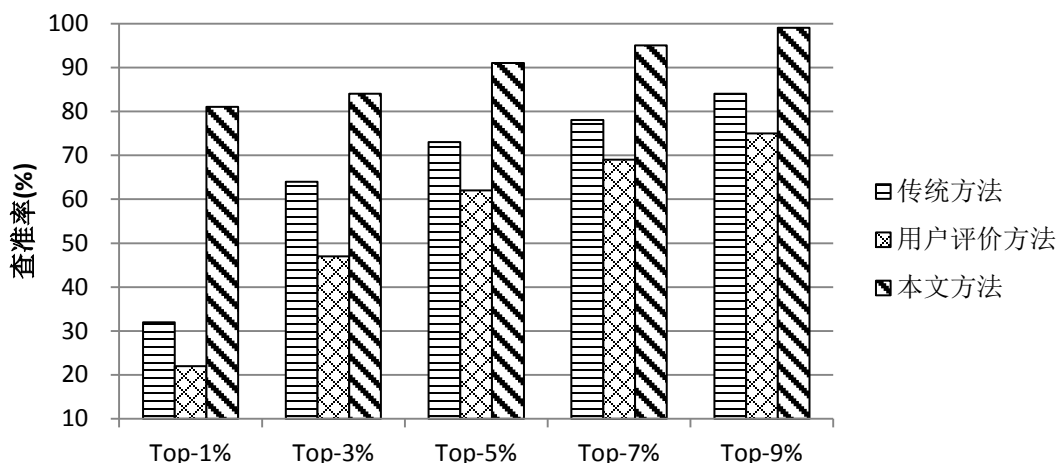


图 6.3 QWS 数据集对比结果

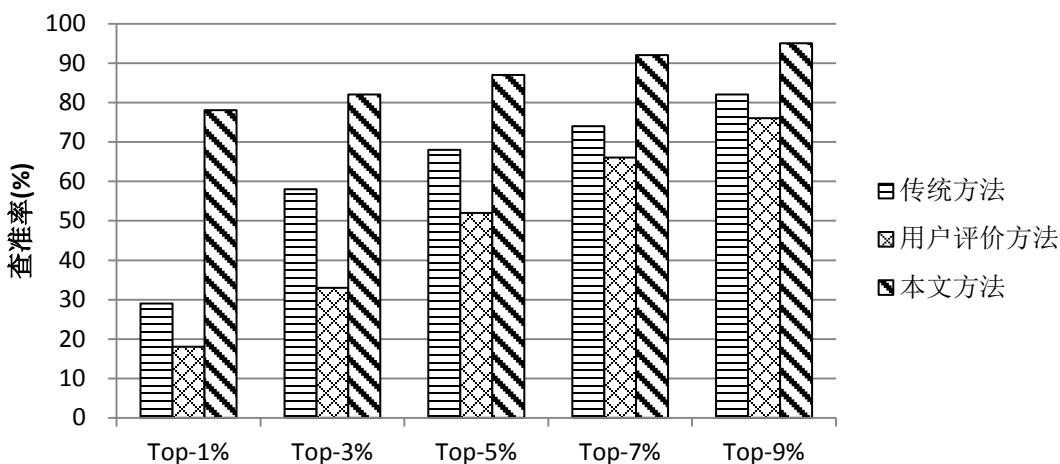


图 6.4 MSR-Bing 数据集对比结果

从图 6.3 和图 6.4 展示的三种方法查准率的对比结果来看，两个数据集所示结果具有相似的趋势，即随着选择出服务数量的增加，所有方法的查准率都会相应提高，但不论选择出的服务数量多少，本文的服务选择方法查准率均高于其他两种服务选择方法。也就是说，相比传统的服务选择方法和仅根据用户评价的服务选择方法，本文方法可以有效选择出既满足用户 QoS 要求同时保证可信性的服务。其次我们可以看出当选择出排名前 9% 的候选服务时，本文方法达到了很高的查准率，接近于 100%，所以针对这个情景，我们可以将该抽象服务类中通过 QoS 度量后选择出前 9% 的候选服务推荐给用户。

6.4 改进的布谷鸟搜索算法在 Web 服务组合全局优化过程中的对比

在全局优化阶段，本文采用了一种改进的布谷鸟优化算法（IMOCs）对 Web 服务组合多目标优化问题进行求解，为了研究该算法对服务组合的优化性能，将 IMOCs 与文献^[71]中的 MOCS 算法以及当前领域比较流行的遗传算法（GA）和蚁群算法（ACO）进行对比，讨论分析在求解 Web 服务组合优化问题时这几种算法的性能和所求解的质

