



浙江工业大学

# 硕士学位论文

论文题目：基于多目标优化的云服务组合与调度研究

作者姓名 王烨坤

指导教师 张美玉 简琤峰

学科专业 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2015 年 4 月 22 日

浙江工业大学硕士学位论文

基于多目标优化的云服务组合与调度研究

作者姓名：王烨坤

指导教师：张美玉 简琤峰

浙江工业大学计算机科学与技术学院

2015 年 4 月

**Dissertation Submitted to Zhejiang University of Technology  
for the Degree of Master**

**RESEARCH ON CLOUD SERVICE COMPOSITION AND  
SCHEDULING FOR MULTI-OBJECTIVE OPTIMIZATION**

**Candidate: Wang-yekun**

**Advisor: Zhang-meiyu JIAN-Chengfeng**

**College of Computer Science and Technology  
Zhejiang University of Technology  
Apr 2015**

# 浙江工业大学

## 学位论文原创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的研究成果。除文中已经加以标注引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的研究成果，也不含为获得浙江工业大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名：

日期： 年 月 日

## 学位论文版权使用授权书

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

本学位论文属于

1、保密□，在\_\_\_\_\_年解密后适用本授权书。

2、不保密□。

（请在以上相应方框内打“√”）

作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

# 基于多目标优化的云服务组合与调度研究

## 摘 要

随着移动互联网的普及及其应用的快速发展,传统的单一功能的云服务已经很难满足复杂的应用需求,所以能满足多功能需求的云服务组合越来越受到关注。针对多种需求的云服务的管理不仅需要解决有限的云服务资源的使用需求,同时还要考虑优化的调度策略以提高云服务请求执行的效率。因此满足多种需求的云服务组合与调度的多目标优化研究具有一定的理论意义和实际应用价值。

论文主要做了以下几方面的研究:

(1) 对云服务的服务质量属性进行了描述,并给出相应的数学表达式,并在此基础上提出了基于全局 QoS 评价与局部 QoS 评价相结合的最优云服务组合的选择模型。

(2) 建立了面向批量服务请求的云服务调度方法。描述了两阶段优化的云服务组合与调度的体系结构与流程,确定了多目标优化的约束规则。

(3) 在传统的粒子群优化算法的基础上,引入了动态因子,二阶振荡机制以及杂交策略三种改进方式,提出了改进的粒子群优化算法 (Modified Particle Swarms Optimization, MPSO), 求解本文提出的多目标优化问题。

(4) 最后,论文进行了不同问题规模下的云服务组合与调度的仿真实验。实验结果证明相对于传统方法,在本文提出的多目标优化问题的情况下,基于 MPSO 的云服务组合与调度的方法能在一定程度上提高解的质量,且算法运算效率较高。同时,本文针对云服务调度模型与算法做了实例仿真,证明了其具有一定的实际应用价值。

**关键词:** 服务组合, 改进粒子群算法, 多目标, 优化调度

# **RESEARCH ON CLOUD SERVICE COMPOSITION AND SCHEDULING FOR MULTI-OBJECTIVE OPTIMIZATION**

## **ABSTRACT**

With the further development of the mobile Internet and mobile applications, cloud service with single function can't meet the requirement of large-scale and complex application demands. So cloud services composition to meet needs of multi-function are attracting attention. When cloud service management platform dealing with the demands of the batch service requests for using limited cloud services, it also needs to optimize the scheduling strategies and improve the efficiency of the cloud service requests execution. So the research on cloud service composition and scheduling has certain theoretical significance and practical application value.

This paper mainly finished the following several aspects of the research:

- (1) The service quality attributes of cloud services were described, and the mathematical expressions were given. Then, the optimal model about cloud service composition selection was put forward, which was based on global QoS evaluation combined with local QoS evaluation.
- (2) A cloud service scheduling model for batch service requests was established. On the basis of the scheduling model, the structure and process of cloud service composition and scheduling based on two stage optimization were described in this paper. At the same time, we determined the constraint rules of multi-objective optimization.
- (3) The dynamic acceleration parameters, hybrid strategy and two-order oscillating equation were the popular way to modify Particle Swarms Optimization(PSO), which were introduced to improve traditional particle swarm optimization algorithm in this paper. The Modified Particle Swarms Optimization(MPSO) algorithm was used to solve this multi-objective problem proposed in this paper.
- (4) Finally, this article has carried on simulation experiment about the cloud service composition and scheduling problems with different problem scales. Experimental results showed that compared with other traditional methods, the proposed method based on MPSO to

solve cloud service composition and scheduling problems proposed in this paper can improve the quality of the solution to a certain extent. The computation efficiency of algorithm proposed is high. we also did the instance simulation for cloud service scheduling model and algorithm proposed in this paper, which was proved that it had certain actual application value.

**Key Words:** cloud service composition, MPSO, Multi-Objective, optimal scheduling

## 目 录

摘    要.....	i
第 1 章 绪    论.....	- 1 -
1.1 研究背景与意义.....	- 1 -
1.2 国内外研究现状.....	- 3 -
1.2.1 最优云服务组合选择.....	- 3 -
1.2.2 云计算平台的调度体系与算法.....	- 4 -
1.3 存在的问题与拟解决方法.....	- 6 -
1.4 本文的研究内容.....	- 7 -
1.5 论文组织结构.....	- 9 -
1.6 本章小结.....	- 9 -
第 2 章 相关研究与关键技术.....	- 10 -
2.1 云计算与云服务的相关介绍.....	- 10 -
2.1.1 云计算的发展与应用.....	- 10 -
2.1.2 云服务及其标准.....	- 10 -
2.2 云服务组合与调度相关研究.....	- 11 -
2.2.1 基于 QoS 的云服务组合研究.....	- 11 -
2.2.2 云服务资源调度研究.....	- 14 -
2.3 本章小结.....	- 15 -
第 3 章 两阶段优化的云服务组合与调度.....	- 16 -
3.1 引言.....	- 16 -
3.2 基于 QoS 的最优云服务组合选择.....	- 16 -
3.2.1 云服务组合结构体系与实现过程.....	- 16 -
3.2.2 基于 QoS 的云服务评价机制.....	- 19 -
3.2.3 全局 QoS 与局部 QoS 相结合的云服务组合模型.....	- 22 -
3.3 面向批量服务请求的调度体系与模型.....	- 23 -
3.3.1 调度体系.....	- 23 -
3.3.2 调度模型.....	- 25 -
3.4 基于两阶段的云服务组合与调度的优化.....	- 27 -
3.4.1 云服务组合与调度体系结构与流程.....	- 27 -
3.4.2 多目标优化.....	- 29 -
3.5 本章小结.....	- 29 -



<b>第 4 章 基于改进粒子群算法的云服务组合与调度.....</b>	<b>- 31 -</b>
4.1 引言.....	- 31 -
4.2 基础粒子群算法简介.....	- 31 -
4.3 粒子群优化算法的改进.....	- 33 -
4.4 应用于两阶段优化的改进粒子群算法.....	- 36 -
4.4.1 定义粒子的含义.....	- 36 -
4.4.2 适应度函数.....	- 38 -
4.5 本章小结.....	- 40 -
<b>第 5 章 实验仿真及结果分析.....</b>	<b>- 41 -</b>
5.1 引言.....	- 41 -
5.2 基于 QoS 的云服务组合问题仿真.....	- 41 -
5.3 面向批量服务请求的云服务调度仿真.....	- 45 -
5.4 云服务调度实例仿真.....	- 48 -
5.5 算法时间分析.....	- 51 -
5.6 本章小结.....	- 53 -
<b>第 6 章 总结与展望.....</b>	<b>- 54 -</b>
6.1 总结.....	- 54 -
6.2 下一步展望.....	- 54 -
<b>参考文献.....</b>	<b>- 56 -</b>
<b>致谢.....</b>	<b>- 61 -</b>
<b>攻读学位期间的科研项目和学术论文.....</b>	<b>- 62 -</b>

# 第 1 章 绪 论

## 1.1 研究背景与意义

进入到当今的信息化时代,移动云计算等领域无疑已经成为各大公司的探索焦点。云计算<sup>[1]</sup>是面向服务计算(SOC)的新型网络计算模型,它的一大特点便是服务资源的按需付费。云计算平台可以根据所拥有的服务资源,向互联网用户提供其所需的服务,同时广大的云平台资源供应商根据客户需求提供自身的计算资源参与到云平台的活动,并从中收取费用。云计算平台所提供的服务的主要形式是以 web 服务为载体的云服务以及云制造产业中提供的制造云服务。云平台提供服务的方式可能是提供某种计算资源或者是云服务供应商操作资源为用户满足某些需求。提出云服务请求的用户,云服务的提供者,云服务管理平台以及相互间数据传输组成了一个整体的服务框架。随着云计算的广泛发展以及云制造等新型应用的出现,云环境下服务请求的执行正向着智能化,自治性的方向发展。

云平台上提供的服务分为两种,一种是具有单一功能的细粒度原子服务,还有一种是复合型的多功能粗粒度服务组合。而随着电商以及云主机等新兴互联网业务的出现,用户提出的服务请求越来越复杂,并且具有多功能需求的特性,所以用户提出的云服务请求依靠单一功能的云服务的情况越来越少。云环境下用户所提交的服务请求的大多数都具有多功能性的服务需求,单一的云服务不能满足这些较大规模且复杂的服务请求的执行,此时就需要组合多种云服务以满足这样的需求。云平台上的各种资源所提供的服务按照一定的顺序或者规则连接起来形成服务组合,即云服务组合<sup>[2]</sup>。随着云计算的广泛发展,云平台上的服务组合技术在云制造等新兴云计算应用产业方面也发挥着巨大的作用。稳定高效的一组云服务组合成为了云平台完成各项服务请求执行的关键。要使这样提供服务的思路成为现实,必须有多个服务层次提供技术支持,包括了分布式的数据中心为互联网用户以及云服务资源供应商提供数据存储与读取,以及硬件服务层为云服务提供内存, CPU 等硬件支持等。而当今云计算技术水平已经发展到了较高水平,特别是分布式的服务资源的整合和供给,这为云服务的组合提供了很大的便利。

作为使用云服务的互联网用户,对云服务以及云服务组合的要求主要体现在经济和性能这两个主要的方面。由于网络环境具有动态不确定性的特点,所以云服务组合的相关研究较多地会考虑动态性因素,当然除此之外,主要需要考虑的是如何满足用户对服务质量方面的要求。基于满足用户对服务质量要求的服务组合选择问题是当前云服务相关研究中

被探讨的最为广泛的问题之一。服务质量的属性分类主要分为功能性相关的属性和非功能性相关的属性。比如功能性属性包括响应时间, 费用, 负载度等等, 非功能性属性包括可靠性, 可用性, 安全性等因素。QoS(Quality of Service)<sup>[3]</sup>是云服务和云服务组合的一个极为重要的综合评价指标, 在面向服务的云平台体系中以度量值的形式体现了服务或者服务组合的优劣程度。随着云计算技术的快速发展, 大量功能相同或相似的云服务不断出现, 但是功能相同的云服务的服务质量的属性值可能不同。如何组合不同的云服务, 使得云服务组合的整体服务质量属性值之和达到最高, 甚至在该前提下, 满足用户对某些局部服务质量属性的要求, 是云服务组合优化研究领域里的一个重要研究问题。云环境下互联网用户提交的服务请求的执行质量很大程度上取决于所选择的云服务组合的质量, 基于 QoS 来选择云服务, 有助于评估所选服务组合的质量。同时服务选择的过程中, 还需要满足用户对局部某些服务质量属性的特别要求。所以在为云服务请求选择最优的服务组合过程中, 需要对多个目标进行优化。

云服务具有时效性和重用性, 并且多个服务请求同时请求需要的服务可能会因为服务资源有限造成冲突, 所以在批量的服务请求在执行过程中必须减少由此带来的等待时间。因此云服务平台中需要对批量的服务请求做出响应的时候, 更需要对服务资源做出最优化的调度方案<sup>[4]</sup>。云环境下批量服务请求的执行除了需要高效的云服务调度策略提高执行的效率, 同时要求该调度策略还必须要统筹兼顾多个服务请求的 QoS 约束。在云平台可以完成资源分布广, 实时协同性强的规模生产和计算任务, 而云服务平台管理系统主要进行服务资源调度分配, 实时监控, 以及对服务之间的数据交换进行统一的管理等。云平台协调服务请求执行过程时, 如何进行服务资源的动态分配和调度, 将直接影响互联网用户提交的批量服务请求执行的时间和成本开销。为了达到服务对请求的快速响应, 研究的重点就应该放在云服务调度策略的优化上。

随着移动互联网服务产业的极速发展以及互联网用户的井喷式增加, 对云服务的请求和使用量也大幅度地增长, 云服务管理平台处理同时到达的批量服务请求的情况越来越多。对批量服务请求的响应包括两方面的内容: (1)为互联网用户提交的每个云服务请求选择出其对应的最优服务组合。(2)面对批量服务请求, 在保证质量的前提下提供高效的云服务资源调度方案。整体而言, 对于服务请求的执行不仅要重视质量, 同时还要考虑效率问题, 所以面对激增的移动互联网用户以及大批量的云服务请求, 云服务管理平台在保证服务请求完成质量的同时, 也需要减少总的时间开销和成本开销。在此背景下, 建立云服务组合选择的优化模型与对应的云服务资源调度模型, 具有重要的理论意义和研究价

值。本文研究处理批量云服务请求的两个阶段——最优服务组合选择与云服务调度优化。

## 1.2 国内外研究现状

### 1.2.1 最优云服务组合选择

为了满足互联网用户提出的云服务请求的多功能性需求,优质的服务组合的选择就显得尤为重要。云服务组合是由一系列功能单一的细粒度云服务按照一定的逻辑顺序关联形成,单一功能的云服务的服务属性对整个云服务组合的组合也有较大的影响。所以在云服务组合的当前研究中,对云服务的描述与匹配,基于 QoS 的最优云服务组合的选择以及对服务组合的模型创建是几个重要的研究点。

#### 1.对云服务的描述与匹配

由于云计算是面向服务的体系架构,所有的服务按照一定的标准进行登记注册以及调度使用,在云服务使用的过程中还要注意服务请求与云服务的对应关系,所以云服务的描述与匹配就显得特别重要。在国内外众多云服务相关研究当中,常用基于语义的本来来描述云服务,利用这种方法,可以通过语义匹配的方式来获取与服务请求匹配的云服务。例如, Martin D<sup>[5]</sup>等使用基于本体描述语言 OWL 的服务描述语言 OWL-S,将语义网与云服务关联起来。OWL-S 将任意的单一服务作为本体,通过 OWL 的描述方式描述其的基本概要信息等。QWL-S 既可以描述云服务的语义信息,还将包含数据信息,甚至还包括服务提供者的信息。正是因为常用语义描述语言对云服务进行描述,过些年来国内有不少的研究是基于语义匹配的角度进行服务组合优化的<sup>[6,7]</sup>。

#### 2.基于 QoS 的最优云服务组合选择

基于服务质量(QoS)进行云服务或者云服务组合的选择是国内外在服务组合研究方面的重点。在国外相关会议和期刊中,基于 QoS 的最优服务组合的选择被普遍定义为:从大量的服务组合的选择中,在满足用户对某些局部服务质量属性要求的情况下,高效率地选出全局服务质量属性度量值最大的服务组合方案<sup>[8]</sup>。所以,选择方法与对应的优化模型则是在云服务组合技术研究中极其重要的部分。互联网用户对云服务质量的评价标准主要在于经济 and 性能两方面来考虑。服务质量属性分为多种,包括响应时间,成本开销, CPU 使用率,可靠性,安全性,供应商硬件负载度等等,由于各个不同的服务属性的度量标准,类型以及数量级不同,必须对不同量纲的属性度量值进行标准化规约<sup>[9,10]</sup>。然后通过各个单一服务之间的依赖关系和结构关系对云服务进行结构建模。不同的结构模型所得出的 QoS 聚合模型也不同,而所有的聚合模型本质上都是一个有多条件的决策模型。该模型

包括云资源供应商提供的服务候选集，服务 QoS 属性的集合以及各个服务质量属性的权重值集合，所要求的决策目标则是最优服务路径。除了较常见的用户对响应时间，成本开销等功能性属性的重视，随着互联网和云平台的进一步发展，基于信任机制，负载均衡的服务属性要求以及考虑物流服务的制造云服务组合的研究也越来越多。在国内相关的研究中，马林，朱明等提出了基于 QoS 的可靠保障服务组合调度算法<sup>[11]</sup>。叶枫，祝家钰，王鹏等分别针对云计算环境下的负载机制做了 QoS 约束和调度等研究<sup>[12-14]</sup>。

