服务计算中服务质量的多目标优化模型与求解研究

林 闯[1)] 陈 莹[1)] 黄霁崴[2)] 向旭东[3)]

[1)](清华大学计算机科学与技术系 北京 100084)

[2)](北京邮电大学网络与交换技术国家重点实验室 北京 100876)

[3)](北京科技大学计算机与通信工程学院 北京 100083)

摘 要 随着信息技术的不断发展和进步,传统的面向组件和系统的架构模式逐渐演变成面向服务的设计模式.服务计算作为一种新兴的计算模式应运而生并广泛应用于各个领域.随着用户和服务供应商需求日趋多样化,如何对服务系统进行最佳配置和管理,提供最优的服务质量,越来越受到研究者的关注,服务计算中服务质量的多目标优化成为研究热点.由于不同目标之间可能存在相互制约和折中的关系,多目标优化问题面临着难题和挑战.文中从服务计算中广泛关注的多维度指标体系出发,结合具体的研究问题,总结了 5 种典型的多目标优化模型,并从适用性、求解难度等多个角度对它们进行了分析和比较,同时讨论了模型的相互关系.对应优化模型,介绍和分析了常用的多目标优化求解方法.最后,对全文进行了总结,并对下一步的研究方向进行了展望.

关键词 服务计算;服务质量;多目标指标体系;多目标优化

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2015.01907

A Survey on Models and Solutions of Multi-Objective Optimization for QoS in Services Computing

LIN Chuang[1)] CHEN Ying[1)] HUANG Ji-Wei[2)] XIANG Xu-Dong[3)]

[1)](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

[2)](*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876)

[3)](*School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing* 100083)

Abstract With the rapid development of information technology, the IT infrastructure paradigm is shifting from component based architecture to service-oriented one. Services computing is hence emerging as a new computing paradigm and is widely used in many fields. As a number of demands from several aspects have arisen for both services and services computing systems, how to configure and manage the systems to satisfy multi-criterion Quality of Service (QoS) requirements from customers and service providers has become a hot topic. Because of possible complex relationships and tradeoffs among different attributes, such multi-objective optimization faces challenges. In this paper, we make a survey on the existing research of multi-objective optimization in services computing. The metrics in optimization are introduced, and five typical models are summarized and compared from several aspects such as applicability and complexity, their relationships are discussed. The corresponding methods for solving these models are introduced and analyzed. Finally, we prospect the possible research challenges in the future.

Keywords services computing; quality of service; multi-objective metrics; multi-objective optimization

收稿日期:2014-06-19;最终修改稿收到日期:2015-02-14.本课题得到国家自然科学基金(61472199)、国家自然科学基金重大国际合作项目(61020106002)和清华大学自主科研项目(20121087999)资助.**林 闯**,男,1948 年生,博士,教授,博士生导师,主要研究领域为计算机网络、系统性能评价、网络安全分析和 Petri 网.E-mail:chlin@tsinghua.edu.cn.**陈 莹**,女,1990 年生,博士研究生,主要研究方向为服务计算系统的性能评价和优化.**黄霁崴**,男,1987 年生,博士,讲师,主要研究方向为服务计算、系统性能模型、评价和优化.**向旭东**,男,1986 年生,博士研究生,主要研究方向为性能评价和最优控制.

1 引 言

近年来,互联网的发展与普及为计算机软件的设计模式带来了新的思路与挑战.随着 SOA 架构和 Web 服务技术的出现,传统的基于硬件和系统的架构模式逐渐转变为面向服务的设计模式.服务计算(Services Computing)作为一种新兴的计算模式应运而生.服务(Service)是指服务提供商和用户之间为实现特定的商业目标或解决方案的一种契约关系.服务计算则是服务提供商将特定的功能封装成为遵循统一标准的、交互式的计算机程序,将其以服务的形式提供给不同商业领域中的多个用户使用的一种计算模式[1].它构成了联结商业服务与信息服务的桥梁,带来了商业模式和计算模式的转型.服务计算技术旨在预先定义好的标准框架下对服务过程进行设计、操作、管理和优化,以满足日益增长的动态的业务需求.服务计算涉及到 SOA 架构,Web 服务,网格计算,云计算等.随着服务计算的逐渐普及,用户需求的不断提高,服务计算系统呈现出动态性、松耦合、大规模等显著特点.对服务计算的研究,形成了涉及服务科学与信息技术的交叉学科,成为了目前学术界和工业界的研究热点.

服务计算兴起之初,绝大多数研究主要关注于服务计算功能性(Functional)需求的满足.随着服务计算被广泛应用,各种类型的应用对系统和服务提出了越来越高的要求,其中以服务质量(Quality of Service,QoS)为代表的非功能性需求,逐渐被关注.如何对服务计算进行优化,选择最佳的系统配置和服务解决方案,以适应用户和服务供应商的需求,成为研究者们关注的问题.服务计算的优化过程覆盖服务系统的整个生命周期,涉及到多个学科,如运筹学、复杂系统建模、系统工程等[1].服务计算的优化是服务计算领域中的一个重要研究问题.

起初研究者们在服务计算的优化过程中仅关注某一个指标,如最小化响应时间或最大化服务收益,因此该类问题常被刻画为单目标优化的问题[2].针对单目标优化的问题,目前已有一些经典的方法能进行求解,如凸优化方法、动态规划方法等[3-4].随着用户和服务提供商需求多样化,关注的指标和属性逐渐增加,研究者们开始考虑将多个目标综合在一起进行优化,如何解决服务计算的多目标优化问题成为研究的热点.多目标优化问题不同于单目标优化问题,不同的目标之间可能存在权衡和折中的关系,很可能难以找到满足所有目标的单一最优解.因此单目标优化的方法无法直接适用于多目标优化,多目标优化在基础理论和实现方法上面临了新的挑战,解决多目标优化问题需要寻求新的思路和方法.

本文综述了服务计算中多目标优化的模型和方法研究.首先,系统地介绍了服务计算优化中所关注的多目标参数,包括 4 维指标体系和这些指标所涉及的服务计算中经典的问题.其次,总结了服务计算多目标优化最常用的 5 种模型,分别是线性加权、基于相互关系、ε-约束、帕累托和基于回报值的模型.并从适用性、解的优劣等方面对这 5 种模型进行了分析和比较.随后,介绍了这 5 种多目标优化模型对应的解法.最后对全文进行了总结和对服务计算领域进行了展望.

2 服务计算中多目标参数

服务包含服务供应商和服务消费者这两个基本的要素[1].服务供应商提供相应的服务给消费者,以实现一定的功能、达到一定的目标.消费者对应于用户,他们接收并使用供应商提供的服务,同时为此支付一定的花费.服务计算的多目标优化旨在通过一定的优化技术来提供最优的服务,满足用户、供应商的需求.由于关注的指标不一样和解决的问题不一样,服务计算的多目标优化问题也不尽相同.在本节中将分别根据关注指标和核心抽象对服务计算多目标优化进行分类总结.

2.1 指标体系

服务计算的多目标优化涉及到一系列的指标,它们代表着用户、供应商的需求.这些指标有的作为优化的目标,需要被最大化或者最小化;有的作为约束变量,需要满足一定的限制条件.近年来,随着定义服务等级的 SLA(Service Level Agreement)广泛应用在服务计算中,和指标相关的约束条件也经常被包含在优化问题中[5].本文将从系统性能指标、安全可信指标、能源指标和经济指标这几个方面来介绍服务计算多目标优化的指标体系.

系统性能指标反映了系统的处理能力或者效率,它是一个综合的因素,包括吞吐率、响应时间、阻塞率和利用率.吞吐率刻画了系统的产量,在不同的系统中,吞吐率的单位可能不一样,例如在传统的计算机网络和通信系统中,吞吐率是每单位时间通过的比特数或字节数,而在服务计算中,吞吐率可以指系统单位时间内通过的请求数或任务数[6].响应时

间是指服务从请求提交到服务完成这段过程中花费的时间.具体地,响应时间又包含了等待时间、服务时间.对于很多实时业务来说,响应时间是一个关键的因素,响应时间过大往往会影响用户体验和用户对服务的满意程度,进而影响供应商的收益.因此,在服务计算的管理和优化中,响应时间常作为最小化的目标,或者包含在 SLA 中并以约束条件的形式出现[7-10].当系统处于繁忙状态,提交的任务无法得到及时的处理,阻塞率刻画的就是这一情况的概率[11].利用率代表了系统资源的使用情况,即在给定的时间段中,系统的部件(包括硬件和软件)被使用的比例.利用率高代表了系统资源得到了较为充分的使用,而利用率低则表示系统资源存在浪费.如何优化系统的资源利用率亦是服务计算设计和改进中一个重要的科研问题[12-16].

服务计算的安全可信指标是衡量系统服务能力的重要指标[17].安全可信指标也是综合的指标,包括可靠性、可用性、可维护性、保险性、完整性和机密性.可靠性是指系统能够持续不间断地提供服务的能力[18].可靠性也常用平均失效时间(Mean Time to Failure,MTTF)来刻画,它是一个稳态的量,代表了系统从正常状态到失效状态平均花费的时间.可用性是指当请求需要时系统处于可工作状态的概率,又可以分为瞬时可用性、稳态可用性及固有可用性[19].可维护性描述了系统调整、修复和容错的能力[17].保险性定义为系统在运行生命周期内,不对用户和环境造成灾难性后果的概率[20].完整性是指系统抵御不适当修改的能力[17].

近年来,随着服务计算中能量消耗的不断增大,能源指标也不断吸引着研究者们的关注.文献[21]指出,服务系统所耗费的能源在全美的所有能源中占到了 1.5%的比例,而且这个比例仍在不断上升.因此,如何在满足系统服务需求的同时降低能耗,是亟待解决的问题.此外,能量消耗过程中所产生的碳氧化合物对环境有不可忽视的影响,很多文献也将碳氧化合物的排放作为优化过程中考虑因素[22-24].

服务计算的经济指标包括系统的收益和开销,它们直接影响服务消费者和服务供应商的利益,是服务计算中重要的指标,常常在优化的过程中被考虑[1].对于服务消费者,如何采用最小的花费,来得到所需的服务,是首要关注的问题;而服务供应商则更关注于在满足用户需求的前提下如何最大化系统的资源利用率,最小化系统的开销,从而增加他们的利润[10,12].

这四维指标之间并不完全独立,它们两两之间皆相互有一定的影响.例如,性能指标中的利用率和安全可信指标中的可靠性皆对服务器的能耗有一定的影响.而能耗又将影响系统的开销.具体地,服务计算中的指标体系如图 1 所示.