量。实验过程中，通过改变服务组合中包含的抽象服务类或每个抽象服务类所包含的候选服务数量（从 5 到 30），来获得更为广泛且更具一般性的实验结果。参照文献^[49,52,70]中的参数比较结果，本文采用的算法参数设置如下所述：在遗传算法中，交叉概率 $p_c=0.8$ ，突变概率 $p_r=0.33$ ；蚁群算法中，信息素挥发率取 0.7， $\alpha=3$ ， $\beta=4$ ，蚂蚁数量为 20；布谷鸟算法中，发现概率均取 $p_a=0.5$ ；所欲算法每次迭代种群规模均为 100。每组实验均重复执行 30 次求平均值以得到较为准确的性能数据。

服务组合的性能评价指标采用 Average Normalized Error。Normalized Error 的概念为服务组合所得解于某一个优化目标上取得的值，与这个目标上所能得到的最优值之间的差别。计算方法为：

- 1) 通过优化算法在各个单独的优化目标上所能得到的最优值 $r_{optimal}$ 和最差值 r_{worst}
- 2) 服务组合方法生成的解在单独的分目标上获得的值 r_0 与最优值和最差值正则化得到服务在这个单独目标上的 Normalized Error 值：

$$r'_0 = \frac{r_0 - r_{worst}}{r_{optimal} - r_0} \quad (6.2)$$

- 3) 将服务组合得到的在各单独分目标上的 Normalized Error 值求平均即得到此服务组合解的 Average Normalized Error 值。

本文采用此评价指标来衡量各服务组合对优化目标的满足情况，从而对它们进行优劣比较。此外，算法都需要进行多次迭代得到结果，在复杂的网络环境中也通常需要实时为用户提供服务组合方案，因此响应时间也是一个十分重要的评价指标，此处的响应时间是指算法达成 Pareto 最优解的时间。因此实验从优化效率（Efficiency）和优化效果（Effectiveness）两个方面对各算法分别在三个数据集上进行了对比，实验结果如图 6.5-6.7 所示，另外表 6.2-6.3 呈现了具体的实验数据，表 6.2 呈现了各算法在三个数据集上的时间效率差异，表 6.3 呈现了各算法在三个数据集上的优化效果对比。

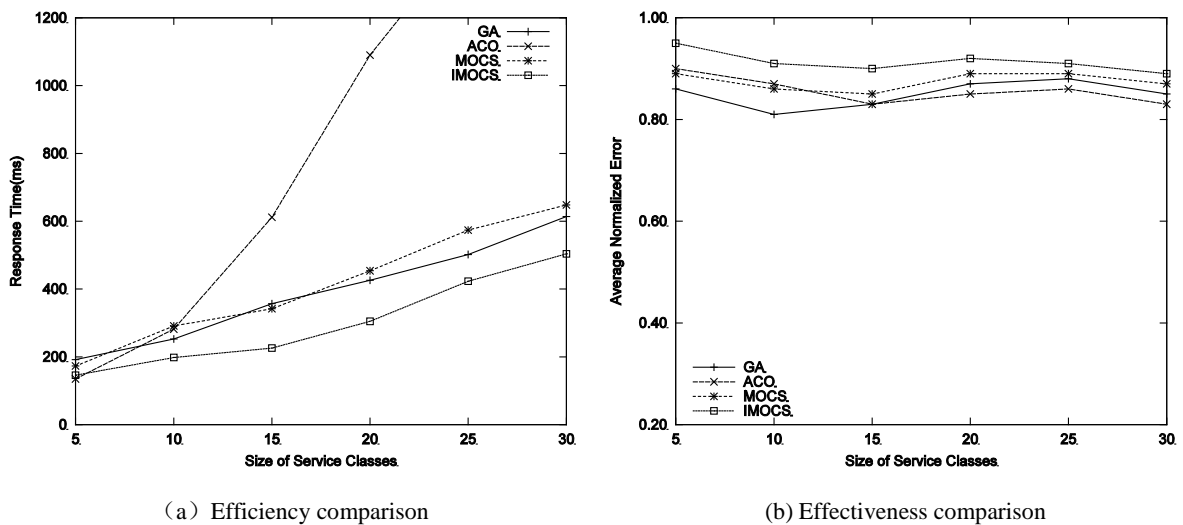
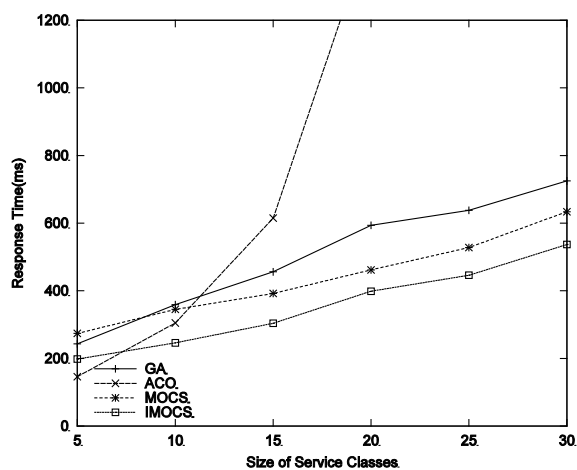
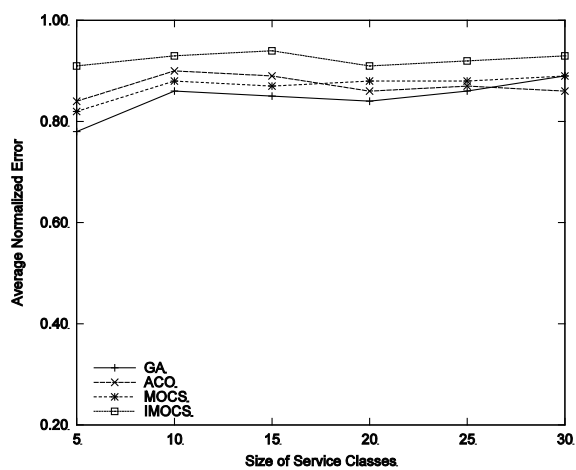