### 3. 服务组合研究思路

从当前对服务组合建模及实施阶段的研究来看，基于 QoS 的云服务组合的研究思路主要还是考虑人工智能，工作流等相关的方法来解决最优服务组合选择问题。例如在国外研究中，在处理云工作流的方法上常采用 BPEL(也称 WSBPEL 和 BPEL4WS)这种基于 XML 的服务交换语言来描述服务组合的流程<sup>[15,16]</sup>。除此之外，还可以将实施计划部署在 BPEL 的服务引擎上执行，但这样就很难实施动态的云服务组合。从人工智能的解决思路上来看，对最优服务组合路径的选择可以看成是对最优解的有策略的穷举。同时，分析服务组合的模型，可以看出基于 QoS 的云服务组合是一个多目标优化问题，限制条件是对某些局部的 QoS 服务属性的限制，而优化目标是全局 QoS 度量值达到最大。对于这种多目标优化问题，演化算法是很好的求解办法。以粒子群优化算法为例来看，每个解代表了一条服务组合路径，都可以看做是一个粒子，整个算法在迭代过程中，不断地靠近最优解。所以，在国内外的研究中，常常通过演化算法解决多目标优化问题这一思路，来分析解决基于 QoS 的云服务组合问题。

#### 1.2.2 云计算平台的调度体系与算法

调度一般指在一定的时间范围内按一定策略分配有限的资源进行服务供给或者生产制造。在云服务管理平台进行服务资源的调度要考虑多方面的因素并且做到：为每个云服务请求提供全局 QoS 最优的云服务组合，同时让所有服务请求的等待时间最短以及所有服务请求的执行时间最短。

#### 1. 服务调度架构

在服务调度架构的国内外研究方面，主要还是基于 Agent 的调度研究。Huang C, Chen D 和 Hu H<sup>[17]</sup>提出了基于 Agent 的调度体系结构，可以分别进行本地的服务调度和整体的服务调度，并且该服务调度架构可以适用于分布式系统，所以基于分布式服务资源的云服务管理平台也可以使用这种服务调度架构。随着制造服务在云环境下的大量使用，不少研

究也致力于制造云资源的服务调度,马军,罗国富,路迪等<sup>[18]</sup>基于 RFID 技术对制造云服务资源进行体系架构,并且提出云制造环境下的 Agent 这一概念,实现了云环境下服务资源自动化的调度。云服务管理平台的调度功能模块不仅要负责服务资源的调度,还要负责对资源供应商与互联网用户的数据交换以及整个宏观系统的负载度的监控。除此之外,用户请求序列的优先级和使用云服务和云服务组合的信任度也是个重要的考虑因素,文献[19,20]提出了基于信任机制的服务调度方法与框架。

## 2.服务资源调度方法

云计算环境下服务资源的调度也可以既借鉴传统的工作流调度或者流水线生产调度的相关方法。传统行业的资源调度和分配已经做得比较成熟了,而资源调度和分配本质上是一个多目标优化问题,迭代演化算法对该问题有着很好的求解效果,并且已经应用于传统资源调度问题中<sup>[21,22]</sup>。粒子群优化算法<sup>[23]</sup>,模拟退火算法<sup>[24]</sup>以及遗传算法<sup>[25]</sup>在传统调度方面就有很多的应用。传统生产服务调度有多种方法,如设置滚动窗口,目标最优化调度方法等。而这些研究思路和方法都可以很好的应用于云服务资源的调度研究中,同时考虑到云环境具有动态性,实时性等特点,云环境下的调度方法在结合传统调度方法的基础上还要考虑时刻都有海量的数据并发及交换这一情况。Laili Y, Tao F, Zhang L 等<sup>[26]</sup>基于制造云环境下数据动态交换的特点,充分考虑了服务资源之间的信息交换,提出了一种新的调度模型。M. Dakshayini D, H. S. Guruprasad D<sup>[27]</sup>提出了一种优化模型来解决云服务资源的有限利用问题。在云环境下进行服务资源调度还要充分考虑服务请求任务的截止时间以及资源供应商的承载能力等限制条件。Van den Bossche R, Vanmechelen K 和 Broeckhove J<sup>[28]</sup>充分考虑了调度截止时间这一因素,提出了一种基于混合云(hybrid cloud)模型的调度方法。Fang Y, Wang F 和 Ge J<sup>[29]</sup>则提出了云环境下平衡资源供应商运载的调度方法。

与传统调度不同的是云服务请求的处理需要考虑实时性,而且需要更加注意服务质量。特别是处理大批量服务请求同时到达的情况下。文献[30]研究了大量 HTTP 请求同时到来并且保证批量请求服务质量和资源供应商负载均衡的前提下进行资源调度和请求分配,为电子商务发展提供有力的解决方案。基于这种情况的研究多数将重点放在服务队列的优先级上,比如文献[31]基于服务质量和请求队列的优先级因素调度不同的服务。云服务请求的实时性让服务调度模型算法提出了较高的要求,特别是批量服务请求同时到达时,必须提高服务调度的效率,减少各个服务请求的等待时间。从处理批量服务请求的目标性来看,提高服务请求的成功率,减少服务请求执行的失败次数可以减少批量服务请

求总的执行时间。

### 1.3 存在的问题与拟解决方法

云服务平台处理批量服务请求分为两个阶段：(1)为每个云服务请求选择其最优的服务组合。(2)为批量服务请求调度其最优服务组合资源供其执行。当前的研究大多都是独立考虑这两个问题，而且在调度云服务资源的过程中，很多研究也并未考虑保证服务请求执行质量的这一前提。云服务大多是以 web 服务与制造云服务作为其表现形式的，而当云服务是这两种表现形式时，云服务管理平台同时处理批量服务请求的情况会很多。所以云服务平台常面对的问题是：如何在保证服务组合质量的前提下，采用高效的方案调度云服务以满足服务请求的多功能需求。为了解决这一问题，就需要将两个阶段综合起来考虑，但是这一方面的研究还较少。另外，作为云服务其中一种表现形式的制造云服务由于开展的时间并不长，所以在这两个阶段的研究也需要进一步的展开。

为了保证云服务请求执行质量的前提下减少执行的总时间，就需要把两个阶段结合起来考虑。所以在本文中，我们提出通过两阶段优化的方法解决这一问题。第一阶段是为每个云服务请求选择最优的服务组合，由于云服务资源供应商数目众多，并且能提供功能相似但是服务质量不同的云服务，所以需要基于服务质量建立云服务组合选择的模型，同时还需要考虑满足用户对局部服务属性的优化要求。在将实际问题转化为数学模型中，可以确定服务组合的全局最优 QoS 度量值为目标函数，用户的局部 QoS 优化目标为数学模型的目标条件。第二阶段是为批量云服务请求调度云服务使得云服务请求的执行总时间达到最少。在建立调度模型的基础上还要考虑保证服务质量，采用的是第一阶段为各个服务请求选择得出的最优服务组合，但是由于服务资源的有限，所以建立的模型要考虑优化调度方案减少服务请求序列的等待时间并建立数学模型。本文提出的两个数学模型都属于多目标优化问题，对于这一问题，传统的演化算法有着较好的求解效果。所以在建立模型的基础上，还要提出对应的解决多目标问题的优化演化算法。下图为拟解决方法的路线图，本文按照该流程做相关研究。

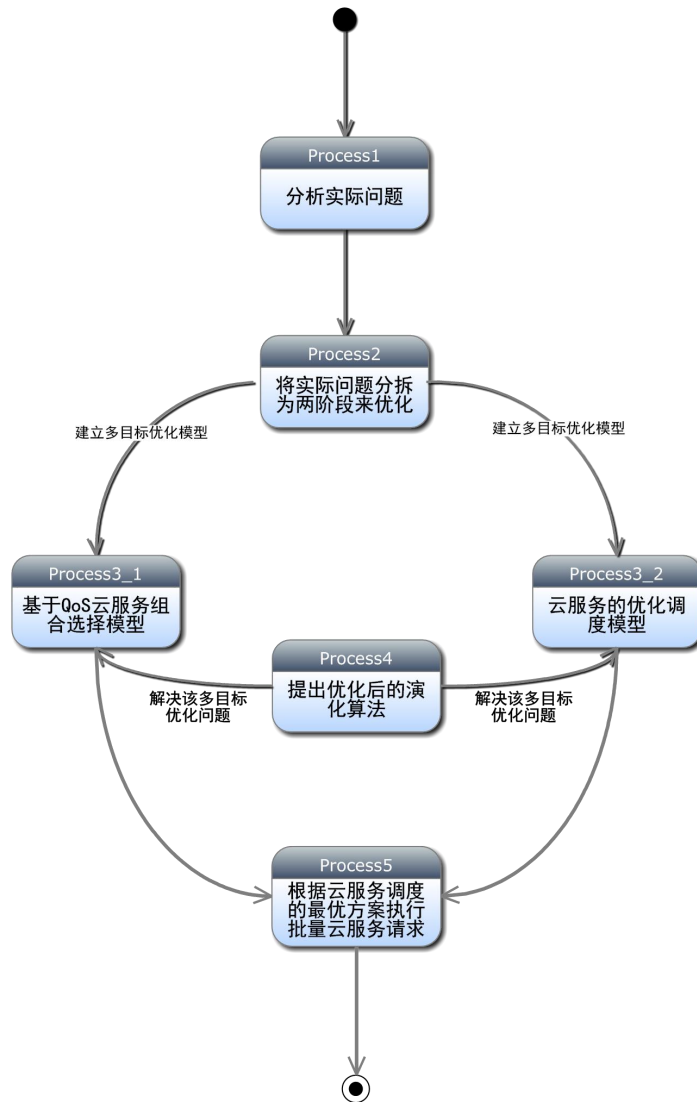


图 1-1 问题拟解决路线

## 1.4 本文的研究内容

本文主要研究的云服务组合与调度优化研究的两个阶段：(1)基于服务质量(QoS)为每个云服务请求选择最优的服务组合路径。(2)为批量云服务请求调度其对应的最优服务组合资源，使得整体执行时间开销达到最小。这两个过程中，有多个优化目标必须满足：(1)为互联网用户提交的服务请求选择全局 QoS 度量值最大的云服务组合路径(2)满足用户对局部的服务质量属性的优化目标。(3)在为批量服务请求调度服务组合资源的过程中，整体的执行总时间达到最少。(4)在调度的过程中，为每个云服务请求提供的服务组合资源达到全局 QoS 值最优(即使用第一阶段的选择的最优服务组合)。本文具体的研究内容展开如下：

### (1) 面向云服务的服务质量(QoS)描述与评价体系



通过分析云服务的具体属性信息,提取若干服务属性作为评价标准,并对这几种云服务属性做具体的描述与分析,给出对应的服务质量(QoS)精确化的数学表达方式,之后对服务质量(QoS)的度量单位做标准化规约。在此基础上,建立整体的云服务组合全局 QoS 度量值的数学模型,并且满足用户提出的局部 QoS 优化目标。

## (2) 基于服务质量(QoS)的最优云服务组合选择

主要针对基于服务质量的最优服务组合选择方法进行研究。分析云计算环境下 QoS 的特点以及当前批量云服务请求的具体情况,研究服务组合选择的具体的步骤,给出结合全局 QoS 与局部 QoS 评价的服务组合选择的具体数学模型。

## (3) 面向批量服务请求的服务资源调度优化

考虑面向批量服务请求同时需要处理时云服务管理平台的调度策略。在调度过程中,通过保证服务请求使用的服务组合是在第一阶段中选出的全局服务质量(QoS)度量值最高的服务组合,来保证服务请求的执行质量,减少失败次数以达到减少批量服务请求执行的等待时间和整体的执行时间的目标。

## (4) 基于改进粒子群算法解决两阶段优化问题

从本质上来讲,基于 QoS 的最优云服务组合路径的选取以及面向批量云服务请求的服务资源调度问题都是属于多目标优化问题,并且通过对问题的分析,可以建立对应的数学模型和限制条件。而人工智能研究领域里的演化算法对求解多目标优化问题有着较好的效果。其中粒子群优化算法是较为基础和常用的一种传统演化算法。但是由于传统的粒子群优化算法及其他演化算法容易造成过早的陷入局部最优,并且在算法迭代后期,算法容易过早收敛。针对这些问题,本文对基础粒子群优化算法做了改进。将粒子群优化算法的基础上加入杂交的环节,并且对学习因子进行非定值变化设置,在一定程度避免了算法早熟,同时将二阶振荡的环节引入到改进粒子群算法中,进一步丰富了粒子种群的多样性。基于改进的粒子群优化算法解决两阶段优化问题,本文主要介绍了粒子设计和适应度函数设计这两个关键的步骤。

## (5) 实验部分

本文的实验部分分为三个部分,分别是:(1)算法效果比对。通过将改进的粒子群优化算法与另外两种对比算法做算法性能方面的对比,从而体现改进算法的效果。(2)仿真本文提出的模型与算法解决云服务组合与调度问题。(3)将本文提出的云服务调度模型应用到一个以各类转接线的批量生产制造为背景的仿真实例当中。

## 1.5 论文组织结构

论文的正文部分一共分为六个章节的内容，具体的组织安排如下：

第1章：绪论。介绍本文研究课题的相关背景内容，目的以及研究意义。同时也介绍了本课题当前国内外的一些研究进展。

第2章：本课题的涉及到的领域中一些关键技术与相关的研究。介绍了云服务与云计算的相关概念，以及在解决云服务组合以及云服务资源调度的研究中所出现的一些新的关键技术与成熟的解决方案。

第3章：提出了本文所主要研究并解决的“两阶段”问题的第一阶段——解决基于QoS的最优云服务组合路径的选取问题的方案与模型，并且提出了云服务与云服务组合的描述方式以及服务质量(QoS)的评价标准。在此基础上，提出了本文所主要研究并解决的“两阶段”问题的第二阶段——面向批量服务请求的调度策略与模型，并且给出了对应的数学模型。

第4章：提出了解决上述两阶段问题的算法——改进的粒子群优化算法，并且结合本文的具体情况给出了算法的描述。

第5章：实验仿真部分。基于本文提出的模型和改进算法，给出了多方面的仿真和算法对比。

第6章：总结和展望。总结本文的主要研究内容和不足，提出下一步的研究方向。

## 1.6 本章小结

本章首先描述了课题的研究背景和国内外相关研究现状。在这个基础上，提出了当前云服务组合与调度研究中存在的问题和拟解决的技术路线。之后对本文的研究内容作了简要的介绍。最后分章节对文章的结构内容作了介绍。

## 第 2 章 相关研究与关键技术

### 2.1 云计算与云服务的相关介绍

#### 2.1.1 云计算的发展与应用

随着互联网技术的发展,云计算与多种技术有着交叉研究,其中与应用科学 workflows 技术的交叉就是一种新的研究,而云 workflow 就是两者结合的产物。除此之外,数据存储,虚拟服务主机等相关新型业务也逐渐成为较为活跃的云计算应用。云计算的运行模式就是通过互联网连通各个服务资源,通过统一的数据交换和服务接口将它们分配给各个服务请求用户使用。国内外诸多企业中,研究和使用的云计算最多的就是 Google<sup>[32]</sup>,因此它也研发出众多的新技术与产品,如 MapReduce, Hadoop, GFS, BigTable 等新的技术方案机制。近些年,云计算在国内的研究也有大幅度的进展,多个研究中心陆续在我国建立。国内的互联网企业也越来越多的建立和开发云平台与云应用。百度,阿里巴巴等企业也开始建立百度云和阿里云等功能越来越完善的云平台。

#### 2.1.2 云服务及其标准

正如上文所言,云计算的一大特点便是资源的按需付费,而云平台资源供应商根据需求提供服务资源参与到云平台的服务供给当中。对于客户而言,选择信誉度高以及可靠性高的云资源供应商尤为重要。如何高效选择使用可靠且性价比高的云服务资源成为了一个需要解决的问题。同时,云服务可以通过共享硬件资源等多种方式来提供存储服务或者大规模运算资源给互联网用户。云服务的集中使用需要统一的标准。WSDL<sup>[33]</sup>就是一种描述服务的文档标准,大多数的服务供应商都支持这种服务描述标准。特别是当调用云服务时必须要知道对应服务的描述标准与描述内容。所以在调用具体服务时,必须提前获得对应服务的 WSDL 文档以便创建服务接口。所以对于云服务管理平台来讲,其拥有的服务资源采用同一种服务描述标准是至关重要。常见的服务搜集和管理平台比如有 UDDI(Universal Description Discovery and Integration)。UDDI 作为公共开放的服务平台,通过服务的在线注册来丰富自己的服务目录信息,同时 UDDI 还提供了标准服务的查询功能。除了服务描述文档标准以外,统一的访问协议和数据传输标准文件也是必不可少的。SOAP(Simple Object Access Protocol)是互联网和云计算平台的常见的服务访问协议<sup>[34]</sup>,而

其使用的 XML 也是常用的标准数据传输文件。因为云计算平台的资源分布广泛，且不利于管理，所以统一的访问协议和数据传输标准就极为重要。

## 2.2 云服务组合与调度相关研究

### 2.2.1 基于 QoS 的云服务组合研究

#### 1. 云服务与云服务组合概述

最早的云服务技术应用于云存储上，通过分布式的数据库以及良好的网络带宽来完成用户与云平台的数据交换工作。除此之外，还有数据同步等涉及到数据的云服务。云服务的主要体现形式是 web 服务，而现在随着制造业与云计算平台的紧密结合，制造云服务也成为新型的云服务形式。云制造产业采用“线上监控，线下生产”的服务模式。云制造作为一种新型的网络化车间制造模式逐渐成为了制造业的创新性的发展方式。同时，制造云服务也可以看作是云计算在传统制造业的具体应用，它集合了物联网等新型技术，对制造服务资源供应商和工序分布广泛的制造业提供了技术支持<sup>[35]</sup>。在云制造生产体系中起着极为重要作用的是制造云服务管理系统，其主要完成对制造云服务资源进行统一的管理，任务划分，物流配送安排以及实时监控等重要功能。