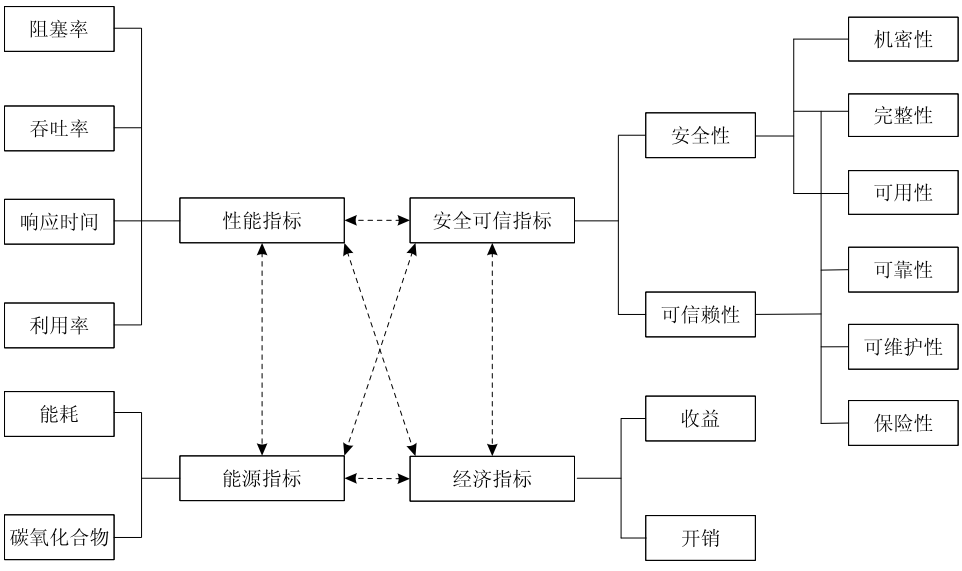

图 1　服务计算的指标体系

2.2　核心抽象

服务计算的多目标优化旨在通过优化的方法使得系统处于最佳的运行状态,提供最优的服务,它在很多方面都有所应用,可以抽象成很多问题.下面本文介绍 3 种核心的问题抽象,分别是资源管理、任务调度和服务组合.

资源管理是服务计算中核心的问题,它监控和维护系统的资源状态,并对资源进行管理.资源管理

关注和优化的指标很广泛,包括资源的利用率、系统的开销等,对服务供应商的利益有直接的影响.良好的资源管理策略有利于提高系统的运行和服务能力,从而降低响应时间等,提高用户的满意度.一般而言,资源包括硬件资源,如服务器、存储空间等;也包括软件资源,如数据库、共享程序模块等.文献[14,25]研究系统的最基本的硬件资源——CPU 和内存.近年来,随着虚拟化技术[26]广泛应用在服务计算中,虚拟数据中心的发展也得到了学术界和工业界的关注,虚拟资源成为一种新兴的资源,关于虚拟数据中心中虚拟机的资源管理成为研究的热点.文献[12,15,27]将虚拟机作为系统资源的单位,讨论在系统中如何管理这些虚拟机.一般而言,虚拟机中需要配置一定的物理资源来保障它正常运行.

更进一步,资源管理又可以划分为资源分配和资源部署.资源分配研究在给定资源总量的情况下,如何将这些资源合理地分配给相应的部分,从而实现系统的某些目标.而资源部署则侧重于给定的资源需要如何部署在系统部件中,例如文献[15]研究虚拟机的部署问题,在给定虚拟机资源需求的情况下,如何找到合适的物理机来承载这些虚拟机,从而达到在最小化资源浪费的同时实现最小化能耗.在很多情况下,资源分配和资源部署常常结合在一起来实现系统的资源管理.如文献[12]研究如何根据到达的请求来计算所需要的资源并完成分配,接着解决分配的资源如何部署能实现在降低开销的同时满足资源利用率的最大化问题.

任务调度也是服务计算中重要的问题,旨在研究当请求或者任务到达时,如何将它们调度到合适的执行单位中,并提供最优的服务,以满足用户、供应商的需求.一般地,根据任务的队列个数、执行任务的服务器个数,任务调度的问题可以分为单队列多服务器,多队列单服务器,多队列多服务器[28].根据任务的相关知识是否确定,任务调度又可以分为静态调度和动态调度[29].静态调度中,任务的到达和执行时间事先确定,因此可以在任务执行之前,提前为任务制定好调度的策略.而服务计算中,由于负载的高度不确定性,静态调度难以适应负载的抖动性和突发性,很多调度都需要根据实时情况动态地进行调整[6].请求路由是服务计算中一类常见的任务调度问题.如文献[22]研究在有多个数据中心且各个数据中心的响应时间、能源各异的情况下,当有请求到达时,如何将这些请求分配到不同的数据中心,以实现响应时间、能耗、碳排放的多目标优化;文献[6]研究如何将请求合理地调度到不同的服务器和虚拟机上,以在满足响应时间限制的同时,最大化系统的吞吐量并最小化系统的能耗.

在服务计算中,一些较综合的服务往往需要得到其他服务的支持,将若干个服务联合起来组成一个工作流,也即服务组合(Service Composition),从而提供较为复杂、全面的服务,以实现一定的功能[1].典型的例子包括旅游行程规划的服务,它涉及到订购机票、预定旅馆、租车等一系列服务[30].服务组合问题中常包括服务选择(Service Selection)问题,即当服务有多个候选项的情况下,如何选择最佳的候选项来提供服务,从而最好地满足用户、服务供应商的需求[31].近年来,随着服务数量和规模的增大,很多供应商开始提供功能类似但具有不同等级的服务.以服务质量(QoS)为代表的非功能参数越来越受到研究人员的关注[30].很多研究致力于解决 QoS 相关的服务组合/服务选择问题.而 QoS 是一个综合的指标,它包括响应时间、吞吐率、可靠性等,因此这些问题常被抽象成多目标优化的问题[5,10,31-39].如文献[5]解决了服务组合中如何最大化系统吞吐率同时最小化响应时间、开销的多目标优化问题,文献[31]综合考虑了开销、响应时间和可靠性的多目标服务组合问题.

这些问题关注了一系列的指标,涉及到服务计算不同的层次.如服务组合问题及优化的经济指标均涉及到服务计算的业务层,资源管理问题和系统性能优化涉及到服务计算的技术层.这 3 种问题还涉及到服务系统生命周期的不同阶段,如资源管理问题涉及到服务生成和服务发布的阶段,服务组合问题涉及到服务管理的阶段.

具体地,服务计算中多目标优化的指标体系和核心抽象如图 2 所示.基本的优化过程是先对多目标优化进行模型抽象,接着设计求解方法,最后进行部署实施.

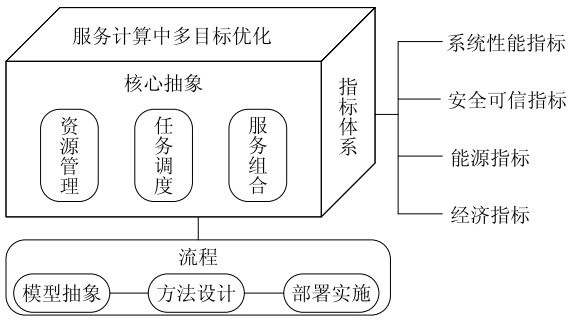

图 2 服务计算多目标优化基础理论及流程

3　多目标优化模型

通俗地说，多目标优化问题是包含多个目标函数的优化问题. 更一般地，多目标优化问题可以用如下的数学模型表示：

$$\begin{aligned}&\text{maximize/minimize}\quad f_1(x)\\&\text{maximize/minimize}\quad f_2(x)\\&\qquad\vdots\\&\text{maximize/minimize}\quad f_k(x)\\&\text{subject to}\quad x\in S\end{aligned}\tag{1}$$

其中 $k\geq 2$ 是优化的目标函数的个数，S 是系统决策变量的可行域. 变量的可行域也可以约束条件的形式出现，如

$$\begin{aligned}&\text{subject to } g_i(x)\leq 0,\ i=1,2,\cdots,m\\&\qquad\quad h_i(x)=0,\ i=1,2,\cdots,p\end{aligned}\tag{2}$$

其中 m 和 p 分别代表约束的个数. maximize/minimize 代表目标函数优化的方向，不同目标的优化方向不尽相同，如吞吐率的优化方向是最大化，而响应时间是最小化. 为了使模型看起来更加简洁，可以将不同的目标进行最大化和最小化的转换，如对于最大化的目标，统一对它们取相反数，那么式(1)可进一步化简成

$$\begin{aligned}&\text{minimize } \{f_1(x),f_2(x),\cdots,f_k(x)\}\\&\text{suject to } x\in S\end{aligned}\tag{3}$$

多目标优化问题不同于单目标优化的问题：单目标优化存在一个全局最优的解，它们或满足系统响应时间最低，或满足系统开销最小；而多目标优化往往难以存在一个满足所有目标函数的最优解. 例如，想要在最小化响应时间的同时最小化系统开销可能是不切实际的，因为响应时间低意味着服务等级高，那么很可能为了高的服务等级需要多支付一定的费用. 因此单目标优化的方法难以直接应用到多目标问题中，解决多目标优化问题需要借助其他的思路. 下面总结并介绍服务计算多目标优化中常用的 5 种模型.

3.1　线性加权

线性加权是多目标优化广泛使用的一种模型. SAW(Simple Additive Weighting)是其中经典的一类线性加权求和方法. 它忽略不同目标函数有不同的单位和范围，通过给不同的目标函数制定相应的权重，将所有的目标函数进行线性加权，用一个综合的效用函数来代表总体优化的目标[36,39]. 最优的效用函数对应的解即被认为是问题的最优解，从而将多目标优化问题转化成单目标优化问题. 对于第 i 个目标函数 $f_i(x)$，用 w_i 表示它的权重，那么多目标优化模型可以转化成

$$\begin{aligned}&\underset{x}{\text{minimize}}\ \sum_{i=1}^{k}w_i f_i(x)\\&\text{suject to } x\in S\end{aligned}\tag{4}$$

SAW 模型中主要包括两个步骤，第 1 个是缩放，第 2 个是制定权重. 缩放过程统一将各个目标函数从它们的原始值缩放，或和目标函数的最大值、最小值比较，或和目标函数的平均值比较. 如针对目标函数 $f_i(x)$，已知它的最大值是 $f_i^{\max}$，最小值是 $f_i^{\min}$，文献[36]采用的缩放方式如式(5)

$$f_i'(x)=\frac{f_i^{\max}-f_i(x)}{f_i^{\max}-f_i^{\min}}\tag{5}$$

缩放之后是制定权重的过程，可依据不同目标函数的优先级，或者用户、供应商对不同目标的偏好程度，来相应地给它们制定权重. 例如优化目标包括最小化响应时间的同时最小化开销，但是用户对响应时间比对开销更为敏感，那么响应时间的权重可以制定得高一些.