图 6.5 QWS 数据集对比结果

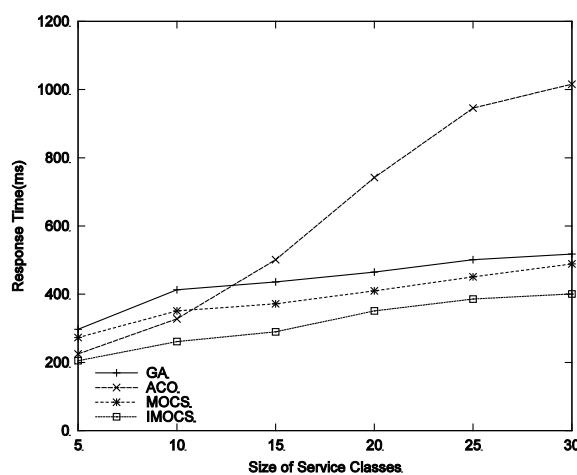


(a) Efficiency comparison

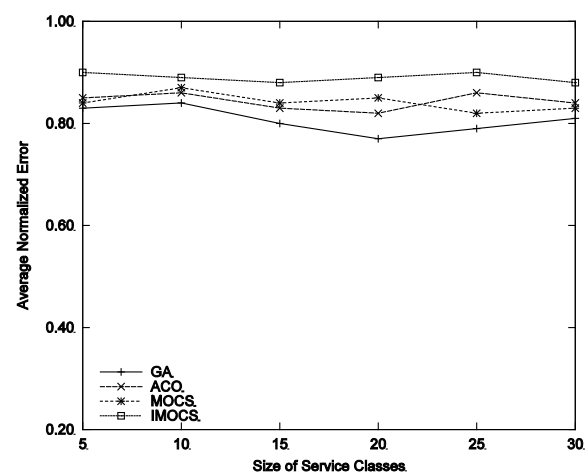


(b) Effectiveness comparison

图 6.6 MSR-Bing 数据集对比结果



(a) Efficiency comparison



(b) Effectiveness comparison

图 6.7 Synthetic 数据集对比结果

表 6.2 Efficiency Comparison

	QWS						MSR-Bing						Synthetic					
	5	10	15	20	25	30	5	10	15	20	25	30	5	10	15	20	25	30
GA	192	253	357	426	502	614	243	359	456	594	638	725	297	413	436	465	501	518
ACO	135	282	612	1090	1452	1896	146	305	615	1468	1689	1899	225	328	501	742	946	1016
MOCS	173	291	342	494	574	648	274	345	392	462	528	634	273	351	372	410	451	489
IMOCS	146	198	226	305	423	504	198	246	304	399	446	537	205	261	290	351	386	401