在云服务平台中常见的是功能单一且只能满足简单需求的云服务。而当前随着移动互联网产业和云计算技术的加速发展，用户提出的服务请求已经越来越复杂，单一的云服务能完成的功能显得越来越有限，所以亟需复合型的服务组合提供更加综合型的功能需求。而云服务组合技术可以有效地连接各类云服务以满足更加复杂的云计算领域的需求。同时由于云服务资源的充足，存在大量功能相似，但服务质量不同的服务，所以如何高效地发现用户需求的服务就显得特别重要。

#### 2. 基于 QoS 的服务组合选择

单个云服务通常指提供的简单且功能单一的云服务，而云服务组合技术可以利用云平台上分布的各个细粒度轻量级且极易调用的云服务组合形成功能强大的粗粒度的云服务组合。云服务组合可以满足某些互联网用户的一些复杂或者个性化的定制需求。为了完成这一目标，可以通过相应的方法和一定的选择标准将一部分资源供应商提供的功能单一的云服务组合起来，形成最优服务组合路径并提供给用户。云服务管理平台向服务请求者提供云服务组合以满足其需求，这不仅提高了云服务的重用性和扩展性，也能满足单一云服务所难以完成的较为复杂的用户请求。所以说使用多个功能单一的云服务组合起来形成复合的云服务组合是提高互联网用户服务使用效率以及产品的个性化程度的必要途径。由此

可见，服务组合的选择办法就显得尤为重要了，在考虑服务质量的前提下，基于 QoS 的服务组合选择办法是相关研究中的重中之重。

基于 QoS 的服务组合选择算法主要还是依靠对各个服务资源的评价机制以及 QoS 度量值的综合计算方式，而如何协调处理这些 QoS 度量值数据则有多种。

#### (1) 根据服务资源的 QoS 度量值之和排序的方法

比较基础简单的处理基于 QoS 的最优服务组合选择问题的方法是将各个服务的 QoS 度量值相加，选出其中服务质量度量值总和最大的一组服务组合作为最优的服务组合。这种办法简单易行，同时将具体的问题转换为了简单的数学模型，所以有较多的数学工具和模型可以用于求解。但是这种办法也有些不足之处。比如，选择的最优服务组合虽然综合的 QoS 度量值是最高的，但是可能其中某些几个比较关键的局部服务质量指标并不符合用户的要求。

#### (2) 综合的服务组合选择方法

文献[36]提出了一种基于混合 QoS 的综合服务组合选择算法 WSSHQoS (Web Service Selection From Hybrid QoS Metric)，该算法还支持模糊 QoS 属性的计算。WSSHQoS 根据不同服务属性的权重值的不同，有主次的处理不同的服务 QoS 属性，将其中的模糊 QoS 属性转换为三角模糊数。除了模糊 QoS 属性之外，常见的 QoS 属性的类型还包括确定数类型和区间范围类型两种。WSSHQoS 可以综合几种服务属性类型，选择出最优的服务组合。但这种算法的时间复杂度相对较高，而且算法过程也比较复杂。

#### (3) 基于不完整的 QoS 属性的服务组合选择方法

由于云计算网络环境的复杂性以及各个服务供应商的可提供的服务数量过多，在服务属性的 QoS 数据收集的过程中，难免出现信息不全，丢失或者无法获取等情况。同时，用户提出对局部 QoS 属性限制的要求时，也可能会出现信息表述不全等状况。所以在实际服务组合的过程中，预处理的服务 QoS 属性有些是不完整的，而基于不完备的服务 QoS 属性数据的服务组合选择方法是很有必要的<sup>[37,38]</sup>。

#### (4) 基于演化算法求解最优服务组合选择问题

基于 QoS 的最优云服务组合选择问题本质上是受约束条件限制的多目标优化问题，而人工智能研究领域中的演化算法是解决这类问题的常用办法。Shangguang Wang 等<sup>[39]</sup>运用粒子群优化算法解决了基于 skyline 的云服务组合问题。由于服务组合选取过程中需要考虑多方面功能性和非功能性的因素，所以最优服务组合解的空间也是高维度的，在此背景下，Fanjiang 等<sup>[40]</sup>提出采用遗传算法进行最优解的搜索以解决基于语义的自动服务组

合问题。Mardukhi F 等<sup>[41]</sup>采用遗传算法将服务组合问题中的全局限制条件分解为若干局部限制条件，再通过线性搜索选择最优服务组合。Wu Q,Zhu Q<sup>[42]</sup>则运用蚁群算法解决了事务层次基于 QoS 的动态服务组合优化问题。但是传统的演化算法易于陷入局部最优，在迭代后期收敛速度过快，所以要使用演化算法解决基于 QoS 的云服务组合问题，就需要对传统的演化算法进行改进。文献[43]通过粒子多维圆周轨道这种新思路改进了算法，增强了算法的全局搜索能力，并且扩展了种群的多样性，这种对粒子的改进的方式是当前较为常用的改进方法。李昌兵等<sup>[44]</sup>提出了基于进化博弈和非支配排序思想的层次粒子群算法(HPSOBLP)，并将其应用于非线性规划问题的求解中。李松等<sup>[45]</sup>引入变异算子改进了基础粒子群算法，使得粒子跳出局部最优的能力进一步加强，并将这种改进的粒子群算法优化算法运用到实际模型中。王廷梁等<sup>[46]</sup>则通过改变速度更新公式中的学习因子以达到改进算法的目的。杂交粒子群算法是基础粒子群优化算法的另一种改进形式，文献[47]证明了其有着更快的收敛速度，并且所求得解的质量更高。

### 3.服务组合仿真工具与成熟模型

近些年对服务组合的研究已经有了较成熟的性能测试平台以及仿真模拟平台等等。云服务主要的表现形式之一的是 web 服务，而乔治州立大学的研究人员在 2003 开发了 web 服务组合平台 SECT(service composition and execution tool)<sup>[48]</sup>。SECT 能够对服务组合进行基于 WSFL 的详细描述，并且有着图形化的展示方式以及较强的用户体验。同时还支持在对服务组合过程的简单的性能分析。Petri 网是一种较为常见的数学建模工具。Petri 网<sup>[49]</sup>具有一般数学工具的优势：支持数学表达式的描述与展示，同时还支持可视化的展现。Petri 网能够支持工作流或者其他服务流程的展示，所以常被用于工作流调度的结构模型仿真工作以及服务组合的模型建立，早在上个世纪八十年代 Stochastic Petri Net 被作为动态的形式化工具，在基础的 Petri 网上加入性能测试的功能，能更加高效地模拟具体问题<sup>[50]</sup>。基于 Petri 网的模型能够全面的描述系统的特征。常见的利用人工智能方法解决的服务组合问题的思路就是，先利用 Petri 网等数学建模工具将具体问题表示为数学优化问题，再通过某些演化算法(如遗传算法)来有策略地搜索最优值。所以基于 Petri 网的仿真模型平台是在国内外服务组合研究中常见的仿真途径。其他常见的服务组合仿真平台比如有斯坦福大学在 2002 年所研制的模拟工具 karmaSim<sup>[51]</sup>。该模拟工具中就包含了 Petri 网的数学模型模块，同时还包含了基于本体的服务描述功能。karmaSim 通过服务对 Petri 网模型的转换，完成具体问题到数学问题的转换。

## 2.2.2 云服务资源调度研究

### 1. 概述

传统意义上的云环境下的服务资源调度属于调度功能单一的云服务,而随着互联网与云计算的技术逐渐发展,单一功能的服务很难满足多功能复杂服务请求的需求,所以针对云服务组合调度策略的优化就显得特别重要。在服务资源有限的情况下,如何制定调度方案,使得批量的服务请求都能由其对应的最优服务组合提供服务。面向批量云服务请求的调度要考虑多种因素:保证提供给请求的服务的质量,网络带宽环境,服务请求队列的优先级等。对于实际的云服务调度问题,必须考虑对应的调度策略,服务供应商的主机性能,同时确定相应请求的最优服务组合路径等。

### 2. 服务调度目标与方法

云服务资源的调度策略主要指:为了满足用户服务请求执行的高效,高性价比以及其他非功能性需求而制定的资源分配策略。而当前的服务资源调度的主要目标与对应方法有:

#### (1) 减少服务请求执行时间或者制造云服务任务的生产时间

由于云计算模式是属于商业模式,同时云计算服务资源的使用是属于按需付费,所以提高服务请求的执行效率也就降低成本开销。制造云服务随着云制造产业的逐步发展而被越来越多的使用,需要快速的服务响应方式以减少云制造整体的生产时间。张卫等<sup>[52]</sup>就提出了基于云模型蚁群优化的一种调度算法以解决制造云资源的调度问题。

#### (2) 均衡服务供应商的负载度

保证服务资源供应商主机运行的成功率,均衡各个供应商的服务请求或者生产任务的数量以降低主机的宕机率,都对批量云服务的执行至关重要,所以负载均衡是较多服务资源调度的研究目标。陈国斌<sup>[53]</sup>提出的以均衡负载度为目标的算法,同时该算法也是基于分布式的资源调度。郭平等<sup>[54]</sup>提出了一种改进了 min-min 的调度算法,提高了云平台资源利用率,同时也均衡云服务供应商主机的负载率。宽带状况会影响数据交换速率,所以它也是云平台调度问题所要考虑的重要因素,郑卉等<sup>[55]</sup>提出了考虑宽带因素的调度算法,同时还保证了负载均衡。

#### (3) 保证服务质量(基于 QoS)

高效的服务调度需要减少服务请求在执行过程中的等待时间,而减少服务请求等待时间的前提是要提高服务请求执行的成功率。为服务请求提供高质量的云服务或者云服务组合可以提高服务请求的执行效率,减少失败次数,从而减少服务请求在执行过程中的等待

时间, 节约成本。基于服务质量(QoS)的调度方案能在保证服务质量的前提下完成调度的执行。施进发等<sup>[56]</sup>通过多个维度的 QoS 分析, 提出了针对性的调度方案, 进行了对应的优化。丁宇光<sup>[57]</sup>基于服务质量 QoS 约束提出了云计算环境下的优化调度方案。针对科学 workflow 在云计算领域的应用, 文献[58]实现了基于服务质量的云 workflow 调度算法。文献[59]在服务质量约束的同时, 保证了主机的负载均衡。上文提到的文献[20]所提出的服务调度框架也是基于保证服务质量的前提, 除此之外文献[60]也提出了基于 QoS 解决复合功能服务调度的解决方案。

## 2.3 本章小结

本章首先介绍了云计算领域的相关概念以及云服务的相关描述与标准。然后在此基础上分别说明了以下两方面的相关研究内容: (1)基于 QoS 的最优云服务组合选择的方法与成熟模型。(2)云计算环境下云服务资源调度的目标与成熟方法。与本课题相关的研究内容为后续研究工作指明了方向。



## 第3章 两阶段优化的云服务组合与调度

### 3.1 引言

云服务管理平台针对批量云服务请求同时到达时的处理主要有两个阶段：(1)基于QoS的最优云服务组合选择。(2)面向批量服务请求的云服务调度。从经济与性能两方面来看，云服务请求的执行在这两个阶段中，可以针对云服务组合的服务质量与执行完成的总时间等多个方面做针对性的多目标优化。所以本章针对两个阶段的各自特点建立了模型，同时确定了优化调度策略，然后给出了两阶段优化的整体流程。

### 3.2 基于QoS的最优云服务组合选择

#### 3.2.1 云服务组合结构体系与实现过程

云服务体系由三个重要的角色构成：云服务管理平台，云服务请求者以及云服务资源供应商。云服务请求者向云服务管理平台发起请求，云平台通过对服务请求信息的分析处理，为服务请求选择最合适的单一云服务或者云服务组合，并向对应的云服务资源提供商发出请求信息，将对应的云服务调度到相应的服务节点。对于云平台而言，拥有较为丰富的云服务资源极为重要，同时云服务提供商也需要及时在云服务管理平台上进行服务信息的注册，以使用户在云平台针对相应需求查找对应的云服务。

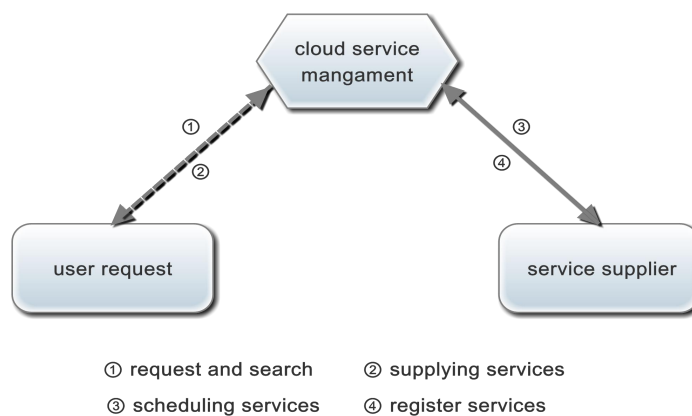


图 3-1 云服务体系构成图

下图是整个云平台服务组合体系自顶向下的结构图。子请求集合由  $T=\{T_1, T_2, \dots, T_m\}$  表示。

各个子请求之间有着清晰且明确的顺序关系,而这种顺序关系通常用 workflow 图或者任务流程图来表示,在本文中由有向无环图(DAG)来表示各子请求之间的依赖关系和逻辑关系。在 DAG 中,结点的集合  $V=\{V_1, V_2, \dots, V_m\}$  代表一个完整的服务请求经划分后得到的所有子请求,所以节点集合与子请求集合  $T$  对应。结构体系的第二层是资源提供层。云平台供应商通过硬件虚拟化技术和以软件为载体的中间件向用户提供云服务。假设有  $j$  个供应商可以为子服务请求  $T_i$  提供对应的云服务,则  $T_i$  的候选云服务集合可以表示为  $CS_{T_i} = \{CS_{T_i}^1, CS_{T_i}^2, \dots, CS_{T_i}^j\}$ 。用来表示子请求  $T_i$  由云服务  $L(i)$  来完成,其中  $L(i)$  是子请求  $T_i$  在其候选云服务集合中选择的服务,即  $L(i) \subset CS_{T_i}$ 。所以子请求集合  $T=\{T_1, T_2, \dots, T_m\}$  所使用的一套完整的云服务组合的路径可以表示为  $Crs = \{rs_{L(1)}^1, rs_{L(2)}^2, \dots, rs_{L(m)}^m\}$ 。第三层是数据读取层,在云平台中,主要由分布式数据库构成,第二层的云服务资源将数据存储到底层的数据库中,同时也从数据库中读取数据。数据读取层不仅负担着数据读取的任务,而且还能实现数据共享和数据传递。而数据交换过程中采取统一的 XML 数据格式。除此之外,数据层还要定义数据控制流的逻辑,同时通过对数据流的控制提高云服务执行的自动化程度。

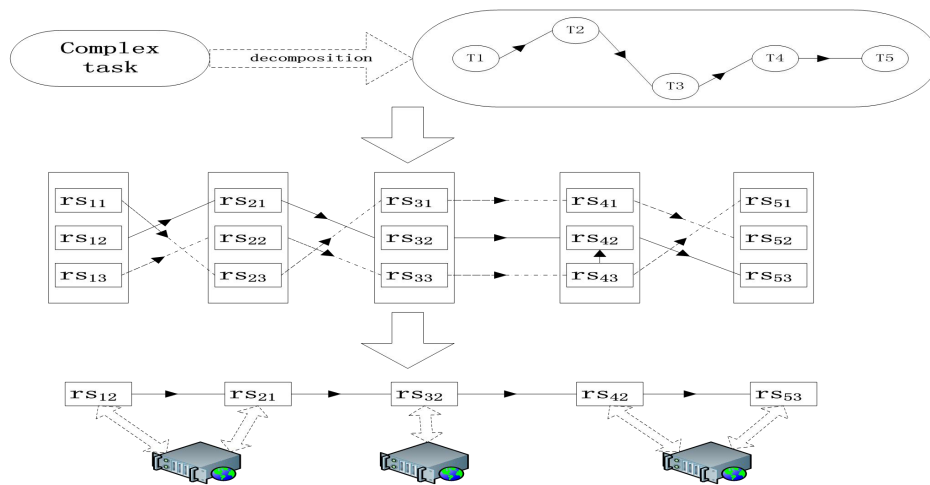


图 3-2 云服务组合结构体系

在云平台上的服务请求与供给是典型的商业模式,云服务资源供应商封装服务并将其提供给互联网用户,如何选择最优的云服务以及最有效地调度服务资源则依赖于云服务管理平台。云平台的用户希望获得的服务是优质的,并且能在最快的时间获取到以及付出尽量少的成本开销。为了达到这样的目标,首先云服务供应商要将自己可以提供的云服务信息发布到互联网上,而云服务管理平台根据相应的发布信息将云服务的稳定,特点,成本开销等 QoS 属性信息登记在管理系统中(比较著名的平台如 UDDI 就是这样做的)。而当有