线性加权模型，其优点在于实现简单，仅用缩放后的值来代表原目标，求解也相对比较容易. 其缺陷在于刻画目标和解不够精细，例如响应时间和开销，这两个目标的单位分别是时间和金钱，用先缩放再加权的方法把它们直接相加，对原始目标的信息有一定的丢失和遗漏[40]. 另外，缩放过程需要提前知道目标的信息，如最大值、最小值或者平均值，而这些信息往往很难确定. 而制定权重过程需要依据的用户、供应商对不同目标的偏好程度也很难提前获知. 即使在已了解偏好程度的情况下，如何准确地制定权重仍然是棘手的问题. 例如，将响应时间的权重设为 0.2 还是 0.21，对于用户来说可能没有大的区别，但是对最优解有不可忽略的影响. 因此，采用线性加权模型虽然简便，但解的优劣程度难以保证.

3.2　基于相互关系

多目标优化过程中，有时多个评价指标(优化目标)之间并非独立存在，可以存在一定的相互影响或制约的关系. 在这样的情形下，线性加权的方法可能不再适用，因为(1)将具有相互关系的指标进行线性相加缺乏理论依据和客观标准；(2)存在相互关联的指标的权值亦非相互独立，确定权值存在难度. 对此，一些研究考虑指标之间的相互关系，提出了评

价公式,用以对多个指标进行更好的综合优化.

一个典型的基于相互关系的评价公式为计算机网络中延迟和吞吐量综合优化过程中常用的 *Power* 公式[41],如式(6)所示.其中,*Throughput* 为网络的吞吐量,*ResponseTime* 为延迟,α 为常数,$0<\alpha<1$. *Power* 公式考虑了吞吐量和延迟之间的相互关系,可以用来评价网络中资源分配策略的有效性,用以确定最优负载. *Power* 公式亦可用来评价 Web 服务的效率[42].

$$Power=\frac{Throughput^{\alpha}}{ResponseTime} \tag{6}$$

3.3 ε-约束

ε-约束(ε-constraint)的多目标优化模型由 Haimes 等人于 1971 年提出[43],它从 k 个目标中选择一个作为优化的目标,剩余的$(k-1)$个目标则通过加界限的方式转化为约束条件.对于最小化的目标,加入上界作为限制条件;对于最大化的目标则加入下界作为限制条件.例如,模型可将式(3)转化为

$$\begin{aligned}&\underset{x}{\text{minimize}} \quad f_i(x)\\ &\text{subject to} \quad f_j(x)\leq\varepsilon_j,\ \forall j=1,\cdots,k,\ j\neq i\\ &\qquad\qquad x\in S\end{aligned} \tag{7}$$

其中 $f_i(x)$ 目标函数被选择为最终优化的目标,而对于其他的目标函数 $f_j(x)$,ε_j 是它们约束条件的上界.

ε-约束模型通过将目标转变成约束条件的形式,将原多目标优化问题转换成单目标优化问题,之后即可用单目标优化的方法来求解该问题.在实际场景中,可根据不同目标的优先级和用户、供应商的偏好程度来选择被优化的目标.如供应商最关注吞吐率,那么可把吞吐率作为优化目标,而其他指标作为限制约束条件.文献[44]采用 ε-约束的方法研究虚拟机资源管理问题,他们综合考虑了开销和响应时间,并把开销作为优化目标而响应时间作为约束条件.文献[45]用 ε-约束解决服务计算中资源部署的问题,作者将系统利益作为优化目标而 QoS 作为限制条件,从而实现最大化系统利益同时满足 QoS 约束.文献[46]研究服务计算中系统收益和性能的多目标优化,采用 ε-约束的方法将性能作为约束条件,而收益作为优化目标,解决服务计算中服务管理和调度问题.ε-约束模型的优势在于转换简便.但模型中需要设定界限值 ε_j,ε_j 设置的合适与否对最优解有一定的影响,解的优劣程度难以保证.另一方面,ε-约束引入了新的约束变量 ε_j,增加了优化问题中的约束条件个数,导致模型求解的难度往往会加大.

3.4 帕累托模型

帕累托(Pareto)是多目标优化中经典的模型[47],下面给出帕累托相关的概念.首先介绍两个解之间的比较以及如何判断一个解的优劣情况.当最小化目标时,对于任意两个可行解 $x_1,x_2\in S$,当条件(8)成立时,x_1 被称为帕累托优胜($\succ$)x_2.

$$\begin{aligned}&f_i(x_1)\leq f_i(x_2),\ \forall i\in\{1,\cdots,k\}\\ &f_i(x_1)<f_i(x_2),\ \exists i\in\{1,\cdots,k\}\end{aligned} \tag{8}$$

条件(8)即是 x_1 对应的所有目标函数值都不大于 x_2 的目标函数值,且至少有一个目标函数严格地小于 x_2 的目标函数.当目标函数需要被最大化时,对帕累托比较没有大的影响,只需修改下比较的符号即可,如式(9)所示.

$$\begin{aligned}&f_i(x_1)\geq f_i(x_2),\ \forall i\in\{1,\cdots,k\}\\ &f_i(x_1)>f_i(x_2),\ \exists i\in\{1,\cdots,k\}\end{aligned} \tag{9}$$

帕累托模型中常出现两个解之间无法判断优胜关系的情况.例如目标函数包括响应时间、吞吐率两个优化目标,解 x_1 对应的目标函数值为(10,20),而解 x_2 对应的目标函数值为(5,10),那么解 x_1 的响应时间比解 x_2 的差,但是解 x_1 的吞吐率要优于解 x_2.这实际上代表了不同目标和不同解间的权衡,除非已经提前获悉响应时间和吞吐率的优先级,否则 x_1,x_2 这两个解不分胜负.帕累托最优解即是这样一组解的集合,它包含了所有不被任何其他解所优胜的解,并且集合中的解没有好坏之分.更直观地,帕累托最优解集合的数学模型可以表示为$\{x\mid\neg\exists x'\succ x,\ \forall x'\in S\wedge x'\neq x\}$.如文献[48]研究服务计算中的服务组合问题,并考虑了包括响应时间、吞吐率等多个目标的优化问题,采用帕累托模型刻画并求解最优解.

帕累托模型由于不需要对目标进行缩放和归一化,也不需要设定或者引入新的参数、变量(如权重、界限值),直接基于原始目标函数和值进行操作,可以适用于任何目标、任何函数.它不会丢失目标函数和解的信息,解的优劣可以较好保证.但帕累托模型的最优解是一个集合,其中包含不止一个最优解,因此要穷尽并求出所有的帕累托最优解有一定的难度.

3.5 基于回报值的模型

基于回报值的优化模型旨在深入刻画优化的本质需求,将评价指标转化为回报值,描述系统或服务的指标属性对优化目标的作用,针对需求进行更为

合理、有效的优化.基于回报值的优化往往结合性能评价同时开展[49].在优化过程中,研究不同评价指标对求解目标的本质影响,形式化刻画评价指标与回报值的量化关系,给出转化公式;随后将回报值进行叠加,得到最终的优化函数.

不同需求情形下,回报值的意义不尽相同.例如,在考虑服务过程时,回报值可以定义为服务的利润,即收益减开销.在考虑服务计算的性能和能耗的综合优化过程中,可以将性能转化为完成用户任务所得到的收益,性能的高低决定了收益或收益速率的多少;将能耗转化为运行服务的开销,统一用经济模型进行表述[50-51].这样,将性能和能耗的综合优化问题转化为服务过程中的利润最大化问题,不仅符合服务计算的商业模式需求,也可以为优化提供理论依据.

进一步,文献[52]将回报模型扩展至性能、可信赖性和能耗的综合优化中,在考虑性能收益的同时,考虑服务计算系统发生故障和进行修复的过程.由于故障状态下系统无法完成用户请求,因而服务收益为0;但系统运行过程中依然有能耗等运行开销,因此故障状态下回报值为负值.这样,除性能和能耗外,可信赖性也可转化为基于商业模式的回报模型,从而可以与性能和能耗同时进行综合优化.

基于回报值的多目标优化思路是对优化问题本质需求的一种探索性解法.它旨在刻画系统性能、可信赖性、能耗等外在表现指标与内在用户需求之间的相互联系,适用于针对可量化用户需求的优化问题建模和求解.由于回报值可以精确量化表达,因而该优化模型可以得到理论最优解.但是,大规模服务计算中的状态空间很大,同时考虑到决策行为的多样性,基于回报值的优化模型可能会遇到状态爆炸问题,需要通过状态合并、近似分析等技术手段加以克服[52].

具体地,这5种多目标模型的比较如表1所示.

表1　多目标5种模型的比较

多目标模型	适用性	解的优劣	解空间大小	求解难易
线性加权	一般	劣	中	易
基于相互关系	弱	优	中	易
ε-约束	一般	一般	小	一般
帕累托模型	强	优	大	难
基于回报值的模型	一般	优	大	一般

3.6　优化模型相互关系讨论

事实上,这几种多目标优化的模型之间不是完全孤立的,它们存在一定的转化关系.文献[53]证明了线性加权、ε-约束和帕累托3种不同优化模型在满足特定条件时可能有相交的解集.这里,我们对3种模型的相互关系进行更深入的探讨,给出更广义的结论.

具体地,文献[53]中的定理证明了线性加权模型的解对特定条件下ε-约束模型解的映射关系.而这里我们扩大讨论的范围,证明了线性加权模型和ε-约束模型在满足一定条件时的等价性.具体结论见定理1.

定理1.　对于线性加权模型和ε-约束模型,当下列条件成立时,两者是等价的.

(1)线性加权模型中式(10)成立,

$$w_i=1 \text{ 且 } w_j=0,\ \forall j\neq i \tag{10}$$

(2)ε-约束模型中,优化目标选择为 $f_i(x)$ 且满足 $\varepsilon_j=\infty,\forall j\neq i$.

同样的,文献[53]证明了在一定条件下,线性加权模型的最优解和ε-约束的最优解与帕累托模型的最优解存在交集.在这里,我们扩充了定理成立的条件,并且写出线性加权模型最优解与帕累托模型最优解交集的表达形式.

定理2.　令 Ω_i 为线性加权模型在式(10)成立下的最优解,Ω_{pareto} 为帕累托模型的最优解.那么这两种模型的最优解存在交集,即 $\Omega_i\bigcap\Omega_{\text{pareto}}\neq\varnothing$,且式(11)成立.

$$\left(\bigcup_{i=1}^{k}\Omega_i\right)\bigcap\Omega_{\text{pareto}}=\{x\mid\exists 1\leq i\leq k,x\in\Omega_{\text{pareto}}\wedge f_i(x)=\min_{x'\in S}\{f_i(x')\}\} \tag{11}$$

定理3.　令 Ω_{ε_i} 表示ε-约束模型在定理1中条件2满足下的最优解,那么它与帕累托模型的最优解 Ω_{pareto} 交集非空,即式(12)成立.

$$\Omega_{\varepsilon_i}\bigcap\Omega_{\text{pareto}}\neq\varnothing,\ \forall 1\leq i\leq k \tag{12}$$

上述3个定理的证明见附录1.