表 6.3 Effectiveness Comparison

	QWS						MSR-Bing						Synthetic					
	5	10	15	20	25	30	5	10	15	20	25	30	5	10	15	20	25	30
GA	0.86	0.81	0.83	0.87	0.88	0.85	0.78	0.86	0.85	0.84	0.86	0.89	0.83	0.84	0.80	0.77	0.79	0.81
ACO	0.90	0.87	0.83	0.85	0.86	0.83	0.84	0.90	0.89	0.86	0.87	0.86	0.85	0.86	0.83	0.82	0.86	0.84
MOCS	0.89	0.86	0.85	0.89	0.89	0.87	0.82	0.88	0.87	0.88	0.88	0.89	0.84	0.87	0.84	0.85	0.82	0.83
IMOCS	0.95	0.91	0.90	0.93	0.91	0.89	0.91	0.93	0.94	0.89	0.92	0.90	0.90	0.89	0.88	0.89	0.90	0.88

A. 结果与分析

图 6.5–6.7 和表 6.2、6.3 呈现了各算法在优化效率和优化效果上的对比，从各图时间效率对比结果可以观察到，这些算法中 **IMOCS** 算法的响应时间相对来说是最短的，与 **MOCS** 相比是有较好改进的。随着服务组合的工作流中任务规模的增大，各算法的响应时间也相应增加，**MOCS**、**IMOCS** 和 **GA** 算法基本小于 800ms，其时间增长是可以接受的，也就是说可以满足服务组合的实时要求。而蚁群算法在服务组合任务规模很小时有较好的时间效率，但随着任务量的增大，响应时间却迅速增长，效率下降。在优化效果对比方面，从各图结果可以看出，本文提出 **IMOCS** 算法要比其他几种算法表现的更好。由实验结果可以得出，对此 Web 服务组合的全局优化问题，这些算法都可以成功有效的得到 Pareto 最优解，而本文提出的改进的布谷鸟多目标搜索算法具有最好的性能：不仅在优化效果上有很高的值，同时响应时间也是最少的。

B. 显著性检测

此外，我们接着对表 6.2、6.3 中算法对比的实验结果数据进行了显著性差异检测，根据统计学实验“T 检测”来确认本文提出的 **IMOCS** 算法与其他三种算法是否具有显著性差异。通过显著水平为5%的双尾T检测实验得到的 p 值结果如表 6.4 所示， $p \leq 0.05$ 代表 **IMOCS** 与其他算法相比具有显著差异， $p > 0.05$ 代表差异不显著。从表 6.4 中 p 值结果可知，本文提出的 **IMOCS** 算法在所有评价指标方法上均显著的优于其他三种对比算法。

表 6.4 T 检测 p-value 结果

		GA	ACO	MOCS
QWS	efficiency	0.002	0.043	0.002
	effectiveness	0.002	0.000	0.002
MSR-Bing	efficiency	0.002	0.049	0.000
	effectiveness	0.001	0.000	0.002
Synthetic	efficiency	0.000	0.029	0.000
	effectiveness	0.001	0.000	0.002

6.5 本章小结

本章阐述了评估本文提出的基于可信度与 QoS 相结合的 Web 服务组合优化方法的实验，具体交代了实验设置和数据集，并进行了三组实验：1) 对本文提出的可信度评估方法进行实验验证，证明了该方法的有效性；2) 针对局部优化阶段，将本文提出的服务选择方法与传统的服务选择方法和用户评价选择方法进行对比，验证了本文方法的有效性；3) 针对全局优化阶段，将本文提出的改进的布谷鸟多目标优化算法与其他几种较为流行的算法进行对比，实验结果表明，在组合优化效率和组合优化效果两个方面本文提出的算法都要显著的优于其他对比算法。

第七章 总结与展望

7.1 本文总结

面向服务的体系架构(Service Oriented Architecture, SOA)作为一种新型计算模式,成为近年来互联网信息技术的新热点。SOA 注重服务,使应用程序集中起来以提供更丰富、目的性更强的业务流程。Web 服务则是 SOA 架构的一种重要的实现方式,它是基于网络的、分布式的、自描述的、模块化的组件,执行特定的任务,遵循一定的技术规范,可以在网上发布、定位和调用。随着该技术的发展,Web 服务的数量日益增长,而单个 Web 服务的功能比较有限,很难满足用户复杂的需求,所以需要依据业务需求将这些单个服务根据功能和非功能属性集成为组合服务,完成复杂的业务。