用户提交云服务请求给云平台时，云平台根据用户需求和个性化定制要求，选择出最优的云服务或者云服务组合。当用户将较小规模或者请求内容单一的服务请求提交后，云服务管理平台只需要为其选择功能单一的对应云服务即可，当然也可以采取语义匹配等方法为服务请求选择最配对的云服务。但是如果提交的请求是大规模复杂且需求功能多样性的，或者具有较高的个性化定制程度的，就需要将单一的服务组合起来。同时从整个云平台的角度来看，高效优质的云服务组合模型及算法还可以为服务供应商节省资源和成本。

从功能目标上来看，云平台通过服务组合完成用户提出的较为复杂和频率较高的粗粒度服务请求。为比较复杂的云服务请求选择其相应的云服务组合，一般经历三个过程：

- (1) 首先将粗粒度的服务请求划分为子请求，保证每个子请求使用功能单一的服务。
- (2) 根据在云服务管理平台登记的候选资源供应商的相应信息，形成对应的候选服务资源集。通过对服务的分类可以提高服务选择的效率，同时也降低了服务组合的复杂度。
- (3) 在各个候选服务集合中确定云服务组合的最优路径。

制造云服务是代表性较强的云服务，随着传统制造业与云计算的充分集合，制造云服务的使用越来越频繁。下面以一个制造云服务组合使用的事例描述这三个过程：有某一互联网用户需要通过云平台定制一批个性化程度较高的产品，首先他通过移动端或者 PC 端向服务器端提交所需产品的细节信息和相关要求。服务器端接收到服务请求后将其划分为若干子请求：选材，焊接，成型打磨，测试等。然后云平台根据已有的服务资源集合以及用户对时间成本等要求完成最优服务组合路径的选取。在选取过程中，云平台不仅要考虑时间成本等功能性属性，还要考虑可靠性，可用性等非功能性属性指标。之后再调度相应的服务完成用户的各个子请求。整个云服务组合的分解与实现过程如下的示意图：

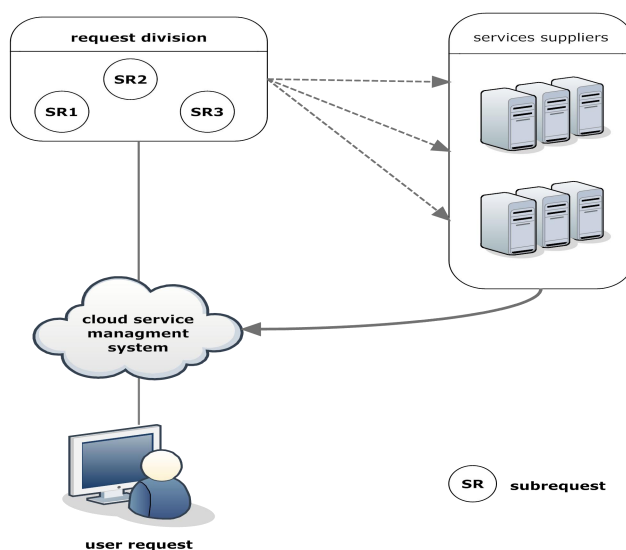


图 3-2 云服务组合分解与实现过程

### 3.2.2 基于 QoS 的云服务评价机制

QoS 作为面向服务体系中重要度量标准, 反应了细粒度功能单一的云服务和粗粒度服务组合的功能性和非功能性服务质量水平。云平台上具有多功能需求的服务请求选择服务组合也是基于服务质量(QoS)的。QoS 属性分为功能性 QoS 属性和非功能性 QoS 属性。常见的功能性 QoS 属性主要包括时间和成本等。功能性 QoS 属性主要描述了使用者较为关注的供应商提供服务的时间和成本开销, 这是云平台面向服务体系中用户非常常见和重视的参考因素。非功能性 QoS 属性包括可靠性, 可用性等, 非功能性指标体现了服务与请求是否匹配以及使用该服务的数据以及程序的安全性等等。

目前对 QoS 属性的研究比较广泛, 同时由于涉及到云服务评价的因素也很多, 所以对于 QoS 属性对服务组合的影响, 不仅要具体了解涉及到的 QoS 属性的具体含义, 还有针对不同的服务属性的具体表达和度量标准进行研究。除此之外, 还需要对不同度量单位的 QoS 属性进行规约, 使之能在统一的标准下进行度量运算。

#### 1. 服务属性描述

云服务评价标准是描述云服务功能性和非功能性的指标, 其必然是一个包含多个指标的集合体, 而每一种服务指标的具体使用环境又各不相同。一部分服务指标属于基本常见性指标, 这些指标在绝大数的云服务评价体系中都被作为基础指标来使用, 而功能性 QoS 属性就属于这一类。而另外一种属于运行性指标(runtime metric), 非功能性 QoS 属性就属于这一类, 这类服务指标对于用户选择可靠, 稳定, 完整性较强的云服务有非常重要的参考意义。除了这两类 QoS 服务属性外, 还有一种依靠历史评价得出的, 比如该云服务的信誉度等等。

下面是对几种属性的定义:

**时间:**该属性属于云服务基础性指标。主要指服务资源完成服务请求的预计响应时间。响应时间开销一直是云平台提供服务中非常重要的参考因素。对于不同的细粒度请求, 服务提供商应事先提供对应请求种类的云服务的响应时间, 以使用户提前进行时间开销的计算。

**成本:**该属性属于云服务基础性指标。表明了用户为了获取云服务所需要支付的实际金额, 属于服务资源完成服务请求的成本开销, 该指标由资源供应商直接确定。成本开销也是非常重要的参考因素。和时间属性一致, 服务提供商也应提前提供对应的服务完成请求的成本开销供参考。

**信誉度:**该属性依靠于历史评价性指标。主要由客户的历史评分和历史交易情况决定。

收集待选择的云资源服务供应商的客户历史评价,并计算每个资源服务商的客户历史评价得分,得出相应的可靠性 QoS 度量值。可用性作为重要的非功能性 QoS 属性采用以往客户的评分作为评价基准,可以较为合理地体现服务机器基本的非功能性属性,也能在一定程度上体现了服务与请求的匹配性。

**可靠性:**主要由服务资源上运行任务的历史成功率来体现。可靠性保证了服务的请求与提供可以正常进行。可靠性作为不可控因素受多方面情况的影响,如网络环境,机器使用时间以及当前在线用户使用数目等。所以作为这样的较难描述的非功能性 QoS 属性,采用服务提供商以往完成服务的成功率来表示。

**可访问性:**云资源供应商向云平台用户提供服务的程度。由于各个资源供应商的个体因素以及网络传输情况,有可能存在访问服务不成功的情况。这个指标表示了一种访问成功的可能性的尺度。主要用于表示在某一段时间内云服务供应商能够成功为用户的服务请求提供服务的比率。由于不同的时间内,访问的成功率不同,在本文中,我们用成功率记录数组 *AccessibilityTable*[] 来存储一天中每个小时的访问成功率。

**安全性:**随着云计算以及大数据领域的大规模发展,数据的安全性也越来越受到关注。若没要求,安全性这一属性可以由模糊评价的方式来描述,比如:“较优”,“一般”等,即在某个区间内。在本文中我们规定为:“较优”,“良好”,“一般”,“不合格”。分别由  $a_1, a_2, a_3, a_4$  来表示。另外,表达式  $[a_1, a_2, a_3, a_4]$  代表了其中一种评价。由于每个子服务的权重值不一样,因此用  $\beta_i$  代表每个子服务安全性方面的权重比例。

**CPU 使用率(以及其他硬件指标):**云服务资源供应商 CPU 使用率以及其他硬件指标在一定程度上反应了云服务提供商的机器性能。特别针对高耗能的服务请求,对云服务资源供应商的硬件选择也较为苛刻,对机器性能的要求也较高。由于不同的硬件指标不同,所以该属性的表达式也较难描述。

下表是本文主要考虑的六种 QoS 属性的度量表达式:

表 3-1 各 QoS 属性的度量表达式

QoS 属性	度量计算公式
时间	$ExeTime_{rs_{L(i)}}^i$
成本	$Cost_{rs_{L(i)}}^i$
信誉度	$\sum Scores_{rs_{L(i)}}^i / number(customers)$
可靠性	$number(succes_{rs_{L(i)}}^i) / number(fail_{rs_{L(i)}}^i)$
可访问性	$AccessibilityTable[TimePo int_{rs_{L(i)}}^i]$
安全性	$\beta_{rs_{L(i)}}^i * [a_1, a_2, a_3, a_4]$

云服务组合用户在要求全局 QoS 值最优的情况下还有局部 QoS 的优化目标,而这些

局部限制也是多目标优化问题的限制条件。使用可靠性高的云服务资源是客户的常见局部 QoS 要求。对于较为复杂的服务请求，用户可能会要求使用的云服务组合必须满足一定的可靠性要求(比如达到候选资源供应商的平均水平以上)，才会使用其服务。可以由下面的表达式描述：

$$reliability_{sum} \geq reliability_{avg} \times Subreq_{num} \quad (3-1)$$

同可靠性一样，用户可能对可用性属性的有一定限制，比如其可用性都必须高于一个基本值。表达式如下：

$$MIN\{serviceability_{rs_{L(i)}}^i\} > serviceability_{basic} \quad (3-2)$$

## 2. QoS 规约

不同 QoS 属性的单位不一样，描述云服务的方式也不一样。而 QoS 属性的多样性体现在 QoS 属性获取方式的多样性以及 QoS 度量值表达的多样性上。从 QoS 属性度量值的获得方式上来看，主要有以下两种：

### (1) 提前提供类型

这一类型的 QoS 属性的度量值由云资源提供商提前提供给云服务管理平台，属于提前预设的常量值。一般如功能性 QoS 属性：响应时间，成本开销等 QoS 属性。

### (2) 历史数据统计类型

这一类型的 QoS 属性的度量值由云资源提供商的历史执行数据统计获得。而这一类统计型数据更能体现云服务长时间的服务稳定情况。所以一般如非功能性属性：可靠性，安全性等 QoS 属性需要历史记录数据来获得。

QoS 属性度量值表达的多样性主要体现在不同的 QoS 指标的评价表达方式的不一致。主要分为以下三种：

(1) 具体数值类型。如整型和浮点型的数值类型。例如响应时间和成本开销等 QoS 属性的度量值。而具体数值类型也是最常见和最基础的度量值表达类型。

(2) 处于某一区间内。例如某服务具有服务时间区间这一属性，表示它在这一时间段内才提供对应的云服务。也就是说该服务的此属性度量值区间在于该服务时间的区间内。

(3) 模糊类型。一些 QoS 属性的度量值使用的是一些评价类的描述词，比如：“一般”，“较好”，“极佳”，“不合格”等等。除此之外，还有用等级话语言描述的，如：“一级”，“高级”，“中级”等等。而这些评价和等级类描述表达了不同度量值之间的比较关系。

在对 QoS 属性进行度量计算的时候，一般需要先对数据进行预处理，使得不同的 QoS

属性拥有相同度量单位。比如 QoS 属性中的时间花费是以“小时”计算，成本开销是以“元”或者“美元”计算的，而可靠性是以“高”，“一般”，“低”等描述的，所以不同的 QoS 属性描述的度量单位不一样。为了在算法处理阶段更加方便，我们需要将不同的 QoS 属性进行规约，即将不同单位的 QoS 属性都转换为度量[1, 50]的数值。由于 QoS 属性分为“正向 QoS”和“逆向 QoS”。比如可靠性就是“正向 QoS”，可靠性越高，客户使用该服务的可能性就越高，而时间和成本就是“逆向 QoS”，花费的时间和成本越多，客户选择该服务的可能性就越低。在本文所考虑的逆向 QoS 属性为时间和成本，正向 QoS 属性是可靠性与可用性。正向 QoS 和逆向 QoS，其规约方式不一样。通过对 QoS 属性的规约达到对正向 QoS 和逆向 QoS 的规范，使之达到同一的度量单位。下面两个公式分别是正向 QoS 属性和逆向 QoS 属性的规约公式，max 和 min 是同一 QoS 属性中的最大值和最小值。

$$QoS_{st} = \begin{cases} \left[ \frac{QoS_{forward} - \min}{\max - \min} \times 50 \right] & \text{(正向 QoS 规约)} \\ \left[ 50 \times \frac{1}{\frac{(QoS_{reverse} - \min)}{\max - \min}} \right] & \text{(逆向 QoS 规约)} \end{cases} \quad (3-3)$$

### 3.2.3 全局 QoS 与局部 QoS 相结合的云服务组合模型

在云环境下为多功能需求的服务请求选取最优服务组合问题，本质上是一个多目标优化问题，特别是有局部 QoS 限制条件的情况下，局部 QoS 属性(特别是非功能性 QoS 属性)的度量值可能受到某些限制，比如被要求高于某一度量值或者高于候选供应商的平均水平。所以有必要提出全局 QoS 与局部 QoS 相结合的云服务组合模型，同时基于 QoS 云服务组合的可配置型也体现在此，不同的用户对不同的服务指标有着不同的重视程度。例如云制造业的用户可能比较看中制造云服务的生产周期和生产成本，而使用分布式计算的用户对稳定性，安全性以及完整性有着较高的要求。对于不同用户，其所看重的 QoS 属性也不尽相同，所以可以通过调整各个权重值来调整目标函数，从而满足不同用户的需求。集合  $Q$  代表了全局 QoS 属性所包含的多种局部 QoS 属性，即  $Q = \{QoS_1, QoS_2, \dots, QoS_i, \dots, QoS_k\}$ 。而  $\alpha_w$  是各种 QoS 属性在全局 QoS 中各自所占的权重比例，且  $\sum_{w=1}^k \alpha_w = 1$ 。下面的表达式描述了子服务请求  $T_i$  使用对应服务资源  $L(i)$  的各个 QoS 属性度量值的总和：

$$QoS(T_i) = \sum_{w=1}^k (\alpha_w \times QoS_w(rs_{L(i)}^i)) \quad (3-4)$$

其中  $QoS_w(rs_{L(j)}^j)$  代表了子服务请求  $T_j$  使用服务资源  $L(j)$  过程中属性  $QoS_w$  的度量值。子服务请求集合  $T=\{T_1, T_2, \dots, T_m\}$  当前选择的一套完整的云服务组合的路径可以表示为  $Crs = \{rs_{L(1)}^1, rs_{L(2)}^2, \dots, rs_{L(m)}^m\}$ 。不受局部 QoS 限制的情况下，云服务组合的全局 QoS 表达式可以描述为：

$$QoS(T) = \sum_{j=1}^m \sum_{w=1}^k (\alpha_w \times QoS_w(rs_{L(j)}^j)) \quad (3-5)$$

结合全局 QoS 与局部 QoS 的云服务组合表达式描述如下：

$$QoS(T) = \begin{cases} \sum_{j=1}^m \sum_{w=1}^k (\alpha_w \times QoS_w(rs_{L(j)}^j)), & Crs \text{ 满足局部限制} \\ 0, & Crs \text{ 不满足局部限制} \end{cases} \quad (3-6)$$

从上面的表达式可以看出，用户对不同的 QoS 属性的权重值设置的不同会在很大程度上影响最优服务路径的选择。因此结合全局 QoS 与局部 QoS 的云服务组合模型的可定制性也体现在该模型可以根据用户的具体请求信息和局部属性的要求来调整服务组合选择模型，从而在云资源供应商中选择中最符合用户需求的云服务组合路径。

### 3.3 面向批量服务请求的调度体系与模型

#### 3.3.1 调度体系

云平台服务调度策略要充分考虑云服务供应商的忙闲状态，整个服务请求队列的整体等待时间以及服务供应商的服务质量。而对多功能需求的服务请求进行其对应最优服务组合的选择也就是为了向服务请求提供 QoS 度量值最大的云服务组合。当一批云服务请求提交给云服务管理平台后，云平台不仅要考虑如何提供各个服务请求的最优服务组合，而且要协调处理整个服务请求队列的整体执行情况，预估整体执行时间以及实时的网络情况，并针对这些信息作出最优的调度方案。从下图可以看出，云服务管理平台的调度模块不仅连接了用户的服务请求(包括移动端和 PC 端)，而且还要对云服务资源供应商发出调度指令，同时还要监控云服务供应商与数据层的数据交换。而用户将服务请求提交给云平台之后，还需要提交相应的信息和对应的细节要求。而云服务管理平台正是根据这些服务请求的具体信息和要求调度云服务协调以满足各个服务请求的执行需求。