在对多个目标进行优化时,这几种优化模型也可以结合起来使用.例如,先采用帕累托模型求得帕累托最优解集合,然后将该集合作为问题的解空间,可采用线性加权模型或者ε-约束模型进一步求得该解集下的最优解.文献[36]采用帕累托模型与线性加权模型结合的方式解决服务计算系统中的服务组合问题.该文献首先采用帕累托模型得到每种服务的帕累托候选服务集合,接着采用线性加权模型,求得这些帕累托候选服务集合的组合最优解.

4　多目标优化求解方法

在第3节中介绍了服务计算多目标优化的模

型,它们以不同的方式、从不同的角度刻画优化中的多个目标,在本节中将总结服务计算多目标优化模型对应的求解方法.

4.1 转化成单目标求解

多目标优化模型中线性拟合、基于相互关系、ε-约束这 3 种模型或将多个目标全部按照权重加在一起,或找到目标之间的相互关系,或选出一个指标作为目标其余作为约束条件,这几种模型的共性是通过一定的方式降低优化中多目标的维数,将多目标优化问题转化成单目标优化问题,之后采用单目标优化的方法进行求解.

单目标优化在近几十年来的研究和发展中已有比较成熟的体系[3].根据是否能够得到最优解,可以将单目标优化的算法分为最优算法和启发式算法.

4.1.1 最优算法

穷举法可以保证能取得最优解,是一种简单直观的最优算法.它通过枚举的方式列出优化问题所有的解,求解的复杂度依赖于解空间的大小.当解空间比较小时,穷举法能够较快地找到最优值.而当解空间增大时,穷举法由于要遍历所有的解而导致复杂度太高,效率太低.尤其在组合优化问题中,解空间随着变量个数增加以指数速度增长.以服务组合问题为例,假设一个复杂服务由 n 个子服务组成,而每一个子服务又有 m 个候选项,那么所有可能的解有 m^n 个,在这样的情况下,使用穷举法无法高效地找到最优解[36,39].穷举法使用简单,当问题规模比较小时,求解方便.但当问题的规模增大到一定程度时,穷举法效率太低,需要寻求更有效的方法.

动态规划是求解最优化问题一类经典的算法,它通过组合子问题的解而解决整个问题[4].能采用动态规划算法求解的问题一般要具备两个关键因素:最优子结构和重叠子问题.最优子结构的性质是指,问题的最优解包含了其子问题的最优解,那么即可通过子问题的逐步递归求得问题最优解.重叠子问题是指采用动态规划算法自顶向下解决问题时,每次产生的子问题并不总是新问题,而有些子问题被反复计算多次,那么这样就可以采用标记等方法保存子问题的值,从而避免重复计算,降低算法的复杂度.动态规划算法能解决的经典问题包括背包问题、装配线调度问题等.如文献[54]研究服务组合的问题,它采用线性拟合的方法将收益和开销作为目标,服务质量如响应时间、可靠性作为约束条件.接着将问题抽象为背包问题,每个物品代表了候选服务,收益代表服务的效用,重量代表了服务质量属性,而端到端的约束条件代表了背包的容量.该文章进一步采用动态规划的方法来求解该模型.动态规划算法的运行时间取决于子问题的总个数和每个子问题中选择的个数,是这两者的乘积.相比于穷举法,动态规划算法的效率能够大大提高,但它的运行时间是伪多项式时间.此外,动态规划方法能够使用需要满足两个条件,即最优子结构和重叠子问题.对于不满足这两个条件的问题,动态规划方法无法高效使用.

图算法也常被使用来解决优化问题.一些研究将服务组合的问题抽象成图论问题,并采用图算法求解[37,55].以文献[37]为例,作者用带约束的最短路径问题来模拟服务组合问题,并用 Constrained Bellman-Ford(CBF)算法求解.该文章建立了对应的有向图,其中图节点代表了候选服务,QoS 参数则由边来代表.最优路径通过遍历图的边求得,要求满足所有的约束条件并具有最好的效用,它对应原服务组合问题的最优解.相较于其他方法,图算法更加直观、简洁.但某些情况下,使用图算法之前常常要先将原问题进行转化.此外,使用图算法往往需要获取服务部署的拓扑信息,在一些情况下甚至需要全局拓扑,优化前的信息获取可能耗费大量成本.

服务计算中的一些优化问题如线性规划、二次规划等,可以归结为凸优化问题.凸优化问题是这样的一类问题,它们的约束条件为凸集,而目标是最小化一个凸函数或者最大化一个凹函数[3].优化问题的凸性往往决定了问题是易解的,因为凸优化问题的局部最优解即为全局最优解.对于无约束的凸优化问题,只需通过对目标函数微分的方法找到一阶微分为 0 的解,即为最优解.对于目标函数不可微的情况,可以使用次梯度法.次梯度法只需很少的存储需求,可以通过迭代的方式求得凸优化的最优解.以文献[23]为例,它研究服务计算中的任务调度和资源分配问题,优化的目标包括最小化响应时间和最小化碳氧化合物.该文采用线性拟合的方式将这两维目标综合起来,并抽象成一个最优化模型,目标是最小化一个凸函数,接着采用次梯度的方法求解该模型.凸优化方法目前已有比较体系的研究,但能用凸优化方法解决的问题需要保持凸性.而对于不具备凸性的问题,往往难以直接应用凸优化方法进行求解.

最优算法能够保证求得结果的最优性,当问题比较简单时,它往往只需要多项式的运行时间,效率高而且准确.但当问题的难度逐渐增大时,最优算法

很难高效地求出问题的最优解. 比如 NP-难问题，就目前而言它们不存在多项式时间内的最优算法[56]. 如采用穷举法解决旅行商(TSP)问题，当城市数达到 31 时，需要约 325 年才能枚举所有解并求出最优解[57]. 使用动态规划算法求解背包问题虽然能提高效率，但它的运行时间是伪多项式时间[4]. 尤其面对人们对响应时间比较敏感的实时业务时，求解时间过长会降低系统的效率，也影响用户的满意度. 因而启发式算法应运而生，它相对于最优算法，可以定义为，一个基于直观或经验构造的算法，在可接受的花费(如计算时间)下给出待解决优化问题的可行解，该可行解与最优解的偏离程度事先不一定可以预计[57].

4.1.2　启发式算法

贪心算法是求解最优化问题的一类有效的启发式算法[4]，它常用于解决资源分配、任务调度问题，在服务组合问题中也有一定的应用[58-59]. 贪心算法通过做出一系列的选择来寻找问题的最优解. 对于其中任何一个决策点，贪心算法总是做出能产生当前最佳解的选择，也即达到局部最优解，进而期望能够通过局部最优解得到全局最优解. 但贪心算法并不总能找到最优解. 以装箱问题中降序首次适应(First Fit Decreasing)算法为例，它是一种经典的贪心算法，每次找到当前第一个能放的箱子，就把物品放进去该箱中，但是这种组合方式并不一定能得到最优的配置方案[60]. 贪心算法往往需要对已知的参数进行预处理，如对输入变量进行排序. 文献[58]采用贪心算法解决服务组合问题，首先将候选服务按照比较函数进行降序排序，接着寻找当前最优的候选服务组合. 贪心算法效率较高，但无法保证得到最优解. 它常常要对参数进行预处理，增加了算法的运行时间. 此外，贪心算法的使用常需具体两个条件，即贪心选择性质和最优子结构.

在实际的系统和工程应用中，鉴于问题的复杂性、约束性、建模困难性和求解困难性等，相对于用复杂的方法求得最优解，很多时候人们更倾向于用快速的算法求得次优解. 因此一些适用于大规模系统且具有智能特征的算法受到学术界和工业界的关注. 自 20 世纪 80 年代以来，很多新颖的智能算法如蚁群算法、模拟退火算法、遗传算法等，为解决这类难题提供了有利的工具[5,10,12-13]. 这些算法模拟自然现象和过程，构造起来直观且易懂[15,31,33,35]. 智能算法虽然能够极大地提高效率，降低运行时间，但解的优劣无法保证，很多参数都对最终解造成影响，例如初始解的产生，搜索结束条件的选择等，因此如何设置合理的参数也是智能优化算法面临和需解决的问题.

服务计算典型的单目标优化方法如图 3 所示.

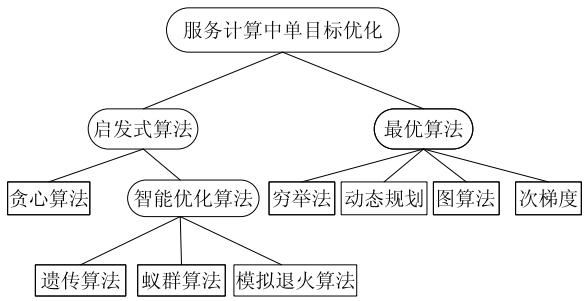

图 3　服务计算中单目标优化方法

4.2　帕累托模型求解方法

帕累托最优解是指所有未被任何其他解所优胜的解，换言之，在维持其他目标函数值不变的情况下，它们拥有某一目标函数的最优值. 帕累托最优解代表了不同目标之间的权衡，它们在多目标优化中具有举足轻重的地位[47,61]. 为了求得帕累托最优解，一种最基本的方法是基于定义，对解进行两两比较，标记比较的结果并淘汰掉被优胜的解，最后未被淘汰的解即为帕累托最优解. 这种基于两两比较的穷举法能够精确地求出帕累托最优解，但算法的效率难以保证. 尤其在解空间较大时，最坏的情况下任意两个解之间都不存在优胜关系，那么所有解都是帕累托最优解，而且都需要进行两两比较. 仍以服务组合问题为例，对于有 n 个子服务，每个子服务又有 m 个候选项的情况，解空间大小是 m^n，那么最坏情况下的算法复杂度是 $O((m^n)^2)$，对响应时间敏感的业务来说效率太低，难以满足用户对响应时间的要求.