在 Web 服务组合中,服务的非功能属性通常用服务质量(QoS)表示,基于 QoS 的服务组合方法被提出后便受到了广泛关注。该方法是在系统功能实现的前提下,使得组合服务整体的 QoS 尽最大可能地满足用户的需求。传统的研究中,通常是根据服务给定的 QoS 信息进行服务选择组合,而现实中,开放、动态的网络环境由于缺少监管,服务会存在不诚信的行为,传统方法得到的高 QoS 并不能保证服务的可信性,不诚信的服务会给企业应用系统带来不可预知的后果。只有既满足可信性需求又保证服务的高质量,才能确保服务最符合用户需要。针对这一问题,本文提出了一种基于可信度计算与 QoS 相结合的 Web 服务组合优化方法。文章的主要工作成果总结如下:

首先,本文提出在 Web 服务选择组合过程中对于服务提供者发布出承诺的 QoS 信息不给予彻底的相信,并在原有研究的基础上提出新的可信性概念,从客观信誉与主观信任两个方面对服务可信度进行评估,分别给出了二者的评估方法以及可信度计算模型。

其次,本文将可信度与 QoS 相结合,同时给出一种 QoS 属性权重计算方法,提出融合了可信度的 QoS 度量模型并将其作为局部优化阶段服务选择的依据,根据可信任的 QoS 指标度量结果对候选服务进行排序选择,将选择出的既满足用户 QoS 需求且具有较高可信性的服务作为后续全局优化的备选。

第三,本文将 Web 服务组合全局优化问题构建为一个多目标优化模型,为求解此问题本文选择了当前最新的元启发式搜索算法—布谷鸟搜索算法,并在原始算法的基础上对其进行多方面的改进,采用改进后适用于本文模型的布谷鸟多目标优化算法进行全局优化求解,以得到最终的 Web 服务组合方案。

最后,本文通过在真实数据集及模拟数据集上一系列的实验,对每个阶段提出的方法进行了验证和对比,并对所有实验结果进行了详细的分析,实验结果表明了本文方法的正确性和有效性。

7.1 未来工作

从可信性的角度考虑对 Web 服务进行组合优化是一个很具有挑战性的问题，本文对此进行了初步的探索、且做了一定研究，但由于资源条件、理论水平、时间等因素，仍有很多值得未来进行改进和深入研究的地方。

首先，本文只是从一个新的角度，在现存理论的基础上对 Web 服务 QoS 的可信度进行了探究，并没有建立完整的可信度评估体系，在未来工作中需重点考虑如何将此问题建立为比较完善的评估体系。

其次，可信度评估方法的前提是对服务的历史运行记录进行监控，需要提供服务的历史调用信息，然而通过什么反馈机制实现，尚无解决方案，需进一步探讨。

最后，融合可信度的 QoS 感知的服务组合是综合了多个学科的研究问题，具有很强的前沿性和较高的实现难度，因各种限制条件未能将其作用于真实的服务组合系统中，这将是未来工作中需重点考虑的事情。

致谢

将近三年的研究生生活已经到了即将结束的时候，回想这段时光，有收获也有彷徨，有快乐也有苦涩，不论路途多么坎坷曲折，现在的我已经站在了终点，谨此在毕业论文即将完成之际，向这期间曾给予我帮助和关怀的老师和同学表示最真挚的感谢。

首先感谢我的导师王红兵教授，他严谨的治学态度、渊博的知识、独到而透彻的问题分析方式使我受益匪浅，在王老师的指导下，我得以初窥学术研究的门径，大大扩展了自己的知识面。在论文的整个研究过程中，王老师为课题的研究提供了全面的指导，为我提供了优越的实验条件，并对论文的撰写提出了有益的意见，感谢他对我悉心指导和真诚关怀。

感谢实验室的成员们，感谢王磊师兄，与你的讨论，让我的研究思路更清晰，你在科研上的指导，以及孜孜不倦的科研态度，让我受益匪浅。感谢已毕业的邹彬、俞超、周书湘、孙海霞、陈鑫、吴琴等师兄师姐们，感谢你们两年里对我在学习、科研、生活上的帮助，有你们在，原本枯燥的研究生生活也充满了欢乐。感谢同级的黄桂成、胡兴国同学，同窗三载，互帮互助，并肩作战，我们建立了深厚的友谊，成为研究生生活里最熟悉的人。感谢顾明珠、杨正萍、张幸芝、陶勇、费欢欢、李佳杰六位师弟师妹，你们的加入，让实验室注入了新的活力，我会永远怀念大家在一起的日子。