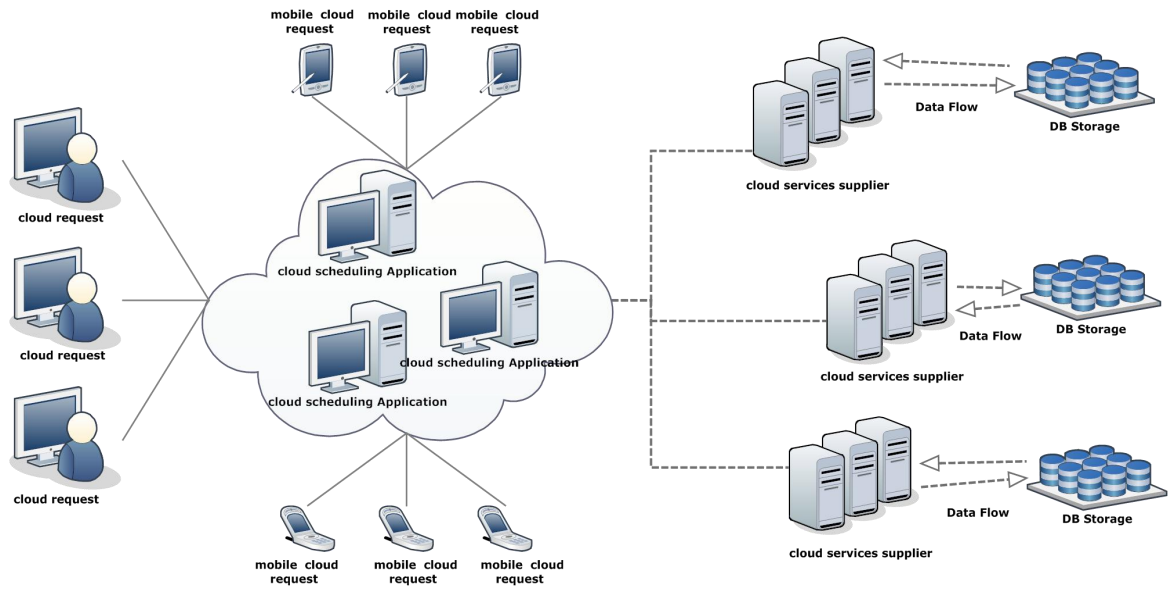


图 3-3 云服务调度体系结构图

云服务管理平台将各个资源供应商提供的服务分类整理，形成各类功能不同的云服务，每一类服务称为一个虚拟服务点。向云服务管理平台提交的多功能需求的服务请求进行逻辑分割后形成的各个子请求，都能在云服务平台的各类虚拟服务点找到对应的服务。通过第一阶段基于 QoS 的最优云服务组合选择模型与算法，可以为每个提交的服务请求找到连接各个虚拟服务点的最优路径。也就是说，每个服务请求划分后形成的子请求都有对应的虚拟服务点中的云服务。每个服务请求所划分出的子请求的位置为子请求点，子请求点与对应的虚拟服务点以及虚拟服务点中的云服务呈一一对应的关系。当云服务管理平台处理一批类型相似的云服务请求时，需要按照一定的策略调度相应的服务资源完成对应的子请求。

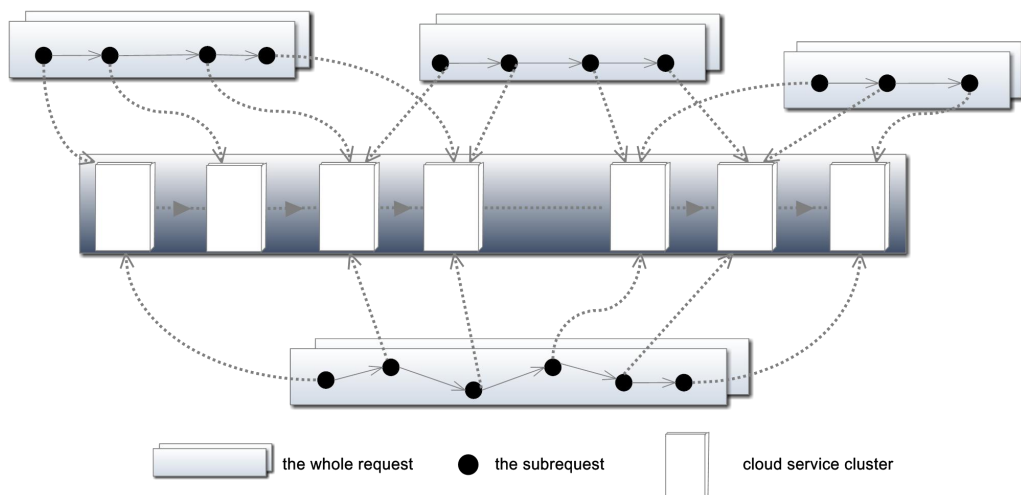


图 3-4 子服务请求与云服务对应关系

用户提出的复杂服务需求需要一系列功能单一的服务形成的服务组合来完成,当执行到该服务请求的某一请求点时,若其对应的单一云服务处于“忙时”,则需要等待。当一批用户同时向云服务管理平台提出请求时,极有可能用造成部分用户同时请求同一种服务资源,所以需要对服务资源进行合理调度,减少等待时间,使得一批次的服务请求能在最短的总时间内完成。

### 3.3.2 调度模型

针对批量云服务请求的服务调度策略主要的优化目标是减少批量服务请求的执行总时间,而调度的前提条件是每个服务请求必须使用其对应的最优服务组合。在云服务调度模型的建立过程中,还必须要考虑云服务切换的时间开销问题。本节将针对两方面的内容展开描述:

#### 1. 考虑服务切换时间开销

云服务管理系统集中协调控制各个单一功能的云服务,从而形成云服务执行链。而不同云服务供应商之间进行服务切换的时间开销则是依赖于数据传输时间开销以及网络环境,文献[15]中也提出了服务组合过程中的动态服务切换的问题,并描述了其对服务组合的影响。当然,不同种类的云服务之间进行切换的时间开销一定程度影响了批量云服务请求完成的时间和成本。

服务切换时间矩阵  $\log_{n \times n}$  记录了虚拟服务节点提供的服务之间的切换时间开销,其中  $\log_{ij}$  代表了虚拟服务点  $i$  到虚拟服务点  $j$  的服务切换时间开销,由于服务完成子请求的顺序是单向的,所以服务切换时间矩阵是一个右上角矩阵,左下部分及对称轴为 0。两两虚拟服务节点之间服务切换时间一定程度上影响了服务请求完成的总时间。若对于某一服务请求而言,云服务执行链中某一虚拟服务点是不需要的,那么对应的服务切换时间即为上一个云服务执行时间不为 0 的虚拟服务点到下一道云服务执行时间不为 0 的服务点之间的切换时间。

#### 2. 总执行时间最小化

假设云平台有  $n$  个服务点组成了服务序列,每道服务序列可以提供一类云服务以供调度。本文提出的第一阶段:基于 QoS 的云服务组合模型与算法,为各个复杂的云服务请求在服务序列中选取了其最优的服务组合路径。假设现有  $m$  个服务请求等待云服务管理平台处理,经过第一阶段最优云服务组合的选取之后可以确定出多维数组  $TimeTable_{m \times n}$ ,其记录了每个服务请求的各个子请求在每个服务点上的时间开销。当确定了完成该服务请

求的最优服务路径后, 每个服务点上就是该子请求对应的单一功能的云服务。  $TimeTable_{ij}$  表示第  $j$  个服务点对应的云服务  $S_j$  完成第  $i$  号服务请求对应的子请求所花费的时间开销。

云服务管理平台按照服务请求顺序  $JQ = \{JQ_1, JQ_2, JQ_3, \dots, JQ_n\}$  来调度云服务以完成各个服务请求的执行。  $StartTime_{rs_k^i}$  与  $EndTime_{rs_k^i}$  分别表示第  $k$  个虚拟服务点的对应服务  $S_k$  执行服务请求  $JQ_i$  的开始时间和结束时间。假设每个服务点每个时刻只满足一个服务请求, 那么存在以下的约束关系:

第  $k$  个虚拟服务点  $S_k$  提供的服务执行服务请求  $JQ_i$  后, 需要由服务点  $S_{k+1}$  继续提供对应的云服务以供  $JQ_i$  执行。此时虚拟服务点  $S_{k+1}$  是否执行请求  $JQ_i$  取决于  $S_{k+1}$  上的云服务现在是否在执行其他云服务请求的子请求。而云服务管理平台按顺序数组  $JQ$  来调度云服务完成服务请求, 所以如果虚拟服务点  $S_{k+1}$  上仍在执行其他子服务请求, 那么必然是  $JQ_{i-1}$ 。所以在执行虚拟服务点  $S_{k+1}$  上提供的云服务执行服务请求  $JQ_i$  的起始时间是在  $S_k$  上服务执行请求  $JQ_i$  的终止时间与在  $S_{k+1}$  上执行  $JQ_{i-1}$  终止时间的较大值。

同时考虑到服务切换时间, 我们需要考虑以下两种情况:

- (1) 对于服务请求  $JQ_i$ , 若第  $k$  个虚拟服务点提供的云服务执行的  $JQ_i$  时间为 0, 即其不需要该服务。那么  $StartTime_{rs_k^i} = EndTime_{rs_k^i} = EndTime_{rs_{k-1}^i}$ 。这是因为服务请求  $JQ_i$  不需要第  $k$  个虚拟服务点提供服务, 那么服务请求  $JQ_i$  在第  $k$  个虚拟服务点的开始时间就是其在上一个虚拟服务点的结束时间, 又因为在该虚拟服务点的执行时间为 0, 所以服务请求  $JQ_i$  在第  $k$  虚拟服务点的开始时间也就是其结束时间。
- (2) 对于服务请求  $JQ_i$ , 若它在第  $k$  个虚拟服务点的执行时间不为 0, 即其需要该虚拟服务点提供云服务。那么服务请求  $JQ_i$  在第  $k$  个虚拟服务点的开始时间, 就应该是第  $k$  个虚拟服务点前面一个执行时间不为 0 的虚拟服务点执行结束时间加上该虚拟服务点切换到第  $k$  个虚拟服务点的切换时间。假设  $d_k^i = lastproNotZero(k, i)$  表示第  $k$  个虚拟服务点工序前面一个执行时间不为 0 的虚拟服务点。那么综合以上两种情况, 有以下的约束关系:

$$StartTime_{rs_k^i} = \begin{cases} EndTime_{rs_{k-1}^i}, & TimeTable_{ij} = 0 \\ \max(EndTime_{rs_{k-1}^i} + \log d_k^i, EndTime_{rs_{k+1}^{i-1}}), & TimeTable_{ij} \neq 0 \end{cases} \quad (3-7)$$

$$EndTime_{rs_k^i} = \begin{cases} StartTime_{rs_k^i}, & TimeTable_{ij} = 0 \\ StartTime_{rs_k^i} + TimeTable_{ij}, & TimeTable_{ij} \neq 0 \end{cases} \quad (3-8)$$

所以综上所述可以看出, 面向批量服务请求的云服务调度问题可以描述为: 在云服务资源有限的情况下, 确定调度云服务执行服务请求的顺序, 使得服务请求整体执行时间最少。而对于确定调度方案的办法, 本文采取的是利用演化算法在批量服务请求执行前预计算各

种调度方案的总执行时间，从中取得最优的调度方案。而选取最优调度方案的过程属于有策略枚举，人工智能中的演化算法就能很好的解决这一类问题。

### 3.容错机制

云服务管理平台调度云服务处理批量云服务请求执行的过程中，可能会遇到某一云服务请求执行出现差错，从而影响后续服务请求的执行，所以建立合适的容错机制有利于云服务管理平台的高效运转，同时也会提高批量云服务的执行效率。本文提出的模型中，可能会出现云服务请求执行过程中因为网络等原因造成的云服务请求执行中断，由于云服务资源的有限性，这种情况可能造成后续服务请求的停滞。所以，云服务管理平台在批量云服务请求执行前需要确认每个服务请求的执行的额外容忍时间，即该服务请求在每个服务点不能按时执行完毕的情况下，最多能容忍其执行的额外时间。 $TolerateTimeTable_{m \times n}$ 记录了批量服务请求的各个子请求在每个服务点上的容忍时间开销。下面将分三种情况具体分析容错机制：

(1)如果云服务请求在每个单一功能的服务点的执行时间都是在正常的执行时间内完成的话，属于正常情况，容错机制不会启动。

(2)如果云服务请求受其他因素影响并没有在规定时间内完成，但是能在 $TolerateTimeTable_{m \times n}$ 中记录的对应的额外容忍时间内完成的话，容错机制启动，并通过云服务管理平台通知在该服务点的下一个云服务请求继续等待。

(3)如果云服务请求在容忍时间内都没有在该服务点完成执行，那么云服务管理平台暂时停止该服务请求的执行，转而顺延进行后续的服务请求的执行。待其他所有的服务请求执行结束后，再进行该服务请求的执行。

容错机制通过引入额外容忍时间，并且将一些影响执行整体时间的云服务请求滞后执行，使得云服务请求执行的模型更加适应实际情况，并且提高了执行的效率。

## 3.4 基于两阶段的云服务组合与调度的优化

### 3.4.1 云服务组合与调度体系结构与流程

云服务管理平台处理互联网用户的批量服务请求是由多个服务进程协同完成。而整个服务组合和调度的过程是基于用户服务请求提交所驱动的，而云服务管理平台作为控制中枢对服务请求就行服务组合的选择以及请求任务在执行过程中服务的调度，这是云平台主要的两项主要工作。传统对云服务组合的研究仅限于基于 QoS 对单一的云服务进行合成以及通过语义信息匹配等方式进行服务的匹配选择，而忽略了批量用户请求云服务平台所

造成服务资源调度的问题。为了保证每个服务请求都能获取最优的服务组合，云服务平台首先将服务请求序列中的服务请求信息根据云平台的服务供应商信息一一进行数据运算，得到每个服务请求的最优服务路径。之后各个服务请求的最优服务组合路径会传输给云服务平台的调度中心，调度中心根据服务请求的具体信息以及最优服务组合，制定服务资源调度方案，最后完成服务资源的调度，从而使得各个服务请求在保证使用最优服务组合的前提下，总的服务请求序列的执行时间能达到最少。因此，第一阶段的最优服务组合选择过程是为了提高服务请求的执行质量，第二阶段的服务调度方案是为了提高执行的效率，而第二阶段调度的过程中为了保证服务请求执行的质量，使用的云服务组合是第一阶段选择的，所以两个阶段模型衔接就在于：第一阶段为各个服务请求选择出的最优服务组合的响应时间属性的度量值存储在  $TimeTable_{ij}$  中，第二阶段的调度模型就是采用  $TimeTable_{ij}$  作为输入源。本文提出的两阶段优化模型的优势在于两阶段的模型可以区分开且耦合度不大，可以根据实际问题进行模型的更新与重组，对于实际问题的适用度也大大提升。

以用户的服务请求为驱动的云服务组合与调度系统整个执行过程可以理解为：以基于 QoS 的最优服务组合选择模块以及服务资源调度模块组成的云服务管理平台的自动化处理进程。这种云服务平台对服务组合的处理方式不仅能够对用户的云服务请求的局部要求有敏锐的反馈处理，并做到根据用户的需求随时调整模型的比重参数以选择最适合用户的服务组合，同时还能做到解决批量用户的服务请求造成服务资源冲突这一问题。由此我们还可以看出，整个服务组合与调度系统是一个以云服务平台为中心的环状系统，也就是说云平台收集服务供应商资源信息，用户的服务请求信息，并将其做最优化且最有效的“请求-服务”匹配，并且完成总时间最短这一高效性的要求。

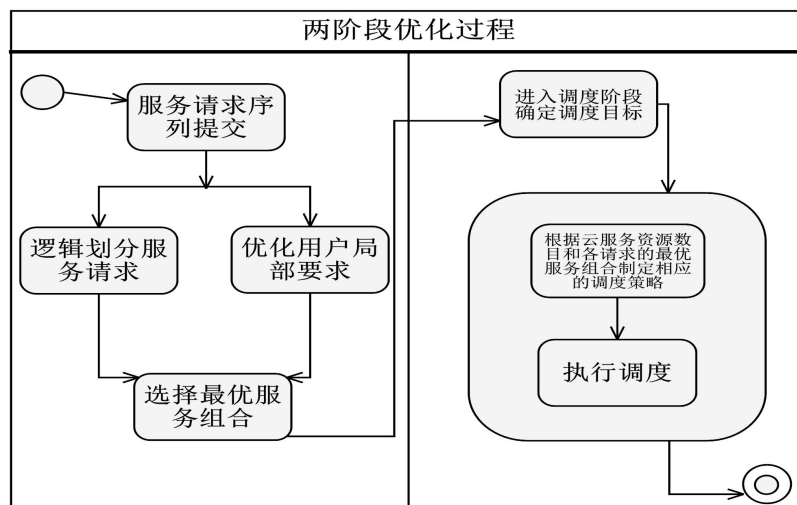


图 3-5 两阶段优化流程图

为了完成两阶段的云服务组合与调度，需要以下几个步骤：(1)可以通过本体描述语

言(owl-s)描述用户对服务组合的细节要求和整体目标,并且提交云服务管理平台,从而形成云服务请求序列。(2)云服务管理平台根据各个服务请求的具体信息,将服务请求进行逻辑分割,形成子服务请求流,并且为子请求进行服务粗选。(3)基于 QoS 为服务请求选择最优服务组合路径,采用的是第三章所描述的最优服务组合选择模型。(4)云服务管理平台根据服务请求序列中的各个请求对应的服务组合路径通过服务调度模型制定调度计划,进入调度准备阶段。(5)进入服务资源调度阶段。