为了提高效率，一些关注在较短时间内求得帕累托次优解的启发式算法被提出. 其中一种数学规划法通过依次考虑多目标中的某一个目标，仅仅将这个目标作为优化的对象而忽略其他的目标，以此得到该目标相关的 L 个最优解，并且计算这 L 个解在所有目标维度的值，之后得到帕累托解. 假设系统中共有 k 个目标，那么相应地执行 k 次求解，得到了 k 组最优解之后，将这些最优解综合起来即可. 以文献[9]为例，作者研究了云计算中的资源分配和任务调度问题，综合考虑了开销和响应时间两维目标，致力于最小化开销、最小化响应时间. 为求得帕累托解，作者采用了 3 个步骤. 在第 1 个步骤中，仅仅考虑开销而忽略响应时间，采用相应的算法求出开销

相关的 L 个最优解，进而计算出这 L 个最优解的开销及对应的响应时间值. 在第 2 个步骤中，仅仅考虑响应时间而忽略开销，求出响应时间相关的 L 个最优解并计算对应的两维目标的值. 在第 3 个步骤中，综合求得的两组 L 个解，并得到这 $2L$ 个解中的帕累托解. 采用这种方法可以得到在各个目标方向比较均衡的帕累托解，解的分布比较平衡. 帕累托解的优劣依赖于参数 L 的选择，当 L 较大时，能够提供较多的帕累托解候选项，解的效果较好，但这会增大算法的复杂度和运行时间；而当 L 较小时，尽管能提高效率降低运行时间，但由于选的解太少会影响帕累托解的优劣. 因此，采用这种方法在运行时间和解的优劣中存在折中.

多目标进化算法也是求解帕累托解常用的方法. 以遗传算法为例，目前已有不少基于它的多目标方法被提出，如 NSGA-II[62]、PESA[63]、SPEA2[64]等. 遗传算法模拟自然界中生物进化的过程，算法最终以非零的概率到达某些收敛点，那些点即为对应问题所求的解. 一般而言，遗传算法具有如下几个步骤：选择，交叉，变异. 选择过程对应了自然界中物种选择，交叉过程模拟了生物的交配，而变异对应于基因变化的过程. 遗传的基本单位是染色体，它代表了问题的解，因此遗传算法的目标是找到较优的染色体基因，也即找到原多目标优化问题较优的解. 染色体的编码是遗传算法中至关重要的因素，好的编码有助于提高效率，快速找到问题的帕累托解. 以服务选择问题为例，文献[1]介绍了染色体编码的两种方法，一种方法如式(13)所示.

$$染色体=[s_1 \& s_2 \& \cdots \& s_m] \tag{13}$$

其中 m 是所有服务候选项的个数，它是子服务候选项个数之和. 染色体中的每一个基因 s_i 是个 0-1 变量，代表了该候选服务是否被选中.

$$s_i=\begin{cases}1, & 第\ i\ 个候选服务被选中了\\ 0, & 第\ i\ 个候选服务未被选中\end{cases} \tag{14}$$

那么$[s_1 \& s_2 \& \cdots \& s_m]$的任意一种情况都代表了具体的某一个染色体，也即原问题的某一个解. 该文章还介绍了服务选择中染色体编码的另一种方法，如式(15).

$$染色体=[s_{11}\cdots s_{1i} \& s_{21}\cdots s_{2j} \& \cdots \& s_{n1}\cdots s_{nk}] \tag{15}$$

其中 n 是子服务的个数，i,j,k 分别代表每一类子服务中有多少个候选服务. 同样地，$s_{pq}\in\{0,1\}$ 表示具体的候选服务是否被选中. 这两种染色体编码方法的区别在于第 2 种方法将候选服务按照它们属于哪一类服务进行了分类，并且第 2 种方法中可以方便地加上限制条件(16)，

$$\sum_{j=1}^{i_l} s_{ij}=1,\ \forall i\in\{1,2,\cdots,n\} \tag{16}$$

即是对任何一类服务，它有且只有一个候选服务可以被选中，这个条件对于服务选择问题是很直观的.

在遗传算法求帕累托解的过程中，还有几个参量对于解的优劣有至关重要的影响. 一个是初始种群的规模和选择，也即初始解的规模和选择. 初始种群规模较大，选择得好，有利于快速收敛到较优解；而初始种群规模过小，则有可能陷入局部解并难以跳出. 另外，适应函数的选择也很重要，它代表了什么样的染色体是人们期望的，也即什么样的解是较优解. 适应函数的选择往往依赖于优化问题的目标和人们的偏好，如服务选择问题中目标是最小化响应时间、最大化效益，那么适应函数的一种选择可以定为该染色体能够帕累托优胜其他染色体的个数[65]. 还有遗传算法的终止规则，也即是当符合什么样的条件时，遗传算法可以结束. 一种简单的终止规则是取定迭代次数，即迭代到一定的步数就停止[5]，这样做虽然简单，但不能灵活地适应不同情况；另外一种比较好的方法是观察当前得到的最优解的情况，如果最优解的重复次数比较大，意味着算法再进化也难以改善解的性能，那么可以终止算法. 遗传算法虽然效率较高，但无法保证得到最优解，只能得到近似较优解. 且解的优劣与参数的设置有很大的关系，初始解的产生、算法结束条件的设置等都对最终的解有很大的影响.

这几种帕累托求解方法比较如表 2 所示.

表 2 帕累托模型求解方法的比较

帕累托求解方法	解的优劣	解空间	求解难易
两两比较	优	大	难
数学规划法	一般	小	易
基于进化算法	一般	大	易

面对众多的帕累托求解方法，如何对它们进行评价和比较也吸引了研究人员的关注. 以文献[35]为例，它采用超体积法(HV)来比较不同的帕累托求解方法. 超体积是指帕累托解在目标空间中能够优胜的解所组成区域的体积. 该文章进一步将所有比较算法的帕累托解集综合起来，形成一个最大的帕累托解集 $S_{\max}$，它包含了这些算法得到的所有未被其他解优胜的解. 那么算法的好坏可以用该算法帕累托解对应的超体积和所有算法的帕累托解超体积的比例 R 来表示，如式(17).

$$R=\frac{HV(S)}{HV(S_{\max})} \tag{17}$$

R 的值大代表该算法帕累托解的超体积在所有帕累托解超体积中占的比例高，那么该算法的效果是比较好的.

4.3　基于随机模型的回报值优化方法

考虑到服务过程的动态性、服务行为的多样性和服务请求的不确定性，在进行服务质量的优化时，有时需要考虑系统动态行为的随机性. 其优化步骤往往是：(1) 建立系统的随机模型，刻画其动态行为；(2) 基于随机模型进行性能分析，研究系统参数和服务质量结果之间的关系；(3) 在已有结论基础上进行优化，给出最佳的系统参数或服务策略.

马尔可夫模型(Markov Model)是性能评价常用的模型工具. 它利用马尔可夫过程(Markov Process)对系统的动态行为进行刻画. 典型的模型方法包括离散时间马尔可夫链(Discrete-Time Markov Chain，DTMC)、连续时间马尔可夫链(Continuous-Time Markov Chain，CTMC)、半马尔可夫过程(Semi-Markov Process，SMP)等.

为了方便进行服务质量的分析，在马尔可夫链中，我们可以对每个状态定义一个回报值，代表系统在利润、性能、可靠性等方面的收益，则该马尔可夫链称为是一个马尔可夫回报模型(Markov Reward Model，MRM). 和马尔可夫链类似，马尔可夫回报模型同样可以分为离散时间和连续时间两种时间维度. 回报值的引入为性能评价提供了巨大的便利. 马尔可夫回报模型可以将模型结构和系统需求紧密联系起来，是性能、可靠性等分析评价的有力工具[49]. 马尔可夫回报模型中，一般考虑稳态意义的 QoS 指标. 因此，可以首先对马尔可夫模型进行分析求解，得到每个状态 $i\in S$ 的稳态概率 π_i，随后结合每个状态的回报值 r_i，得到系统稳态回报值的形式化解析解：

$$\bar{R}=\sum_{i\in S}\pi_i\cdot r_i \tag{18}$$

这里，式(18)对离散时间和连续时间马尔可夫回报模型皆可适用. 进一步，若离散时间 MRM 中每次状态变迁的间隔时间相同，则可以证明，离散时间马尔可夫回报模型和连续时间马尔可夫回报模型是等价的[51].

通过回报值的量化表达，可知优化目标和系统参数或系统设计结构之间的对应关系，通过一些典型的优化方法即可求解最优的 QoS 指标. 这里，我们着重介绍一种基于马尔可夫回报模型的优化算法——马尔可夫决策过程(Markov Decision Process，MDP). 它是基于马尔可夫过程理论的随机动态系统的一种最优的决策过程，是运筹学中数学规划的一个重要分支.

一个基本的马尔可夫决策过程包括 5 个基本要素：(1) 状态集合 S；(2) 行为集合 A；(3) 收益函数 $r(s,a)$，其中 $s\in S$，$a\in A$；(4) 状态转移关系 S^M；(5) 优化目标函数. 这里，优化目标函数可以有两种形式. 第一是带有折扣因子的优化目标. 它借鉴经济学原理，区别对待既得利益和未来收益，将每一步的优化目标定义为当前行为的既得回报值加乘以折扣因子的未来回报期望值. 服务计算中，这类优化目标往往应用于优化服务过程中的服务收益或利润的最大化[51]. 我们用 $V_n(S(n))$ 表示第 n 次迭代过程中状态 $S(n)$ 的 MDP 收益值，$p(s'\mid S(n),a_n)$ 表示在决策行为 a_n 下状态 $S(n)$ 至状态 s' 的转移概率，$\eta\in(0,1)$ 为折扣因子. 则 MDP 的优化目标可以形式化表示为

$$V_n(S(n))=\max_{s'\in S}\left\{r(s_n,a_n)+\eta\sum_{s'\in S}p(s'\mid S(n),a_n)V_{n+1}(s')\right\} \tag{19}$$

上式也可以写成如下所示的向量形式，即

$$\boldsymbol{V}=\boldsymbol{R}+\eta\boldsymbol{P}\boldsymbol{V} \tag{20}$$

带有折扣因子的 MDP 依照时间的推移对未来所得收益进行逐步折扣，可以保证对时间累加的总收益总是收敛的[66]. 它是最常见的一类 MDP 问题，可以由经典的值迭代算法和策略迭代算法求解得到最优解[67]. 理论上可以证明，若将每个状态的收益值 $r(s,a)$ 定义为马尔可夫回报模型中的回报值 r_s，则该 MDP 求解对 MRM 模型参数优化是有效的，并且 MDP 的稳态收益值 $\boldsymbol{V}$ 和 MRM 的稳态回报值 $\bar{R}$ 具有如下对应关系[51].

$$(1-\eta)\boldsymbol{\pi}\boldsymbol{V}=\bar{R} \tag{21}$$

另一类优化目标为无折扣因子的 MDP 优化，亦被称作平均时间马尔可夫决策过程优化. 该模型不区分当前收益和未来收益，优化目标是稳态期望值的最大化. 它可用于服务计算系统可信赖性优化，因为无论何时发生系统故障，对系统的功能都是严重的影响；尤其是一些关键部件的故障，产生的后果是灾难性的[52]. 该模型的优化目标形式化表示为

$$\underset{a\in A}{\text{maximize}}\ \lim_{N\to\infty}\frac{1}{N}E\left[\sum_{n=1}^{N}r(s_n,a_n)\right] \tag{22}$$

对于该类型的 MDP 问题，可以由相关值迭代算法[67]进行求解. 文献[52]进一步给出，结合文献[68]

中的 MDP 最优性定理，可以证明由该算法得到结果的最优性.