感谢我的家人，他们用勤劳的双手与无私的爱养育我长大，他们是我前进的坚实后盾，也是我最强的精神支柱，未来我会加倍努力，来回报他们长久的关怀和照顾。

再次感谢所有关心、帮助过我的老师、同学、朋友和亲人，谢谢你们。

参考文献

- [1] Trummer I, Faltings B. Optimizing the tradeoff between discovery, composition, and execution cost in service composition[C]//Web Services (ICWS), 2011 IEEE International Conference on. IEEE, 2011: 476-483.
- [2] Tabatabaei S G H, Kadir WMNW, Ibrahim S. A comparative evaluation of state-of-the-art approaches for web service composition[C]//Software Engineering Advances, 2008. ICSEA'08. The Third International Conference on. IEEE, 2008: 488-493.
- [3] Li L, Liu D, Bouguettaya A. Semantic based aspect-oriented programming for context-aware Web service composition[J]. Information Systems, 2011, 36(3): 551-564.
- [4] Milanovic N, Malek M. Search strategies for automatic web service composition[J]. International Journal of Web Services Research (IJWSR), 2006, 3(2): 1-32.
- [5] Wang H, Zhou X, Zhou X, et al. Adaptive and dynamic service composition using q-learning[C]//Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on. IEEE, 2010, 1: 145-152.
- [6] Zeng L, Benatallah B, Ngu A H H, et al. Qos-aware middleware for web services composition[J]. Software Engineering, IEEE Transactions on, 2004, 30(5): 311-327.
- [7] Yu Q, Bouguettaya A. Framework for web service query algebra and optimization[J]. ACM Transactions on the Web (TWEB), 2008, 2(1): 6.
- [8] Yu Q, Rege M, Bouguettaya A, et al. A two-phase framework for quality-aware web service selection[J]. Service Oriented Computing and Applications, 2010, 4(2): 63-79.
- [9] Zou G, Xiang Y, Gan Y, et al. An agent-based web service selection and ranking framework with QoS[C]// IEEE International Conference on Computer Science & Information Technology. IEEE Computer Society, 2009:37-42.
- [10] Jaeger M C, Rojecgoldmann G, M&Xfchl G. QoS Aggregation for Web Service Composition using Workflow Patterns[C]// Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International. IEEE Computer Society, 2004:149-159.
- [11] Ding C, Sambamoorthy P, Tan Y. QoS Browsing for Web Service Selection.[J]. Lecture Notes in Computer Science, 2009, 5900:285-300.
- [12] Zou G, Xiang Y, Gan Y, et al. An agent-based web service selection and ranking framework with QoS[C]// IEEE International Conference on Computer Science & Information Technology. IEEE Computer Society, 2009:37-42.
- [13] Xu Z, Martin P, Powley W, et al. Reputation-Enhanced QoS-based Web Services Discovery[C]// Web Services, 2007. ICWS 2007. IEEE International Conference on. IEEE, 2007:249 - 256.
- [14] Kalepu S, Krishnaswamy S, Loke S W. Reputation = f(user ranking, compliance, verity)[C]// Web Services, 2004. Proceedings. IEEE International Conference on. IEEE,