### 3.4.2 多目标优化

传统的研究处理云服务请求一般只是为服务请求选择其最优路径,而并未考虑实际问题中批量的服务请求等待处理时应该考虑的多个具体问题。在这一过程中,也存在多个可优化的目标。首先考虑云服务组合阶段,为每个服务请求选择其最优的服务组合是主要的目标条件。由于本文对云服务或者云服务组合的评价标准是基于 QoS 度量值的,所以云服务组合阶段的主要目标也可以看成是为服务请求选择 QoS 度量值最大的服务组合,并且每个用户对服务可能有某些局部优化目标。然后考虑云服务资源调度阶段,对于云服务管理平台而言,每一时刻不可能只有一个服务请求需要处理,常常需要面对的是多个服务请求同时需要处理,所以调度阶段的总目标是在面向批量服务请求的云服务资源调度中总的执行时间最少。当然调度的前提是每个服务请求都使用为其选择出的最优服务组合。所以说调度的前提是第一阶段的最优云服务组合选择的。在整个云服务组合与调度过程中,各个优化目标以及之间的依赖关系由下面的图表表示出来:

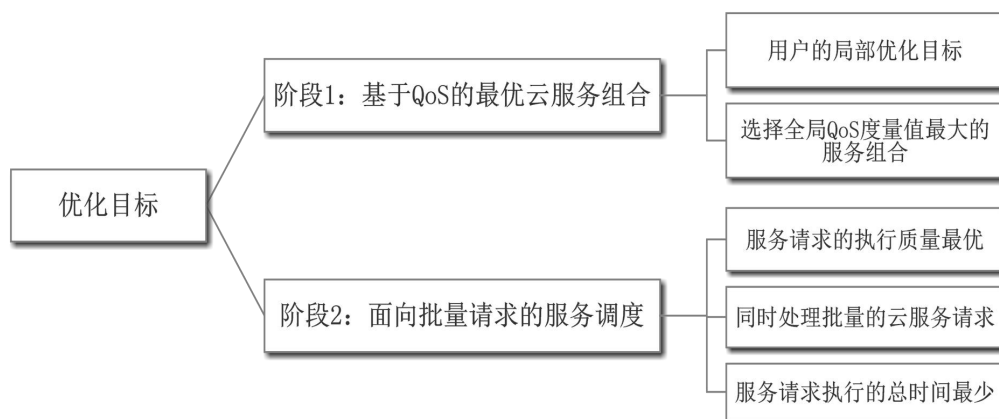


图 3-6 优化目标关系图

## 3.5 本章小结

本章针对批量云服务请求同时到达时,云服务管理平台处理中主要的两个过程进行研究:(1)为每个服务请求选择基于服务质量的最优云服务组合(2)基于批量云服务请求调度云服务。同时给出了这两个阶段的体系结构,在此基础上给出了对应的优化模型与两个阶段相互关联关系。同时介绍了两阶段优化模型对批量服务请求执行的效率和质量的优化。最后本章说明了在这两个过程中可优化的多个目标。本章充实了本文模型建立方面的内容,提出了云服务组合与调度本质上是多目标优化问题,而下一章要提出的改进粒子群算法将用于求解该多目标优化问题。

## 第4章 基于改进粒子群算法的云服务组合与调度

### 4.1 引言

本文提出的两阶段优化的云服务组合与调度的问题本质上属于多目标优化问题,在第三章中我们将模型的优化目标和限制条件用数学表达式的方式展现了出来,而针对这种多目标优化数学模型的求解,人工智能领域里使用的演化算法是常常被考虑的求解方法。在本章中,我们首先会介绍经典的粒子群优化算法,然后在基础上引入三种较为常见的改进方式。最后针对本文提出的实际问题,介绍改进的粒子群优化算法(Modified Particle Swarm Optimization,MPSO)在云服务组合与调度问题中如何应用。

### 4.2 基础粒子群算法简介

粒子群优化算法(Particle Swarm Optimization,PSO)是一种启发式的自适应算法,最早由 Kennedy 和 Eberhart 在 1995 年提出<sup>[61]</sup>,其基本原理起源于模仿动物的觅食行为。粒子群优化算法能解决较多数值优化问题,同时也被证明优于大多数传统演化算法。粒子群优化问题已经被证明可以运用于大多数的工程应用和数学问题,而粒子群优化算法也因其较为简单的思路和灵活的应用方式越来越来受到大多数科研人员的青睐。粒子群优化算法的核心是通过粒子的不断更新来寻求最佳优质的解,而每个粒子均具有速度和位置两个属性。粒子群优化算法的思路:首先初始化若干粒子,由它们形成一个粒子群,并且每个粒子的位置在合理范围内随机分配,而一个粒子群中的各个粒子的位置代表该问题的一个解。每个粒子开始移动,不断地寻找自己的最优位置,若发现其他位置优于现在位置,即“飞向”其他位置,种群中的各个粒子均在不断地将当前位置与历史最优位置进行比较,当比较一定的次数(达到设定的最大迭代次数)或者粒子的位置超过合理范围,即算法结束。

粒子每次迭代的过程中都会不断地调整自身的速度和位置,不断地向最优位置移动。在算法进行的过程中,需要有变量记录每个粒子的历史最优位置以及整个粒子群中的最优位置,以便在迭代过程中更新最优位置。粒子的位置是根据具体问题的适应度函数计算得出的,也就是说粒子的位置就是当前粒子代表的目标问题的适应度值。为了避免传统演化



算法常出现的再算法后期收敛速度过快的问题，一个较优的速度位置更新方式是必须的。

下面的公式将简明地对传统的速度及位置更新公式进行一个描述：

$$v_i^{k+1} = \omega v_i^k + c_1 * r_1 * (pbest_i - x_i^k) + c_2 * r_2 * (gbest - x_i^k) \quad (4-1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (4-2)$$

符号描述：

$v_i^k$  第 k 次迭代中粒子第 i 维的速度

$\omega$  惯性权重

$c_i$  加速度系数

$r_i$  取值在[0, 1]范围内的随机数

$x_i^k$  第 k 次迭代中粒子第 i 维的位置

$pbest_i$  第 i 维的局部最优值

$gbest$  全局最优值（位置最好的粒子）

粒子群优化算法速度更新公式中主要有局部最优值(当前粒子的历史最优值)和全局最优值(所有粒子中的最优值)，并且通过随机数调整两者的比重。因为粒子群优化算法进行的是粒子向更加优秀粒子靠近的过程，而靠近的方式分为向自己历史最优位置和种群最优位置两种。粒子下一次的位置由当前位置和更新后的速度决定，因此更新速度决定了下一步粒子的位置。由上面粒子更新速度公式可以看出，粒子“有意识”地向局部最优位置和全局最优位置的方向靠近。

粒子群算法也存在一定的问题，如在一些问题的求解过程中，不可避免的存在离散整数解空间的限制，而将由 PSO 所求得连续非整数解映射到相应的整数解过程中，可能会造成算法解的精度降低。多年来有很多对 PSO 算法的改进，比如文献[62]提出了一种新的粒子更新方法。文献[63]则提出了一种更简化而高效的粒子群优化算法，文中将一般使用的 PSO 称为 bPSO(基本粒子群优化算法)，并且证明了 bPSO 为二阶微分方程，在此基础上，简化了粒子的更新速度的过程，将 bPSO 进一步化简为一阶微分方程，即仅由粒子的位置来控制种群的变化过程。作者将这种算法称为简化粒子群优化算法(sPSO)，并证明了该算法具有更优的收敛性和解的精度。而众多的研究表明， $\omega$  的取值将会直接决定 PSO 算法的收敛速度。文献[64]提出了将基本粒子群算法优化后所得到的 BNPSO 算法，并证明该算法对目标优化问题的求解效率更高。文献[65]改进了基础粒子群优化算法，提出了

新的全局——局部解法，其中 $\omega$ 是一个变量，随着迭代的不断进行而改变。文献[66]则证明当 $\omega$ 在 0.8 与 1.2 之间时，算法将更快的得到最优值。在本文中，惯性权重 $\omega$ 的取值为 1.2，当不对学习因子 $c_1, c_2$ 进行改变时， $c_1, c_2$ 均取值为 0.8。

具体算法描述如下：

---

#### 粒子群优化算法（Particle Swarm Optimization）

---

- 1 随机初始化粒子群，注意设置合适的权重因子和自学习因子。
  - 2 通过具体问题的目标函数(即适用度函数)计算粒子的适应度值。如果适应度值大于当前的局部最优值  $pbest_i$ ，则将该适应度值置为局部最优值。如果适应度值大于当前的全局最优值  $gbest$ ，则将该适应度值置为全局最优值。
  - 3 对粒子群中的每个粒子按照步骤 3 进行计算，并且随时更新全局最优粒子。
  - 4 按照速度以及位置更新公式更新每个粒子的位置和速度。
  - 5 执行步骤 2—4，直到达到算法停止条件（达到最大迭代次数）。
- 

### 4.3 粒子群优化算法的改进

#### 1. 速度更新公式改进

由于云计算环境下的服务组合节点数目大，供应商多，存在着大量的组合方案以供选择。如果采用基础演化算法容易陷入局部最优，在求解服务组合优化问题的过程中也会出现大量的劣质解包围最优解的情况<sup>[43]</sup>，因此需要改进原有算法，提高粒子种群的多样性，避免过早出现“早熟”。粒子群算法及其改进算法中，速度的更新公式一直是重要的研究内容，同时改进速度更新公式也有助于丰富服务组合解集的多样性。本文提出的改进粒子群算法(MPSO)改进了传统粒子群算法中的速度更新公式，增强了全局最优搜索能力并且提高算法跳出局部最优的能力，进一步扩大粒子种群的候选范围，从而避免非最优解的大量累积。MPSO 算法借鉴了文献[67]中的改进方法，将二阶振荡环节加入到粒子速度更新公式。改进后的算法在求解粒子第  $k+1$  次迭代速度的公式中，加入了第  $k-1$  次迭代时该粒子的速度信息。充分利用第  $k-1$  代速度信息可以加强粒子的自学习能力，同时丰富种群的多样性。改进后粒子的速度与位置更新公式如下<sup>[67]</sup>：

$$v_{i-j}^{k+1} = \omega v_{i-j}^k + \beta_1(pbest_i - (1 + \varepsilon_1)x_{i-j}^k - \varepsilon_1 x_{i-j}^{k-1}) + \beta_2(gbest - (1 + \varepsilon_2)x_{i-j}^k + \varepsilon_2 x_{i-j}^{k-1}) \quad (4-3)$$

$$x_{i-j}^{k+1} = x_{i-j}^k + v_{i-j}^{k+1} \quad (4-4)$$

其中  $\beta_1 = c_1 r_1$  以及  $\beta_2 = c_2 r_2$ 。 $\omega$  与  $c_i$  分别代表了惯性权重和学习因子， $r_i$  是取值在  $[0, 1]$

范围内的随机数。其中  $x_{i,j}^k$  和  $v_{i,j}^k$  为第  $k$  次迭代中第  $i$  个粒子  $P_i$  的第  $j$  维方向的位置和速度。

文献[67]证明了在迭代过程的前期，即迭代次数  $k < G_{\max}/2$  时，当  $\varepsilon_1 < \frac{2\sqrt{\beta_1}-1}{\beta_1}$  且  $\varepsilon_2 < \frac{2\sqrt{\beta_2}-1}{\beta_2}$ ，算法具有较强的搜索最优解的能力。迭代次数  $k \geq G_{\max}/2$  时， $\varepsilon_1 \geq \frac{2\sqrt{\beta_1}-1}{\beta_1}$  且  $\varepsilon_2 \geq \frac{2\sqrt{\beta_2}-1}{\beta_2}$ ，算法收敛速度较快。

为了获取更快的收敛速度以及加强算法的全局搜索能力，本文中提出的改进杂交粒子群优化算法的取值借鉴文献[67]的结论，如下：

$$\varepsilon_1 = \begin{cases} \frac{2\sqrt{\beta_1}-1}{\beta_1} = \frac{2\sqrt{c_1 r_1}-1}{c_1 r_1}, & k < G_{\max}/2 \\ \frac{(2\sqrt{\beta_1}-1)(1+r_3)}{\beta_1} = \frac{(2\sqrt{c_1 r_1}-1)(1+r_3)}{c_1 r_1}, & k \geq G_{\max}/2 \end{cases} \quad (4-5)$$

$$\varepsilon_2 = \begin{cases} \frac{2\sqrt{\beta_2}-1}{\beta_2} = \frac{2\sqrt{c_2 r_2}-1}{c_2 r_2}, & k < G_{\max}/2 \\ \frac{(2\sqrt{\beta_2}-1)(1+r_4)}{\beta_2} = \frac{(2\sqrt{c_2 r_2}-1)(1+r_4)}{c_2 r_2}, & k \geq G_{\max}/2 \end{cases} \quad (4-6)$$

( $r_3, r_4$  为(0, 1)范围内的随机数)

## 2. 动态参数机制

云环境的动态不确定性使得云服务组合问题更加复杂。特别是针对部分非线性的服务组合问题，基础粒子群优化算法和其他的一些粒子群改进算法由于过于依赖定值的学习因子参数，导致局部范围内粒子的速度和位置的有规律线性变化，从而造成求解过程中较早进入局部最优，所以较难很好地适应该类问题。同时，定值的学习因子也限制了算法搜索最优解的能力，造成种群多样性降低，这样也会导致服务组合解集中劣质解数量的增加。文献[43]基于三角函数对学习因子做了非线性动态变化的设计，通过将  $\cos^2(\rho_1 \times k / G_{\max})$  和  $\sin^2(\rho_1 \times k / G_{\max})$  分别加入到学习因子  $c_1$  和  $c_2$  中( $\rho_1$  为定值权重)，强化了粒子的全局搜索能力。MPSO 在文献[43]的基础上添加权重因子  $\alpha_1$  与  $\alpha_2$  控制三角变换对学习因子  $c_1$  和  $c_2$  的影响，同时根据三角函数的基本转换公式，可以得出以下的推倒公式：

$$\alpha_1 \cos^2(\rho_1 \times k / G_{\max}) = \alpha_1 \left( \frac{\cos 2(\rho_1 \times k / G_{\max}) + 1}{2} \right) = \frac{\alpha_1}{2} \cos(2\rho_1 \times k / G_{\max}) + \frac{\alpha_1}{2} \quad (4-7)$$

$$\alpha_2 \sin^2(\rho_1 \times k / G_{\max}) = \alpha_2 \left( \frac{1 - \cos 2(\rho_1 \times k / G_{\max})}{2} \right) = -\frac{\alpha_2}{2} \cos(2\rho_1 \times k / G_{\max}) + \frac{\alpha_2}{2} \quad (4-8)$$

所以 MPSO 算法的动态学习因子表达式的推倒过程如下：

$$c_1 = c_1' + \alpha_1 \cos^2(\rho_1 \times k / G_{\max}) = c_1' + \frac{\alpha_1}{2} \cos(2\rho_1 \times k / G_{\max}) + \frac{\alpha_1}{2} = \beta_1' + \beta_1 \cos(\rho_2 \times k / G_{\max}) \quad (4-9)$$

其中,  $\beta_1' = c_1' + \frac{\alpha_1}{2}$ ,  $\beta_1 = \frac{\alpha_1}{2}$ ,  $\rho_2 = 2\rho_1$ 。

$$c_2 = c_2' + \alpha_2 \sin^2(\rho_1 \times k / G_{\max}) = c_2' - \frac{\alpha_2}{2} \cos(2\rho_1 \times k / G_{\max}) + \frac{\alpha_2}{2} = \beta_2' + \beta_2 \cos(\rho_2 \times k / G_{\max}) \quad (4-10)$$

其中,  $\beta_2' = c_2' + \frac{\alpha_2}{2}$ ,  $\beta_2 = -\frac{\alpha_2}{2}$ ,  $\rho_2 = 2\rho_1$ 。

(其中  $k$  代表当前迭代次数,  $G_{\max}$  代表最大的迭代次数,  $c_1'$  与  $c_2'$  是改进前的定值学习因子)

由于将学习因子  $c_1$  和  $c_2$  设置为动态变换后, 每次迭代过程中, 都要对  $c_1$  和  $c_2$  重新计算, 若采用文献[43]的三角变换改进办法, 每次迭代过程中, 还要运算两次平方计算。而通过三角变换后, MPSO 在每次迭代过程中处理动态学习因子时, 无需进行平方, 减少了算法运算的复杂度和时间。MPSO 算法对学习因子做了动态非定值设定, 减少了算法对系数的依赖。其中对学习因子设定非定值变化可以提高算法的全局搜索能力以及扩宽算法适应问题的范围。