基于随机模型的回报值优化方法以随机理论为基础，可以刻画系统动态行为，并对服务质量属性进行预测. 在模型基础上，可以清晰描述属性之间的相互关系，有针对性地进行多目标的优化. 同时，建立模型需要一些系统参数，因而增加了一定的测量开销. 马尔可夫理论需要对任务到达和服务时间的分布进行一定的假设，会损失部分分析的准确性.

确定求解方案后，便是具体的部署实施的过程. 按照是否采用集中的控制器，可以分为集中式实施和分布式实施. 一般而言，集中式方案通信代价小，但存在单点失效问题；而分布式方案效率高，但可能增加通信代价[69].

5 多目标动态优化

优化方法可从时间维度上分为静态优化和动态优化. 在静态优化中，参数是固定的，系统被当作是一个时不变的系统，系统的资源需求量和资源总量被视作与时间无关的常量. 而服务计算系统中，随着用户需求的不断提高，服务之间常以服务组合的形式共同协作，呈现出动态性和松耦合的特点，采用静态优化的方法常无法刻画系统随时间变化的性质. 相对于静态优化方法，动态优化方法能够描述系统的变化，更好地反映系统当前决策对系统时间累积的影响[66]. 动态优化理论的基本模型是马尔可夫决策过程[67]. 按照决策者的行为是否受到客观条件的影响，可将动态优化问题划分为非受限的动态优化问题和受限的动态优化问题[66].

5.1 非受限的动态优化问题

类似于单目标动态优化问题，以最小化系统成本为例，非受限的多目标动态优化问题的目标函数通常具有如下形式：

(1) 有限时间(finite horizon)累积成本函数：

$$f=E\left\{\sum_{t=0}^{T-1}f(s_t,a_t)\right\} \tag{23}$$

(2) 无穷时间(infinite horizon)累积折扣成本函数：

$$\hat{f}=E\left\{\sum_{t=0}^{\infty}\eta^t f(s_t,a_t)\right\} \tag{24}$$

(3) 无穷时间平均成本函数：

$$\bar{f}=\lim_{T\to\infty}\frac{1}{T}E\left\{\sum_{t=0}^{T-1}f(s_t,a_t)\right\} \tag{25}$$

其中，$\eta\in[0,1)$为折扣因子，一步(one-slot)成本函数 $f(s_t,a_t)$可以定义为

$$f(s_t,a_t)=\sum_{n=1}^{N}w_n f_n(s_t,a_t) \tag{26}$$

式(26)中，s_t与 a_t分别表示系统在时槽 t 起始所处的状态和采取的决策，$f_n(s_t,a_t)$表示决策者优化的第 n 个目标，$w_n\in\mathbb{R}$为分配给该目标的权值.

针对有限时间马尔可夫决策问题(23)，可运用基于动态规划的算法求解离线最优策略；针对无穷时间马尔可夫决策问题(24)与(25)，可运用值迭代(value iteration)与策略迭代(policy iteration)算法[67]，通过迭代贝尔曼方程求解最优策略. 上述算法在实际应用中主要存在两方面的问题：(1) 算法复杂度通常随着问题规模的扩大急剧增加，常陷入“维数灾难”(curse of dimensionality)的困境；(2) 算法的迭代需要外部随机变量的分布信息. 然而，在实际应用中，外部随机变量(如服务请求的到达过程)通常是非稳态的，很难用分布精确刻画其统计特征. 针对这些难题，可运用强化学习[70]或近似动态规划[71]等方法，设计在线或离线学习算法，寻求动态优化问题的最优解或次优解[72-73].

5.2 受限的动态优化问题

在受限的动态优化问题中，限制条件的定义类似于式(23)～(25). 这里重点考察限制条件为时间平均(time average)约束的多目标动态优化问题：

$$\text{minimize: } \bar{f} \tag{27}$$

$$\text{subject to: } \bar{g}_i\le 0,\ i\in\{1,\cdots,I\} \tag{28}$$

$$\bar{h}_j=0,\ j\in\{1,\cdots,J\} \tag{29}$$

$$a_t\in\boldsymbol{A}_t(s_t),\ \forall t \tag{30}$$

其中，$\boldsymbol{A}_t(s_t)$是系统处于状态 s_t时的可行决策集，目标函数 $\bar{f}$，约束函数 $\bar{g}_i$和 $\bar{h}_j$分别定义为

$$\bar{f}=\lim_{T\to\infty}\frac{1}{T}E\left\{\sum_{t=0}^{T-1}f(a_t,\omega_t)\right\} \tag{31}$$

$$\bar{g}_i=\lim_{T\to\infty}\frac{1}{T}E\left\{\sum_{t=0}^{T-1}g_i(a_t,\omega_t)\right\},\ i\in\{1,\cdots,I\} \tag{32}$$

$$\bar{h}_j=\lim_{T\to\infty}\frac{1}{T}E\left\{\sum_{t=0}^{T-1}h_j(a_t,\omega_t)\right\},\ j\in\{1,\cdots,J\} \tag{33}$$

式(31)～(33)中，ω_t表示外部随机变量在时槽 t 内的样本实现. 为便于后续表述，将时间平均函数 $f(a_t,\omega_t)$，$g_i(a_t,\omega_t)$和 $h_j(a_t,\omega_t)$分别简记为 $f(t)$，$g_i(t)$和 $h_j(t)$. 形如式(27)～(30)的多目标动态优化问题可归类为受限的马尔可夫决策问题(Constrained Markov Decision Problem，CMDP)[74]. 该类多目标动态优化问题可运用李雅普诺夫(Lyapunov)优化

技巧[75]，设计漂移成本和(Drift-Plus-Cost，DPC)最小化算法进行求解. 下面介绍在 DPC 算法的设计中几个重要的步骤.

(1)定义虚拟队列

为每个时间平均约束定义相应的虚拟队列，并利用队列稳定性理论保证满足时间平均约束. 对于不等式约束(28)和等式约束(29)，其虚拟队列可分别定义为

$$X_i(t+1)=\max[X_i(t)+g_i(t),0],\ \forall i \quad (34)$$

$$Y_j(t+1)=Y_j(t)+h_j(t),\ \forall j \quad (35)$$

定义队列向量$\boldsymbol{\Theta}(t)=[[X_i(t)]_{i\in\{1,\cdots,I\}},[Y_j(t)]_{j\in\{1,\cdots,J\}}]$表示时槽 t 起始时刻系统所有虚拟队列所处的状态.

(2)定义李雅普诺夫函数和漂移

为了度量系统的积压(拥塞)程度，定义二次李雅普诺夫函数：

$$L(\boldsymbol{\Theta}(t))=\frac{1}{2}\sum_i\alpha_i[X_i(t)]^2+\frac{1}{2}\sum_j\beta_j[Y_j(t)]^2 \quad (36)$$

其中，α_i和β_j分别表示赋予队列 $X_i(t)$和 $Y_j(t)$的权值，可实现对不同队列的区分. 在系统当前队列状态为$\boldsymbol{\Theta}(t)$的条件下，定义李雅普诺夫漂移(Lyapunov drift)为相邻时槽李雅普诺夫函数之差的条件期望：

$$\Delta(\boldsymbol{\Theta}(t))=E\{L(\boldsymbol{\Theta}(t+1))-L(\boldsymbol{\Theta}(t))\mid\boldsymbol{\Theta}(t)\} \quad (37)$$

(3)定义 DPC 函数及其上界

针对最小化问题，定义漂移成本和函数：

$$\Delta(\boldsymbol{\Theta}(t))+VE\{f(t)\mid\boldsymbol{\Theta}(t)\} \quad (38)$$

其中，可调参数 $V>0$ 用于控制漂移函数 $\Delta(\boldsymbol{\Theta}(t))$与成本函数 $E\{f(t)\mid\boldsymbol{\Theta}(t)\}$之间的平衡(tradeoff). 若 V 的取值越大，则表明决策者在系统队列稳定性和成本之间，更倾向于成本控制. 根据李雅普诺夫优化的设计原则[75]，原问题(27)～(30)可转换为求解如下解优化问题：

$$\begin{aligned}&\text{minimize } Vf(t)+\sum_{i=1}^{I}X_i(t)g_i(t)+\sum_{j=1}^{J}Y_j(t)h_j(t)\\&\text{subject to } a_t\in A_t(s_t),\ \forall t\end{aligned} \quad (39)$$

对于任意时槽 t，问题(39)是一个静态优化问题，可利用问题自身的特性，设计静态优化算法进行求解，得到最优决策 a_t^*. 然后，将 a_t^* 代入式(34)和(35)，更新虚拟队列的状态.

若原问题(27)～(30)存在可行解，且外部随机变量 ω_t关于时间 t 独立同分布，则 DPC 最小化算法能够保证虚拟队列 $X_i(t)$和 $Y_j(t)$满足稳定性条件. 此外，系统平均成本满足[75]

$$\lim_{T\to\infty}\frac{1}{T}E\left\{\sum_{t=0}^{T-1}f(t)\right\}\le\bar{f}^*+\frac{B}{V} \quad (40)$$

其中，$B>0$ 为常数，$\bar{f}^*$ 表示原问题目标函数(27)的最优值. 式(40)表明，若平衡控制参数 $V>0$ 足够大，则 DPC 最小化算法能够实现系统平均成本的最优控制.

DPC 最小化算法使决策者能够根据系统当前的状态选取决策，实现算法稳定性与最优性之间的可控平衡. 该算法的优势在于决策的选取仅依赖于系统在当前时刻所处的状态，可在线部署实现. 另外，不同于传统值/策略迭代算法，DPC 最小化算法无需迭代基于系统状态的值函数(value function)，因而能在一定程度上抑制“状态空间爆炸”. 但该算法在应用中也存在一定的局限性：(1)仅适用于求解目标函数和约束函数均为时间平均函数的马尔科夫决策问题；(2)不能解决潜在的决策空间(action space)过大的问题.