- 2004:200 - 207.
- [15]Chen M, Chen A N K, Shao B B M. The Implications and Impacts of Web Services to Electronic Commerce Research and Practices.[J]. J.electron.commerce Res, 2003, 4:128-139.
 - [16]Tweedale J, Cutler P. Trust in Multi-Agent Systems[M]// Knowledge-Based Intelligent Information and Engineering Systems. Springer Berlin Heidelberg, 2006:1-25.
 - [17]Mehdi M, Bouguila N, Bentahar J. A QoS-Based Trust Approach for Service Selection and Composition via Bayesian Networks[C]// 2013 IEEE 20th International Conference on Web Services. IEEE Computer Society, 2013:211-218.
 - [18]Jøsang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision[J]. Decision Support Systems, 2006, 43(2):618-644.
 - [19]Artz D, Gil Y. A survey of trust in computer science and the Semantic Web[J]. Web Semantics Science Services & Agents on the World Wide Web, 2007, 5(2):58-71.
 - [20]Kim Y, Doh K G. A Trust Type Based Model for Managing QoS in Web Services Composition[C]// Convergence Information Technology, 2007. International Conference on. 2007:438-443.
 - [21]Letia I A, Pop O. Web service selection based on trust[C]//Proceedings of 9th International conference on Development and Application Systems (May 2008). 2008.
 - [22]Paradesi S, Doshi P, Swaika S. Integrating Behavioral Trust in Web Service Compositions[C]// Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009:453--460.
 - [23]Li Z, Su S, Yang F. WSrep: A Novel Reputation Model for Web Services Selection[M]// Agent and Multi-Agent Systems: Technologies and Applications. Springer Berlin Heidelberg, 2007:199-208.
 - [24]Malik Z, Bouguettaya A. Reputation Bootstrapping for Trust Establishment among Web Services[J]. IEEE Internet Computing, 2009, 13(1):40-47.
 - [25]Hang C W, Singh M P. Trustworthy Service Selection and Composition[J]. Acm Transactions on Autonomous & Adaptive Systems, 2011, 6(1):627-662.
 - [26]Mehdi M, Bouguila N, Bentahar J. Trustworthy Web Service Selection Using Probabilistic Models[C]// Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012:17-24.
 - [27]Zheng H, Zhao W, Yang J, et al. QoS Analysis for Web Service Compositions with Complex Structures[J]. IEEE Transactions on Services Computing, 2013, 6(3):373-386.
 - [28]Mehdi M, Bouguila N, Bentahar J. Trustworthy Web Service Selection Using Probabilistic Models[C]// Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012:17-24.
 - [29]Mehdi M, Bouguila N, Bentahar J. Trust and Reputation of Web services Through QoS Correlation Lens[J]. IEEE Transactions on Services Computing, 2015:1-1.
 - [30]Casati F, Ilnicki S, Jin L J, et al. Adaptive and Dynamic Service Composition in[M]//

- Seminal Contributions to Information Systems Engineering. Springer Berlin Heidelberg, 2000:215-233.
- [31] Benatallah B, Dumas M, Sheng Q Z, et al. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services[J]. USA, IEEE Computer Society, 2003:297 - 308.
- [32] Ran S. A model for web services discovery with QoS[J]. Acm Sigecom Exchanges, 2003, 4(1):1-10.
- [33] Liu Y, Ngu A H H, Zeng L, et al. QoS Computation and Policing in Dynamic Web Service[J]. In World Wide Web (WWW, 2004:66--73.
- [34] Zeng L, Benatallah B, Dumas M, et al. Quality driven web services composition[C]// Proceedings of the 12th international conference on World Wide Web. ACM, 2003:411-421.
- [35] Yu T, Zhang Y, Lin K J. Efficient algorithms for Web services selection with end-to-end QoS constraints[J]. Acm Transactions on the Web, 2007, 1(1):2007.
- [36] Xia H, Li Z. Particle Swarm Algorithm for the Quality of Service-Oriented Web Services Selection[C]// Knowledge Acquisition and Modeling, 2009. KAM '09. Second International Symposium on. 2009:303-306.
- [37] Lin W, Dou W, Luo X, et al. A History Record-Based Service Optimization Method for QoS-Aware Service Composition.[C]// IEEE International Conference on Web Services, ICWS 2011, Washington, DC, USA, July 4-9, 2011. 2011:666-673.
- [38] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition[C]// International Conference on World Wide Web. ACM, 2009:881-890.
- [39] Alrifai M, Skoutas D, Risse T. Selecting skyline services for QoS-based web service composition[C]// In Proc. of the 19th International World Wide Web Conference (WWW'10. 2010:11-20.
- [40] Qi L, Tang Y, Dou W, et al. Combining Local Optimization and Enumeration for QoS-aware Web Service Composition.[C]// 2010 IEEE International Conference on Web Services. IEEE Computer Society, 2010:34-41.
- [41] Coello C C, Lamont G B, Van Veldhuizen D A. Evolutionary algorithms for solving multi-objective problems[M]. Springer Science & Business Media, 2007.
- [42] Farina M, Deb K, Amato P. Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications[M]// Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2003:311-326.
- [43] Marler R T, Arora J S. Survey of multi-objective optimization methods for engineering[J]. Structural & Multidisciplinary Optimization, 2004, 26(6):369-395.
- [44] Rangaiah G P. Multi-objective optimization: techniques and applications in chemical engineering[M]. world scientific, 2009.
- [45] H. A. ABBASS, R. SARKER. THE PARETO DIFFERENTIAL EVOLUTION