### 3. 杂交策略

正如上文所提到的, 在实际的云计算环境下, 不同用户对局部 QoS 属性的关注度不一致, 所以针对不同的用户需求, 需要调整相应的权重因子甚至调整目标函数或者局部限制条件。因此用于解决这种目标函数常变的优化问题的算法, 需要具有不依赖于问题模型这样的特点。简单遗传算法(SGA)的杂交思想满足这样的要求。MPSO 的杂交策略来源于 Lovbjerg<sup>[68]</sup>将传统遗传算法的思想加入基础粒子群优化算法的这种思路。粒子群中的每个粒子以一定的概率  $P$  (杂交概率) 进入杂交池, 并且杂交池中的粒子两两进行杂交。每一对粒子杂交后产生两个子代, 所以杂交完成后, 父代与子代数相同。由于每个粒子都是由二维矢量来表示, 所以杂交过程中, 需要对一对粒子的速度和位置分别进行杂交。杂交结束后, 由子代取代父代进入后续的迭代过程。Lovbjerg<sup>[68]</sup>提出了两个粒子进行杂交所产生的子代的位置和速度的公式:

$$child_1(x_i) = P_i \times parent_1(x_i) + (1 - P_i) \times parent_2(x_i) \quad (4-11)$$

$$child_2(x_i) = P_i \times parent_2(x_i) + (1 - P_i) \times parent_1(x_i) \quad (4-12)$$

$$child_1(v_i) = \frac{parent_1(v_i) + parent_2(v_i)}{|parent_1(v_i) + parent_2(v_i)|} \times parent_1(v_i) \quad (4-13)$$

$$child_2(v_i) = \frac{parent_1(v_i) + parent_2(v_i)}{|parent_1(v_i) + parent_2(v_i)|} \times parent_2(v_i) \quad (4-14)$$

MPSO 算法不仅通过杂交池中的粒子两两杂交扩宽了种群的范围,同时在父代夹杂产生子代的过程中,不需要依靠适应度函数,使得这种算法可以更加适应于多类型的最优值求解问题。因此,融入了杂交策略的 MPSO 算法应用于限制条件较多,问题模型常变的云服务组合问题求解中有较好的效果。

算法描述:

---

#### 改进的粒子群优化算法 (MPSO)

---

- 1 初始化粒子群及相关参数。确定粒子个数,随机生成每个粒子的位置和速度。设定最大迭代数  $G_{\max}$ , 学习因子  $c_1$ ,  $c_2$  的初始值以及杂交概率。确定适应度函数。
  - 2 计算步骤 1 中随机生成的各个粒子的适应度值。将所有粒子适应度值中的最大值赋值于  $gbest$ 。对应将粒子  $P_i$  的适应度值赋值给  $pbest_i$ 。
  - 3 开始迭代演化过程。首先计算出本次迭代过程中的学习因子  $c_1$ ,  $c_2$  的取值。用当前  $c_1$  和  $c_2$  值更新粒子的速度以及位置。更新粒子  $i$  的当前最优值  $pbest_i$  以及全局最优值  $gbest$ 。
  - 4 根据杂交概率  $P$  选定进入杂交池的粒子数目。进入杂交池的粒子随机两两杂交,按杂交公式计算产生的后代粒子的速度和位置,产生的后代粒子的数目不变。
  - 5 若达到最大迭代数,则算法结束;否则转到步骤 3。
- 

## 4.4 应用于两阶段优化的改进粒子群算法

在上一章已经提到,两阶段的云服务组合与调度有各自的模型,且属于多目标优化问题,而两者的衔接仅在于数组  $TimeTable_{m \times n}$ 。这样不仅减少了耦合度,而且便于各自进行模型的更新,以便适应于更多的实际问题。为了解决两个阶段提出的两个多目标优化问题,本文将改进的粒子群优化算法应用于问题的求解。粒子群算法及其改进系列算法应用于实际问题中时,主要需要解决定义粒子含义和适应度函数这两方面的问题,本节将针对这两方面进行阐述:

### 4.4.1 定义粒子的含义

将粒子群优化算法运用于具体问题的求解中的关键步骤之一就是针对具体问题设计相应的粒子。将粒子群优化算法运用到云服务组合与调度过程中,可以将每个粒子看做是一组解。由于云服务组合与调度问题本质上是一个多目标优化问题,解决这一问题的办法是有规律地穷举。粒子群算法可以通过不断迭代过程中粒子飞行位置的局部最优值和全局最优值确定解的最优值。同时粒子如何构成就取决于具体的问题。而粒子群的一大优势就

在于能够根据具体问题的不同设计不同含义的粒子,当然粒子设计的合适与否决定了问题求解的便利程度,甚至解的质量。

在本文的服务组合与调度问题的模型构建是属于多目标优化的问题构建,即有多个局部限制条件下求目标函数的最优值。而针对服务组合与调度的两个目标函数和多个局部限制,采用优化算法进行有条件的穷举。所以说算法只是解决模型提出的目标问题的工具。本文中需要解决的基于 QoS 的云服务组合与面向复杂请求的云服务资源调度问题是属于两种优化目标,即需要解决的是两个目标函数。而在目标问题中一直存在的对应关系是子服务请求与子服务资源的对应关系。所以说粒子的构建也依赖于这种关系。本文提出的两阶段的优化模型可以由改进的粒子群优化算法解决。但是基于 QoS 的云服务组合与面向批量服务请求的云服务调度是两个目标函数不同的问题,所以针对不同的问题有不同的粒子设计与含义。

### 1. 基于 QoS 云服务组合的粒子设计

在云平台环境下,若一个多功能服务请求经过逻辑划分后得到  $m$  个子请求,则这个服务请求使用的服务组合是一套由  $m$  个单一资源组合而成的。在优化算法的求解过程中,一套服务组合对应一个  $m$  维的粒子。一个粒子的位置和速度可以由  $m$  维向量  $x_i = \{x_{i\_1}, x_{i\_2}, x_{i\_3}, \dots, x_{i\_m}\}$  和  $v_i = \{v_{i\_1}, v_{i\_2}, v_{i\_3}, \dots, v_{i\_m}\}$  来表示,  $x_{i\_j}$  和  $v_{i\_j}$  分别表示粒子  $P_i$  第  $j$  维方向的位置和速度。其中粒子的位置  $x_{i\_j}$  在服务组合过程中表示子服务请求  $T_j$  选择的服务资源编号。下面的示例图表示了粒子与服务组合的对应关系。

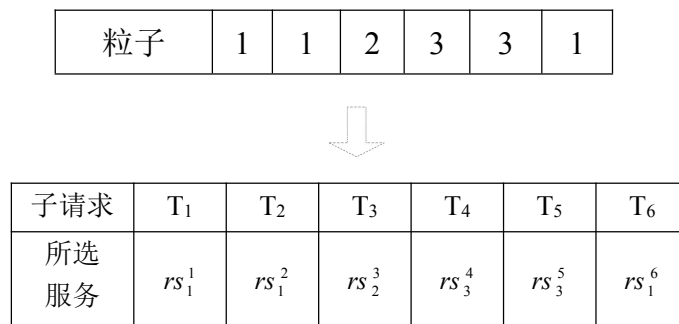


图 4-1 粒子与服务组合的对应关系

### 2. 云服务调度粒子设计

在调度云服务以满足云服务请求的多功能需求的情况下,若一个服务请求需要经过  $m$  个功能单一的子服务才能完成,则说明这个服务请求经过逻辑划分后可以得到  $m$  个子请求。所以在这种情况下,云服务管理平台调度云服务完成服务请求执行的顺序对应改进的粒子群算法的一个  $m$  维的粒子。正如上文所言,一个粒子的位置和速度可以由  $m$  维向量

$x_i = \{x_{i\_1}, x_{i\_2}, x_{i\_3}, \dots, x_{i\_m}\}$  和  $v_i = \{v_{i\_1}, v_{i\_2}, v_{i\_3}, \dots, v_{i\_m}\}$  来表示,  $x_{i\_j}$  和  $v_{i\_j}$  分别表示粒子  $P_i$  第  $j$  维方向的位置和速度。其中粒子的位置  $x_{i\_j}$  在服务调度过程中表示执行云服务请求  $T_j$  的顺序编号。下面的示例图表示了粒子与服务请求的对应关系。

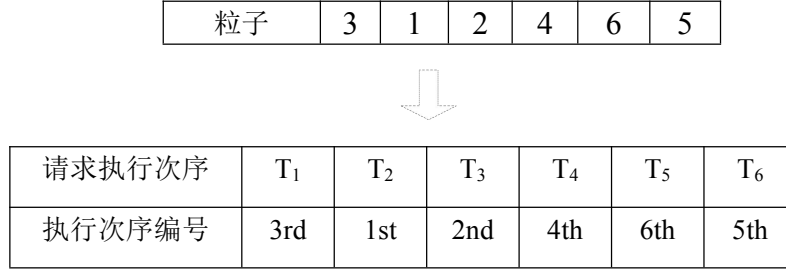


图 4-2 粒子与服务请求的对应关系

$Pbest_i$  和  $gbest$  分别代表了粒子  $P_i$  当前的最优位置和整个粒子群中的全局最优位置。  
 $Pbest_i = \{Pb_{i\_1}, Pb_{i\_2}, Pb_{i\_3}, \dots, Pb_{i\_m}\}$ ,  $Pb_{i\_j}$  代表粒子  $P_i$  在第  $j$  维方向上的最优位置。  
 $gbest = \{gb_1, gb_2, gb_3, \dots, gb_m\}$  表示子服务请求集合  $T = \{T_1, T_2, \dots, T_m\}$  所选择的最优服务组合与通过云服务调度完成服务请求的对应顺序。在第一次迭代过程中, 随机生成粒子  $P_i$ , 并将其赋值给  $Pbest_i$ , 同时将适应度函数值最大的粒子赋值给  $gbest$ 。之后的每次迭代过程中, 根据适应度函数更新  $Pbest_i$  与  $gbest$ 。当达到最大迭代次数后, 结束算法, 此时的  $gbest$  就是全局最优值。

#### 4.4.2 适应度函数

适应度函数在粒子群优化算法解决具体问题中具有极其重要的作用, 它是具体问题的数学模型在算法中的体现。不同的粒子可以看做是不同的解, 而将不同的解代入适应度函数得出对应的适应度函数值, 将其与之前的全局最优值对比或者更新。本文针对两阶段优化的具体问题提出了两个数学模型, 不同的目标函数对应于不同的适应度函数。

##### 1. 基于 QoS 的云服务组合适应度函数

假设集合  $Q$  是影响最优云服务组合选择过程的 QoS 属性集合, 即  $Q = \{QoS_1, QoS_2, \dots, QoS_k\}$  (在本文中分别是时间, 成本, 信誉度和可靠性四种 QoS 属性)。由于粒子  $P_i = \{P_{i\_1}, P_{i\_2}, \dots, P_{i\_m}\}$  代表了子服务请求集合  $T = \{T_1, T_2, \dots, T_m\}$  当前选择的一套服务组合, 所以在不受局部 QoS 限制的情况下, 适应度函数可以详细描述为:

$$fitness\ 1(P_i) = \sum_{j=1}^m \sum_{w=1}^k (\alpha_w \times QoS_w(rs_{P_{i\_j}}^j)) \quad (4-15)$$

其中  $QoS_w(rs_{P_{i-j}}^j)$  代表了子任务  $T_j$  使用服务资源  $P_{i-j}$  过程中属性  $QoS_w$  的度量值。

若当前的服务组合问题受到局部 QoS 的限制, 适应度函数描述如下:

$$fitness1(P_i) = \begin{cases} \sum_{j=1}^m \sum_{w=1}^k (\alpha_w \times QoS_w(rs_{P_{i-j}}^j)), & P_i \text{ 满足局部限制} \\ 0, & P_i \text{ 不满足局部限制} \end{cases} \quad (4-16)$$

## 2. 面向批量服务请求的云服务调度适应度函数

云服务管理平台调度服务资源以处理批量服务请求的模型已在第三章中提出, 从该模型不仅能获取整个服务请求执行完成的总时间, 而且还能获取每个服务请求的各个子请求执行开始和完成的具体时间点以及每个服务请求开始和最终完成的时间点。这些开始和完成的时间点都存在开始时间矩阵  $StartTime_{m \times n}$  和结束时间矩阵  $EndTime_{m \times n}$  中, 其中  $m$  和  $n$  分别表示云服务平台提供的云服务数目以及云服务请求的数目。根据上一节对粒子的设计, 云服务管理平台调度云服务完成服务请求执行的顺序对应改进的粒子群算法的一个  $m$  维的粒子  $P_i = \{P_{i-1}, P_{i-2}, \dots, P_{i-m}\}$ 。矩阵  $PT$  是第一个阶段得出的结果, 包括了各个服务请求对应的最优云服务的具体执行时间。 $PT_{ij}$  表示第  $j$  个服务点对应的云服务  $S_j$  完成第  $i$  号服务请求对应的子请求所花费的时间。将粒子群优化算法应用于第二阶段, 适应度函数的输入包括: (1) 云服务资源的数目 (2) 云服务请求的数目 (3) 由第一阶段基于 QoS 的云服务组合选择得出的各个服务请求对应的最优服务组合的执行时间矩阵。 (4) 代表解的粒子群  $P_i$ 。

根据上述结论, 一个  $m$  维粒子  $P_i = \{P_{i-1}, P_{i-2}, \dots, P_{i-m}\}$  对应的适应度函数运算过程如下:

---

```

fitness2(Pi,n,m,pt)
for k=1:n
    for l=1:m
        if pt(l, Pi-k) != 0
            indexLast1=findlastProNotZero(pt, l, Pi-k); % 上一个执行时间不为0的子请求。
            StartTime(l, k)=max(EndTime(l-1, k)+log(indexLast1, l), EndTime(l, k-1));
            EndTime(l, k)=StartTime(l, k)+pt(l, Pi-k);
        else
            EndTime(l, k)=EndTime(l-1, k);
            EndTime(l, k)=EndTime(l, k);
        end
    end
end
fitness2(Pi,n,m,pt) = EndTime(k,l)

```

---

一个粒子代表一组解, 通过第三章提出的调度总时间的计算模型计算出每个粒子的适应度值, 并且不断更新最优解, 所以服务请求执行总时间就是最后一个服务请求执行的结束时间。本文提出的基于改进粒子群优化算法解决多目标优化问题的优势在于可以提前进



行预运算,得出调度云服务执行云服务请求的最优方案,从而免去了主观调度策略分析不准确的风险。

## 4.5 本章小结

本文提出的云服务组合与调度模型本质上是多目标优化算法,本章主要讨论了基于改进的粒子群优化算法解决该问题。首先本章介绍了传统的基础粒子群优化算法,之后在此基础上,提出了三处可优化点对粒子群优化算法进行改进,最后结合第三章提出的两阶段优化的模型,分别讨论了改进粒子群优化算法在此应用背景下的粒子含义与适应度函数的设计。本章内容完善了全文的算法设计模块。

## 第 5 章 实验仿真及结果分析

### 5.1 引言

为了验证本文提出的基于改进粒子群算法的两阶段云服务组合与调度优化模型与算法的有效性,我们分别进行了两个阶段的仿真实验。其中为了准确仿真基于 QoS 的云服务组合问题,我们将该问题的求解仿真又分为这两种情况下的仿真:(1)无局部 QoS 限制(2)全局 QoS 与局部 QoS 相结合。同时为了说明云服务调度的模型与算法能够适应于实际问题的解决,在本章还进行了一个制造云服务请求执行的实例仿真。算法解决问题的运行时间对评价算法性能与预估实际问题的解决时间有着重要的作用,所以在本章的最后一个部分,我们对算法的执行时间进行了测试,得到了对应数据并进行了分析。本章的实验仿真部分在云计算仿真平台 cloudsim 和算法仿真平台 matlab 上完成。

### 5.2 基于 QoS 的云服务组合问题仿真

针对第一阶段基于 QoS 的云服务组合仿真,主要进行了(1)无局部 QoS 限制(2)全局 QoS 与局部 QoS 相结合(即有局部 QoS 限制)这两种情况下的仿真,并给出三种算法所求得的全局 QoS 最优值之间的对比。同时为了减少偶然情况对实验结果产生的影响,我们进行了多次模拟,并且针对不同的问题规模均进行了相应的算法对比。在本文的第三部分中详细描述了云服务的 6 种服务属性:响应时间,成本,信誉度,可靠性,可访问性和安全性,在本节中将选取前四种服务属性作为实验的 QoS 属性,同时随机生成度量范围为[1, 50]的各个候选云资源供应商的 QoS 属性值。实验过程中,假设 4 种 QoS 属性分别是时间,成本,信誉度和可靠性的权重因子分别取值为 0.3, 0.3, 0.2, 0.2。与本文提出的改进粒子群优化算法(MPSO)进行算法对比的是二阶振荡粒子群算法(OscPSO)<sup>[67]</sup>以及基础杂交粒子群(BreedPSO)<sup>[68]</sup>。这两种比较算法在粒子群改进中都算是较为优秀的改进方法,而且有着大量的应用。三种算法迭代次数和粒子数目均设置为 500 和 30。

#### 1. 无局部 QoS 限制的云服务组合

随着互联网技术的逐步发展,用户所提出的云服务请求的复杂度越来越高,划分出得子服务请求的数量也会越来越多,甚至一个事务的完成也会需要一个单一的云服务请求来完成。所以模型与算法能否适应较大规模的问题也是一个重要考察点。图 3, 图 4 与图 5