李雅普诺夫优化技巧在多目标动态优化问题中的应用已有一些研究成果[6,46,76-79]. 例如，文献[6]针对 SaaS 云平台的应用请求调度和资源分配问题，运用李雅普诺夫优化方法，设计了一种在线控制算法. 该算法通过在线接纳控制、负载均衡和虚拟机调度，实现 SaaS 云平台服务收益的最大化，同时降低系统耗电量. 文献[46]采用李雅普诺夫优化技巧解决服务计算中服务调度与管理的问题，实现了系统收益、能耗、响应时间的多目标优化. 文献[76]考察了电价和碳排放率(Carbon Emission Ratek，CER)在时间和空间尺度上的多样性，运用李雅普诺夫优化，设计在线控制算法，通过负载均衡、动态容量分配和服务器速率调节，实现地理分布式云数据中心电力成本、碳排放和服务质量 3 个优化目标之间的平衡. 文献[77]对分布式云环境下多组织的任务调度问题进行了研究，利用数据中心电价在时间和空间维度上的差异性，设计了优化系统能耗和任务调度公平性的在线调度算法 GreFar；通过控制李雅普诺夫参数，GreFar 能无限逼近问题的离线最优值，且无需外部随机变量(如任务到达、电价等)的统计信息.

6　总结与展望

服务计算的多目标优化关注在系统的设计和管理中，如何合理地进行系统配置和管理，以更好地满足用户和服务供应商的需求. 优化过程覆盖服务系统的整个生命周期，直接影响用户和服务供应商的利益，是服务计算中重要的研究问题. 本文结合服务

计算的多目标参数,介绍了优化涉及的指标体系和问题.在此基础上,总结了服务计算中5种典型的多目标优化模型,并从模型的适用性、难易程度等方面对它们进行了比较和区分.同时讨论模型的相互关系.之后,分别介绍了这些多目标优化模型对应的求解方法.我们期望本文对多目标优化的模型和方法的研究综述可以为服务计算的系统设计和改进提供一定的理论参考.

考虑到多目标优化中采用的模型和求解方法不同,得到的解也不尽相同.对不同的解进行评价,有利于找到该问题下合适的模型和求解方法,为优化策略的分析和比较提供理论依据和支撑,是一个值得关注的问题,这也是我们下一步工作的方向.此外,如何在多目标优化中根据不同目标之间的关系,建立全面系统的多目标优化理论,设计普适性的优化模型和方法,是亟待解决的研究难点,也是计算机学科中优化理论的一个发展方向.

参 考 文 献

[1] Zhang L J, Zhang J, Cai H. Services Computing. Beijing: Springer and Tsinghua University Press, 2007

[2] Li Y, Lin C. QoS-aware service composition for workflow-based data-intensive applications//Proceedings of the 2011 IEEE International Conference on Web Services (ICWS 2011). Washington, USA, 2011: 452-459

[3] Boyd S, Vandenberghe L. Convex Optimization. Cambridge, UK: Cambridge University Press, 2004

[4] Cormen T H, Leiserson C E, Rivest R L, Stein C. Introduction to Algorithms. MIT, USA: MIT Press, 2005

[5] Wada H, Champrasert P, Suzuki J, Oba K. Multiobjective optimization of SLA-aware service composition//Proceedings of the IEEE Congress on Services. Honolulu, USA, 2008: 368-375

[6] Zhou Z, Liu F, Jin H, et al. On arbitrating the power-performance tradeoff in SaaS clouds//Proceedings of the IEEE INFOCOM 2013. Turin, Italy, 2013: 872-880

[7] Leitner P, Hummer W, Satzger B, et al. Cost-efficient and application SLA-aware client side request scheduling in an infrastructure-as-a-service cloud//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD 2012). Honolulu, USA, 2012: 213-220

[8] Kong X, Lin C, Jiang Y, et al. Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. Journal of Network and Computer Applications, 2011, 34(4): 1068-1077

[9] Bessai K, Youcef S, Oulamara A, et al. Bi-criteria workflow tasks allocation and scheduling in cloud computing environments//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD 2012). Honolulu, USA, 2012: 638-645

[10] Wagner F, Klein A, Klopper B, et al. Multi-objective service composition with time- and input-dependent QoS//Proceedings of the 2012 IEEE 19th International Conference on Web Services (ICWS 2012). Honolulu, USA, 2012: 234-241

[11] Ye X, Proietti R, Yin Yawei, et al. Buffering and flow control in optical switches for high performance computing. IEEE/OSA Journal of Optical Communications and Networking, 2011, 3(8): A59-A72

[12] Chen Wei, Qiao X, Wei J, Huang T. A profit-aware virtual machine deployment optimization framework for cloud platform providers//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD 2012). Honolulu, USA, 2012: 17-24

[13] Yusoh Z I M, Tang M. Composite SaaS placement and resource optimization in cloud computing using evolutionary algorithms//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD 2012). Honolulu, USA, 2012: 590-597

[14] He S, Guo L, Ghanem M, Guo Y. Improving resource utilisation in the cloud environment using multivariate probabilistic models//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD 2012). Honolulu, USA, 2012: 574-581

[15] Jing X, Fortes J A B. Multi-objective virtual machine placement in virtualized data center environments//Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications (GreenCom 2010) & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom 2010). Hangzhou, China, 2010: 179-188

[16] Karve A, Kimbrel T, Pacifici G, et al. Dynamic placement for clustered web applications//Proceedings of the 15th International Conference on World Wide Web. Edinburgh, UK, 2006: 595-604

[17] Avizienis A, Laprie J-C, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1): 11-33

[18] Birolini A. Reliability Engineering, Theory and Practice. 5th Edition. Berlin, Heidelberg: Springer-Verlag, 2007: 2-24

[19] Madan B, Goševa-Popstojanova K, Vaidyanathan K, Trivedi K S. A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation, 2004, 56(1-4): 167-186

[20] Lin Chuang, Kong Xiang-Zhen, Zhou Huan. Enhance the dependability of computing systems: Integration of virtualization and SOA. Journal of Software, 2009, 20(7): 1986-2004(in Chinese)
(林闯, 孔祥震, 周寰. 增强计算系统可信赖性: 融合虚拟化和SOA. 软件学报, 2009, 20(7): 1986-2004)

[21] Feng W, Feng X, Ge R. Green Supercomputing Comes of Age. IT Professional, 2008, 10(1): 17-23

[22] Gao P, Curtis A, Wong B, Keshav S. It's not easy being green//Proceedings of the ACM SIGCOMM. Helsinki, Finland, 2012: 211-222

[23] Doyle J, O'Mahony D, Shorten R. Server selection for carbon emission control//Proceedings of the 2nd ACM SIGCOMM Workshop on Green Networking. Toronto, Canada, 2011: 1-6

[24] Le Kien, Bianchini R, Nguyen T D, et al. Capping the brown energy consumption of Internet services at low cost// Proceedings of the 2010 International Conference on Green Computing. Chicago, USA, 2010: 3-14

[25] Chen Y, Zhang P, Kong X, Lin C. Reliability-aware energy efficiency in Web service provision and placement//Proceedings of the 2013 IEEE 20th International Conference on Web Services (ICWS 2013). Santa Clara, USA, 2013: 411-418

[26] Garfinkel T, Warfield A. What virtualization can do for security? The USENIX Magazine, 2007, 32(6): 28-34

[27] Xu J, Fortes J. A multi-objective approach to virtual machine management in datacenters//Proceedings of the 8th ACM International Conference on Autonomic Computing. New York, USA, 2011: 225-234

[28] Rykov V V. Monotone control of queueing systems with heterogeneous servers. Queueing Systems, 2001, 37(4): 391-403

[29] Casavant T L, Kuhl J G. A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering, 1988, 14(2): 141-154

[30] Strunk A. QoS-aware service composition: A survey// Proceedings of the 2010 IEEE 8th European Conference on Web Services (ECOWS 2010). Ayia Napa, Cyprus, 2010: 67-74

[31] Wang J, Hou Y. Optimal Web service selection based on multi-objective genetic algorithm//Proceedings of the International Symposium on Computational Intelligence and Design. Wuhan, China, 2008: 553-556

[32] Chang W, Wu C, Chang C. Optimizing dynamic Web service component composition by using evolutionary algorithms// Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence. Compiegne, France, 2005: 708-711

[33] Claro D, Albers P, Hao J. Selecting Web services for optimal composition//Proceedings of the 2nd International Workshop on Semantic and Dynamic Web Processes (SDWP 2005). Orlando, USA, 2005: 32-45

[34] Wiesemann W, Hochreiter R, Kuhn D. A stochastic programming approach for QoS-aware service composition// Proceedings of the 2008 IEEE 8th International Symposium on Cluster Computing and the Grid (CCGRID'08). Lyon, France, 2008: 226-233

[35] Wagner F, Honiden S, Ishikawa F. Towards robust service compositions in the context of functionally diverse services// Proceedings of the 21st International Conference on World Wide Web. New York, USA, 2012: 969-978

[36] Alrifai M, Skoutas D, Risse T. Selecting skyline services for QoS-based Web service composition//Proceedings of the 19th International Conference on World Wide Web. North Carolina, USA, 2010: 11-20

[37] Barakat L, Miles S, Poernomo I, Luck M. Efficient multi-granularity service composition//Proceedings of the 2011 IEEE International Conference on Web Services (ICWS 2011). Washington, USA, 2011: 227-234

[38] Yu T, Zhang Y, Lin K. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on the Web, 2007, 1(1): 6-32

[39] Qi L, Tang Y, Dou W, Chen J. Combining local optimization and enumeration for QoS-aware Web service composition// Proceedings of the 2010 IEEE International Conference on Web Services (ICWS 2010). Miami, USA, 2010: 34-41

[40] Skoutas D, Sacharidis D, Simitsis A, Sellis T. Ranking and clustering Web services using multicriteria dominance relationships. IEEE Transactions on Services Computing, 2010, 3(3): 163-177

[41] Peterson L L, Davie B S. Computer Networks: A Systems Approach. Netherlands: Elsevier, 2007

[42] Huang J, Chen Y, Lin C, Chen J. Ranking Web services with limited and noisy information//Proceedings of the 21th IEEE International Conference on Web Services (ICWS 2014). Alaska, USA, 2014, to be published

[43] Haimes Y Y, Ladson L S, Wismer D A. Bicriterion formulation of problems of integrated system identification and system optimization. IEEE Transactions on Systems Man and Cybernetics, 1971, SMC-1(3): 296

[44] Tordsson J, Montero R, Moreno-Vozmediano R, Llorente Ignacio M. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems, 2012, 28(2): 358-367

[45] Breitgand D, Marashini A, Tordsson J. Policy-driven service placement optimization in federated clouds. IBM Research Division, New York, USA: Technical Report: H-0299, 2011

[46] Chen Y, Huang J, Xiang X, Lin C. Energy efficient dynamic service selection for large-scale Web service system// Proceedings of the 21th IEEE International Conference on Web Services (ICWS 2014). Alaska, USA, 2014, to be published

[47] Vira C, Haimes Y Y. Multiobjective Decision Making: Theory and Methodology. Amsterdam, Holland: North-Holland, 1983