- ALGORITHM[J]. International Journal on Artificial Intelligence Tools, 2011, 11(4):531-552.
- [46]Marler R T, Arora J S. Survey of multi-objective optimization methods for engineering[J]. Structural & Multidisciplinary Optimization, 2004, 26(6):369-395.
- [47]Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6):712-731.
- [48]Adamatzky A. Unconventional Computing[J]. International Journal of General Systems, 2014, 43(7):671-672.
- [49]Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. on Evol[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182 - 197.
- [50]Davis L. Genetic Algorithms and Simulated Annealing[M]. Morgan Kaufmann Publishers Inc., 1987.
- [51]Liu M, Wang M, Shen W, et al. A quality of service (QoS)-aware execution plan selection approach for a service composition process[J]. Future Generation Computer Systems, 2012, 28(7):1080-1089.
- [52]Zhang W, Chang C K, Feng T, et al. QoS-Based Dynamic Web Service Composition with Ant Colony Optimization[C]// Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual. IEEE, 2010:493-502.
- [53]喻坚 韩燕波. 面向服务的计算--原理和应用[M]. 清华大学出版社, 2006.
- [54]Mockford K. Web Services Architecture[J]. Bt Technology Journal, 2004, 22(5):19-26.
- [55]Pires P F, Benevides M R F, Mattoso M. Building Reliable Web Services Compositions[J]. International Workshop Web Services Research Standardization & Deployment, 2003, 2593:59--72.
- [56]Sirin E, Parsia B, Wu D, et al. HTN planning for Web Service composition using SHOP2[J]. Web Semantics Science Services & Agents on the World Wide Web, 2004, 1(4):377-396.
- [57]Ponnekanti S R, Fox A. Sword: A developer toolkit for web service composition[C]//Proc. of the Eleventh International World Wide Web Conference, Honolulu, HI. 2002, 45.
- [58]Casati F, Ilnicki S, Jin L J, et al. Adaptive and dynamic service composition in eFlow[C]//Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2000: 13-31.
- [59]Shehu U, Epiphaniou G, Ali Safdar G. A Survey of QoS-aware Web Service Composition Techniques[J]. International Journal of Computer Applications, 2014, 89(12):10-17.
- [60]Bryson J J, Martin D L, Mcilraith S A, et al. Toward Behavioral Intelligence in the Semantic Web[J]. Computer, 2002, 35(11):48-54.
- [61]龚光鲁. 概率论与数理统计[M]. 清华大学出版社, 2006.
- [62]Pawlak Z. Rough sets[J]. International Journal of Computer & Information Sciences,

- 1982, 11(5): 341-356.
- [63]Pawlak Z, Grzymala-Busse J, Slowinski R, et al. Rough sets[J]. Communications of the ACM, 1995, 38(11): 88-95.
- [64]Yang X S. Firefly algorithm[J]. Nature-inspired metaheuristic algorithms, 2008, 20: 79-90.
- [65]Yang X S. Biology-Derived Algorithms in Engineering Optimization[J]. Chapman & Hall, 2010, 58.
- [66]Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison[J]. ACM Computing Surveys (CSUR), 2003, 35(3): 268-308.
- [67]Yang X S, Deb S. Cuckoo search via Lévy flights[C]//Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, 2009: 210-214.
- [68]Zeng L, Benatallah B, Dumas M, et al. Quality driven web services composition[C]//Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 411-421.
- [69]Bouguila N. Clustering of count data using generalized Dirichlet multinomial distributions[J]. Knowledge and Data Engineering, IEEE Transactions on, 2008, 20(4): 462-474.
- [70]Andersson J. A Survey of Multiobjective Optimization in Engineering Design[C]//ReportsoftheDepartementofMechanicalEngineering, LiTHIKP-R-1097. 2010.
- [71]Yang X S, Deb S. Multiobjective cuckoo search for design optimization[J]. Computers & Operations Research, 2011, 40(6): 1616-1624.
- [72]Walton S, Hassan O, Morgan K, et al. Modified cuckoo search: a new gradient free optimisation algorithm[J]. Chaos, Solitons & Fractals, 2011, 44(9): 710-718.
- [73]Srinivas N, Deb K. Muultiobjective optimization using nondominated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3):221-248.

附录 A 攻读硕士学位期间科研成果

Wang H, Zou B, Guo G, et al. Integrating Trust with User Preference for Effective Web Service Composition[J]. IEEE Transactions on Services Computing, 2015:1-1.