[48] Chen Ying, Huang J, Lin C. Partial selection: An efficient approach for QoS-aware Web service composition//Proceedings of the 21th IEEE International Conference on Web Services (ICWS 2014). Alaska, USA, 2014, to be published

[49] Bolch G, Greiner S, Meer H, Trivedi K. Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Hoboken, New Jersey, USA: John Wiley & Sons, 2006

[50] Huang J, Chen Y, Lin C, Chen J. Ranking Web services with limited and noisy information//Proceedings of the 21th IEEE International Conference on Web Services (ICWS 2014). Alaska, USA, 2014, to be published

[51] Huang J, Lin C. Improving energy efficiency in Web services: An agent-based approach for service selection and dynamic speed scaling. International Journal of Web Services

Research, 2013, 10(1): 29-52

[52] Huang J, Lin C, Wan J. Modeling, analysis and optimization of dependability-aware energy efficiency in services computing systems//Proceedings of the 10th IEEE International Conference on Services Computing (SCC 2013). Santa Clara, USA, 2013: 683-690

[53] Miettinen K. Nonlinear Multiobjective Optimization. Massachusetts, USA: Kluwer Academic Publishers, 1999

[54] Yu Tao, Lin Kwei-Jay. Service selection algorithms for Web services with end-to-end QoS constraints//Proceedings of the 2004 IEEE International Conference on e-Commerce Technology (CEC 2004). California, USA, 2004: 129-136

[55] Gao Y, Zhang B, Na J, et al. Optimal selection of Web services for composition based on interface-matching and weighted multistage graph//Proceedings of the 2005 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005). Dalian, China, 2005: 336-338

[56] Hochbaum D S. Approximation algorithms for NP-hard problems. ACM SIGACT News, 1997, 28(2): 40-52

[57] Xing Wen-Xun, Xie Jin-Xing. Modern Optimization Methods. Beijing: Tsinghua University Press, 2013(in Chinese)
(邢文训, 谢金星. 现代优化计算方法. 北京: 清华大学出版社, 2013)

[58] Thomas W, Bleul S, Comes D, Geihs K. Different approaches to semantic Web service composition//Proceedings of the 2008 3rd International Conference on Internet and Web Applications and Services (ICIW'08). Athens, Greece, 2008: 90-96

[59] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition//Proceedings of the 18th International Conference on World Wide Web. Madrid, Spain, 2009: 881-890

[60] Johnsonf D S, Ullman J D, Gareyi M R, Grahamii R L. Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM Journal on Computing, 1974, 3(4): 299-325

[61] Steuer R E. Multiple Criteria Optimization: Theory, Computation, and Application. New York: John Wiley, 1989

[62] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197

[63] Corne D, Knowles J, Oates M. The Pareto envelope-based selection algorithm for multiobjective optimization//Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI). Paris, France, 2000: 839-848

[64] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization//Proceedings of the Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems Conference (EUROGEN 2001). Athens, Greece, 2001: 95-100

[65] Ma H, Bastani F, Yen I-Ling, Mei H. QoS-driven service composition with reconfigurable services. IEEE Transactions on Services Computing, 2013, 6(2): 20-34

[66] Lin Chuang, Wan Jian-Xiong, Xiang Xu-Dong, Meng Kun, Wang Yuan-Zhuo. Dynamic optimization in computer systems and computer networks: Models, solutions, and applications. Chinese Journal of Computers, 2012, 35(7): 1339-1357(in Chinese)
(林闯, 万剑雄, 向旭东, 孟坤, 王元卓. 计算机系统与计算机网络中的动态优化: 模型、求解与应用. 计算机学报, 2012, 35(7): 1339-1357)

[67] Puterman M. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New Jersey, USA: John Wiley & Sons, 1994

[68] Derman C. Denumerable state Markovian decision processes-average cost criterion. The Annals of Mathematical Statistics, 1966, 37(6): 1545-1553

[69] Lin Chuang, Li Yin, Wan Jian-Xiong. Optimization approaches for QoS in computer networks: A survey. Chinese Journal of Computers, 2011, 34(1): 1-14(in Chinese)
(林闯, 李寅, 万剑雄. 计算机网络服务质量优化方法研究综述. 计算机学报, 2011, 34(1): 1-14)

[70] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge, Massachusetts: A Bradford Book, 1998

[71] Powell W B. Approximate Dynamic Programming: Solving the Curses of Dimensionality. 2nd Edition. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2011

[72] Xiang X, Wan J, Lin C, Chen X. A dynamic programming approximation for downlink channel allocation in cognitive femtocell networks. Computer Networks, 2013, 57(15): 2976-2991

[73] Xiang X, Lin C, Chen X. Energy-efficient link selection and transmission scheduling in mobile cloud computing. IEEE Wireless Communications Letters, 2014, 3(2): 153-156

[74] Altman E. Constrained Markov Decision Processes. Boca Raton, FL, USA: Chapman and Hall/CRC Press, 1999

[75] Neely M J. Stochastic Network Optimization with Application to Communication and Queueing Systems. California, USA: Morgan & Claypool, 2010

[76] Zhou Z, Liu F, Xu Yong, et al. Carbon-aware load balancing for geo-distributed cloud services//Proceedings of the 2013 IEEE 21st International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS 2013). San Francisco, USA, 2013: 232-241

[77] Ren S, He Y, Xu F. Provably-efficient job scheduling for energy and fairness in geographically distributed data centers//Proceedings of the IEEE International Conference on Distributed Computing Systems. Macau, China, 2012: 22-31

[78] Xiang X, Lin C, Chen X. EcoPlan: Energy-efficient downlink and uplink data transmission in mobile cloud computing. Springer Wireless Networks, DOI: 10.1007/s11276-014-0795-x, 2014, in publication

[79] Xiang X, Lin C, Chen X, Shen X. Toward optimal admission control and resource allocation for LTE-A femtocell uplink. IEEE Transactions on Vehicular Technology, DOI: 10.1109/TVT.2014.2351837, 2014, in publication

附录 1.

定理 1 证明.

当定理 1 中条件成立时，线性加权模型目标函数转化为

$$\sum_{i=1}^{k} w_i f_i(x) = w_i f_i(x) + \sum_{j \neq i} w_j f_j(x) = w_i f_i(x) = f_i(x).$$

因此模型转化为

$$\underset{x}{\text{minimize}}\ f_i(x)$$
$$\text{suject to}\ x \in S$$

而 ε-约束模型转化为

$$\underset{x}{\text{minimize}}\ f_i(x)$$
$$\text{suject to}\ f_j(x) \leq \varepsilon_j = \infty,\ \forall j = 1, \cdots, k,\ j \neq i$$
$$x \in S$$

那么此时线性加权模型和 ε-约束模型是等价的. 证毕.

定理 2 证明.

对于任意的 i，当式(10)成立时，Ω_i 包含了解空间中所有第 i 维目标函数为最优的解集，即

$$\Omega_i = \{x \mid f_i(x) = \min_{x' \in S}\{f_i(x')\}\} \tag{41}$$

下面用反证法证明帕累托解集 Ω_{pareto} 中存在某个解 x_p，使得 $x_p \in \Omega_i$.

假设 $\Omega_i \cap \Omega_{\text{pareto}} = \varnothing$，那么任取 $x' \in \Omega_i$，有 $x' \notin \Omega_{\text{pareto}}$. 因而必定 $\exists x_p \in \Omega_{\text{pareto}}$，使得 $x_p \succ x'$. 则根据帕累托的定义，有 $f_i(x_p) \leq f_i(x')$. 又根据 Ω_i 的定义，有 $f_i(x') = \min_{x \in S}\{f_i(x)\}$，因此 $f_i(x') \leq f_i(x_p)$. 那么有且只有一种情况成立，即 $f_i(x_p) = f_i(x')$. 由 Ω_i 的定义，有 $x_p \in \Omega_i$，即 $x_p \in \Omega_i \cap \Omega_{\text{pareto}} \neq \varnothing$，与假设矛盾.

因此可得，$\Omega_i \cap \Omega_{\text{pareto}} \neq \varnothing$，由式(41)，可得

$$\Omega_i \cap \Omega_{\text{pareto}} = \{x \mid x \in \Omega_{\text{pareto}} \wedge f_i(x) = \min_{x' \in S}\{f_i(x')\}\}$$

由集合的分配律，可得

$$\begin{aligned}(\bigcup_{i=1}^{k} \Omega_i) \cap \Omega_{\text{pareto}} &= \bigcup_{i=1}^{k} (\Omega_i \cap \Omega_{\text{pareto}}) \\ &= \bigcup_{i=1}^{k} \{x \mid x \in \Omega_{\text{pareto}} \wedge f_i(x) = \min_{x' \in S}\{f_i(x')\}\} \\ &= \{x \mid \exists 1 \leq i \leq k, x \in \Omega_{\text{pareto}} \wedge f_i(x) = \min_{x' \in S}\{f_i(x')\}\}\end{aligned}$$

即式(11)成立. 证毕.

定理 3 证明.

定理 1 中已经证明，在满足定理 1 的条件下，ε-约束模型与线性加权模型是等价的，因此 $\Omega_{\varepsilon_i} = \Omega_i$，$\forall 1 \leq i \leq k$. 又定理 2 中证明了在同样的条件下，$\Omega_i \cap \Omega_{\text{pareto}} \neq \varnothing$，$\forall 1 \leq i \leq k$ 成立，那么 $\Omega_{\varepsilon_i} \cap \Omega_{\text{pareto}} \neq \varnothing$，$\forall 1 \leq i \leq k$ 成立. 证毕.

LIN Chuang, born in 1948, Ph. D., professor, Ph.D. supervisor. His research interests include computer networks, performance evaluation, network security analysis, and Petri net theory.

CHEN Ying, born in 1990, Ph. D. candidate. Her research interests includes performance evaluation and optimization.

HUANG Ji-Wei, born in 1987, Ph.D., assistant professor. His research interests include services computing, performance modeling, evaluation and optimization.

XIANG Xu-Dong, born in 1986, Ph. D. candidate. His research interests include performance evaluation and optimal control.

Background

Services computing is a new computing paradigm which is widely used in many fields. The multi-objective optimization problems in services computing have received much attention from academia and industry. This paper surveys multi-attribute metrics in services computing and introduces five important models for multi-objective optimization. In addition, their corresponding solutions are presented and compared.

This work is partly supported by the National Natural Science Foundation of China (Nos. 61020106002, 61472199) and the Tsinghua University Initiative Scientific Research Program (No. 20121087999). These projects aim to provide better principles for the design and management of services computing systems. Our group has been working on the performance evaluation and optimization of services computing systems for years. Several research papers have been published on respectable journals and conferences, such as IEEE Transactions on Services Computing, IEEE Transactions on Computers, IEEE INFOCOM and ICWS. This paper summarizes the research processes of the multi-objective optimization methods in recent years and prospect future research challenges.