分别展示了子服务请求个数为 30, 100 和 180 情况下三种算法在无局部 QoS 限制时所求的全局最优 QoS 值的对比,三种不同的实验设置代表了三种不同的问题规模。这三种情况体现了本文提出的模型与算法解决不同规模问题的效果,同时全局 QoS 最优值的大小体现了选取最优服务组合的优劣。从下面三个图中可以看出,随着供应商数目增加的随机变化,MPSO 算法相对于其他两种算法,在选取最优云服务组合问题求解中具有一定的优势。从分析算法的角度上来讲,MPSO 具有较强的全局搜索能力,并且父代粒子杂交产生子代粒子的过程还进一步丰富了种群的多样性,同时 MPSO 算法在基础粒子群优化算法的基础上引入振荡环节不仅能增强全局搜索能力,还能避免算法过早进入局部最优。实验中,100 个子服务请求在进行云服务的选择和组合过程中供应商的个数范围是[5, 20],那么此时解空间的范围是 $[100^5, 100^{20}]$ 。180 个子服务请求选取最优服务组合过程中解空间范围最大可以达到  $180^{40}$ 。通过仿真实验证明,在子服务请求数目和资源供应商数目都较大的情况下,本文提出的选择最优服务组合的模型与算法也具有较好性能表现。在解空间范围比较大的情况下从大量劣质解中选择出最优解需要算法有较强的全局搜索能力,而基于 MPSO 的云服务组合算法通过改进速度更新公式等方式增强了全局搜索能力。以上结果说明,本文提出的基于 QoS 的云服务组合优选方法能适合用户提出的多功能需求的云服务请求。

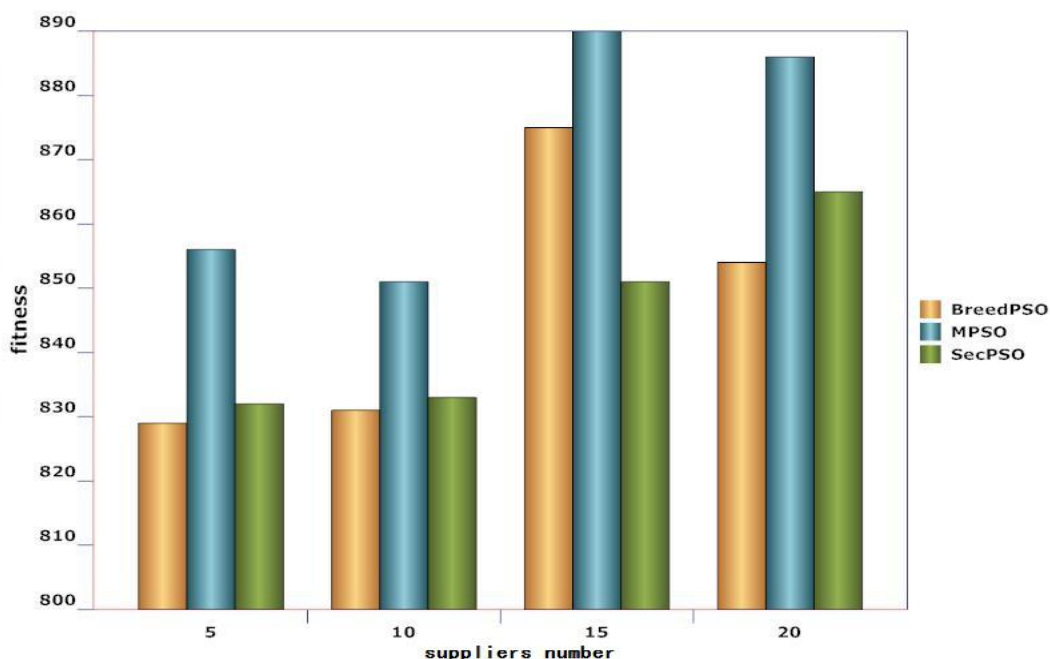


图 5-1a 子服务请求数 30, 供应商数分别为 5, 10, 15, 20

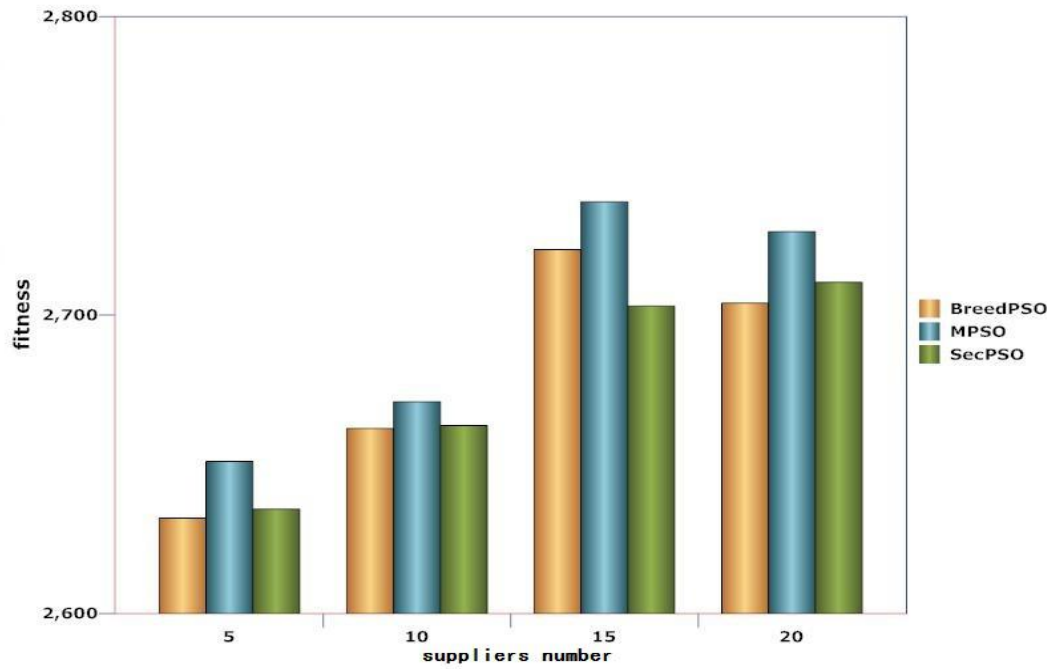


图 5-1b 子服务请求数 100，供应商数分别为 5，10，15，20

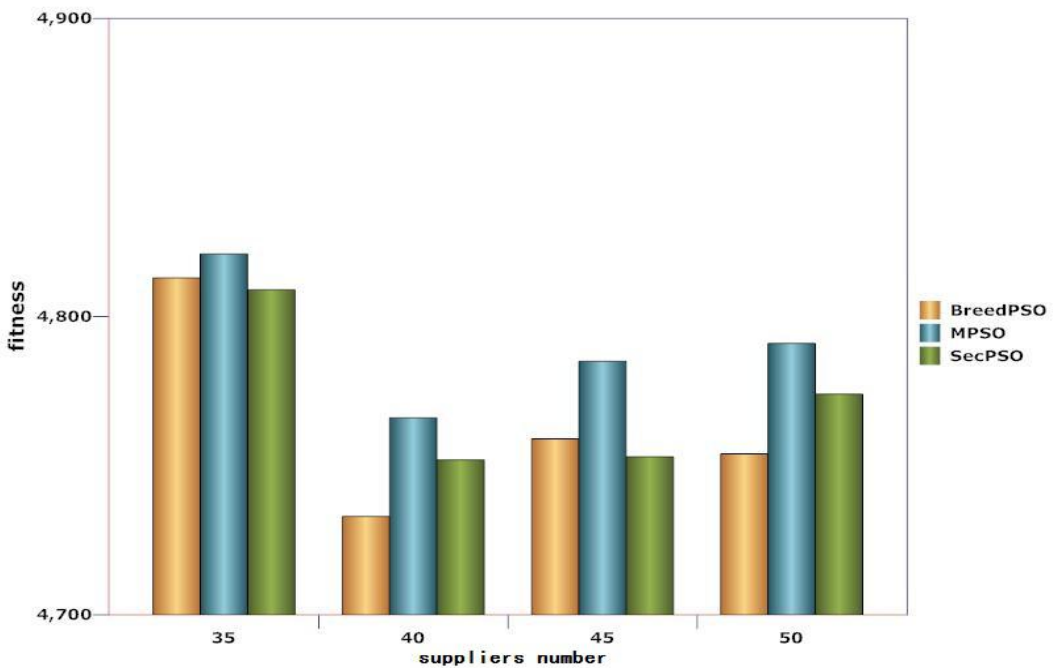


图 5-1c 子服务请求数 180，供应商数分别为 35，40，45，50

## 2. 有局部 QoS 限制的云服务组合

在实际云服务组合过程中，用户可能针对某一 QoS 属性有特别的要求。为了比较三种算法在用户有局部 QoS 要求的情况下求得的全局最优 QoS 属性值，实验中设定可靠性

属性为受限制的 QoS 属性，并且要求选择出的云服务组合的可靠性 QoS 度量值达到候选资源供应商的平均水平以上。实验模拟了子服务请求数从 20 到 500 的 12 组云服务组合过程，在该实验中不同情况下的服务组合可靠性的 QoS 限制值为所有候选供应商可靠性 QoS 度量值的平均值和子服务请求个数的乘积，并且要求选出的一套服务组合的可靠性 QoS 度量值必须大于该限制值。表 5-1 是 12 组服务组合的模拟仿真结果，展示出了 12 组仿真中对应的局部 QoS 限制值，以及全局最优 QoS 值。表中  $n$  代表子服务请求数， $m$  代表供应商数。

局部 QoS 的限制体现了用户对非功能性 QoS 属性的要求，受局部 QoS 限制的云服务组合在实际云平台事务中更为常见。从下表可以看出，当子任务个数在 100 以内时，实验结果表明：MPSO 算法在子服务请求数为 50 时所取得的最优全局 QoS 值不是三种算法中最大的。子服务请求数不大的情况下，解空间的范围也相对较小，劣质解数量也较少。所以在这种情况下，不太需要算法有很强的搜索能力。而另外几组实验数据中，MPSO 所取得的全局最优 QoS 值都在一定程度上大于另外两种算法。当解空间范围变大之后，算法的全局搜索能力对于搜索最优解就有着较大的影响。而 MPSO 通过学习因子非定值设定和引入二阶振荡环节来改进杂交粒子群算法后，提高了在大量劣质解包围优质解情况下搜索全局最优服务组合的能力。同时这 12 组实验中，解空间的范围区间为 $[20^{10}, 500^{200}]$ ，问题规模的跨度较大。以下实验结果说明 MPSO 算法在解决不同规模有局部 QoS 限制的云服务组合问题时也有良好的求解效果。

表 5-1 有局部 QoS 限制的情况下 3 种算法的对比

$n$	$m$	BreedPSO			MPSO		OscPSO	
		局部 QoS 限制值	全局 QoS 最优值	局部 QoS 值	全局 QoS 最优值	局部 QoS 值	全局 QoS 最优值	局部 QoS 值
20*10		487	446	584	462	622	446	591
30*15		749	657	817	680	969	656	820
50*25		1223	1042	1313	1036	1227	1038	1335
60*30		1515	1208	1771	1259	1740	1222	1689
80*40		1982	1618	2150	1652	2223	1630	2466
100*50		2468	1971	2964	1990	2869	1963	2600
130*65		3264	2600	3404	2692	3540	2598	3739
150*75		3771	2958	4164	3114	4234	3003	4042
160*80		4033	3156	4461	3297	4229	3161	4183
200*100		5013	3931	5374	3985	5242	3930	5551
300*150		7545	5786	8437	5879	8017	5825	8131
500*250		12457	9519	12674	9588	13081	9507	12797

### 5.3 面向批量服务请求的云服务调度仿真

通过第一阶段基于 QoS 的最优服务组合选择,可以得到各个服务请求对应的最优服务组合的集合,即第二阶段需要的数组  $TimeTable_{ij}$ ,作为第二阶段的云服务调度的输入源。下面进行的云服务组合调度的仿真实验,是在各个服务请求的最优服务组合确定的情况下进行的,也就是说每个服务请求使用的服务组合都是其在第一阶段中选择出的其对应的最优服务组合。假设每个子服务的执行时间和服务切换时间范围是 $[1, 60]$ 。为了测试本文提出的面向批量服务请求的云服务调度的模型和算法在不同的云服务请求数目和不同的云服务总数目搭配下的仿真情况,同时也为了体现考虑服务切换时间的情况下所得到的服务请求执行总时间的对比,我们将实验分为云服务总数分别是 10, 20 和 30 这三大类,并且每组实验搭配都是随机模拟了服务执行和服务切换的时间开销,以期得到在多种问题规模的情况下的算法求解效果。下表分别展示了三种算法仿真得到的总时间开销的平均值和最小值的对比结果。(n 代表云服务请求数目, m 代表云服务总数, 单位: 分)

表 5-2a 云服务调度仿真的环境下 3 种算法的对比

		MPSO		BreedPSO		SecPSO	
n	m	Min	Avg	Min	Avg	Min	Avg
40	10	1744	1757	1725	1759	1810	1833
50	10	2086	2111	2154	2174	2186	2205
60	10	2329	2376	2393	2480	2403	2424
70	10	2718	2769	2758	2785	2805	2856
80	10	3171	3196	3235	3240	3278	3305
90	10	3299	3331	3418	3443	3488	3558

表 5-2b 云服务调度仿真的环境下 3 种算法的对比

		MPSO		BreedPSO		SecPSO	
n	m	Min	Avg	Min	Avg	Min	Avg
50	20	2786	2845	2871	2909	2919	2955
60	20	3147	3174	3182	3193	3208	3271
70	20	3436	3511	3594	3632	3622	3686
80	20	3794	3854	3919	3956	3991	4054
90	20	4185	4224	4328	4358	4369	4403
100	20	4517	4538	4659	4701	4770	4803

表 5-2c 云服务调度仿真的环境下 3 种算法的对比

		MPSO		BreedPSO		SecPSO	
n	m	Min	Avg	Min	Avg	Min	Avg

60	30	3807	3854	3869	3916	3880	3954
70	30	4160	4206	4206	4246	4324	4377
80	30	4382	4413	4574	4663	4707	4749
90	30	4885	4929	5001	5076	5159	5198
100	30	5122	5146	5316	5414	5470	5511
110	30	5729	5773	5821	5830	5902	5918

上面三个表描述的是本文提出的基于 MPSO 的云服务调度算法解决面向批量服务请求的调度问题得到的总时间开销情况,是不同服务请求数目和不同云服务总数情况下的实验仿真结果。在问题规模情况较大的情况下,本文提出的算法均小于另外两种算法求得的时间开销。在本文的模型中,还加入了服务切换时间这一参考因素,所以问题的求解复杂度更高。从上述三个表可以看出,本文提出的调度算法能提高整个云服务请求队列的执行效率,同时采用最优服务组合作为服务调度的输入源,也提高了服务请求执行的质量。MPSO 算法在多次实验中所求得的平均值的优势也体现了足够的稳定性。同时,随着服务请求数的增加,问题规模也在逐渐增加,本文提出基于 MPSO 的调度算法能够在保持收敛速度的情况下,保持解的质量和稳定性,说明其能适应大规模的多目标优化问题的求解。

MPSO 算法在保证收敛速度的具有较强的避免局部优化的能力。下面三个图是云服务请求总数分别为 30, 50, 100 的情况下基于三种算法对应求得的总的生产时间随迭代次数增加的趋势图。

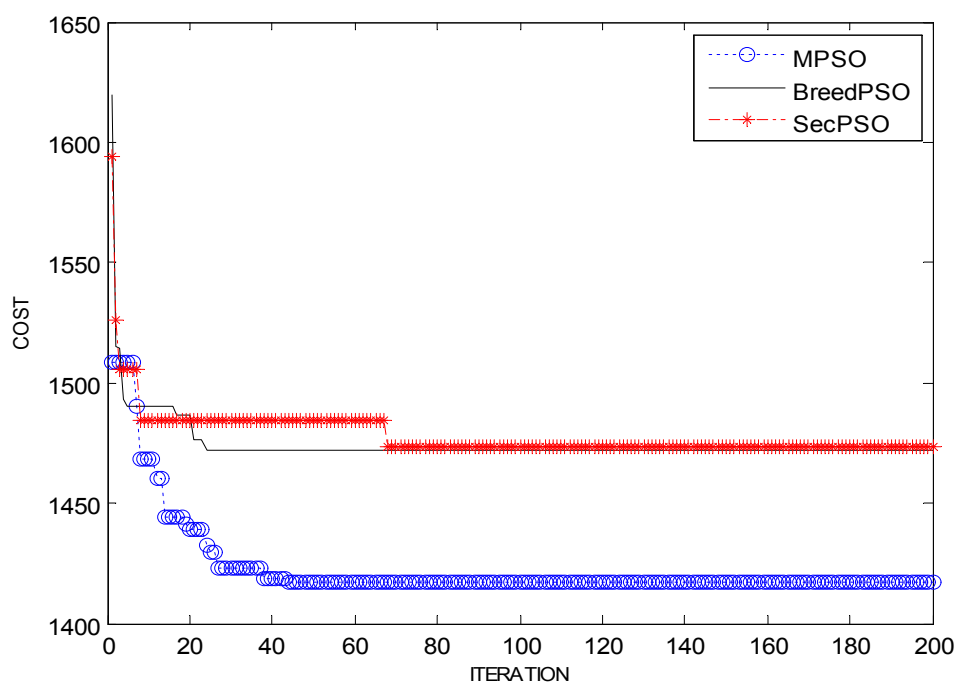


图 5-2a 三种算法的迭代图(请求数为 30)

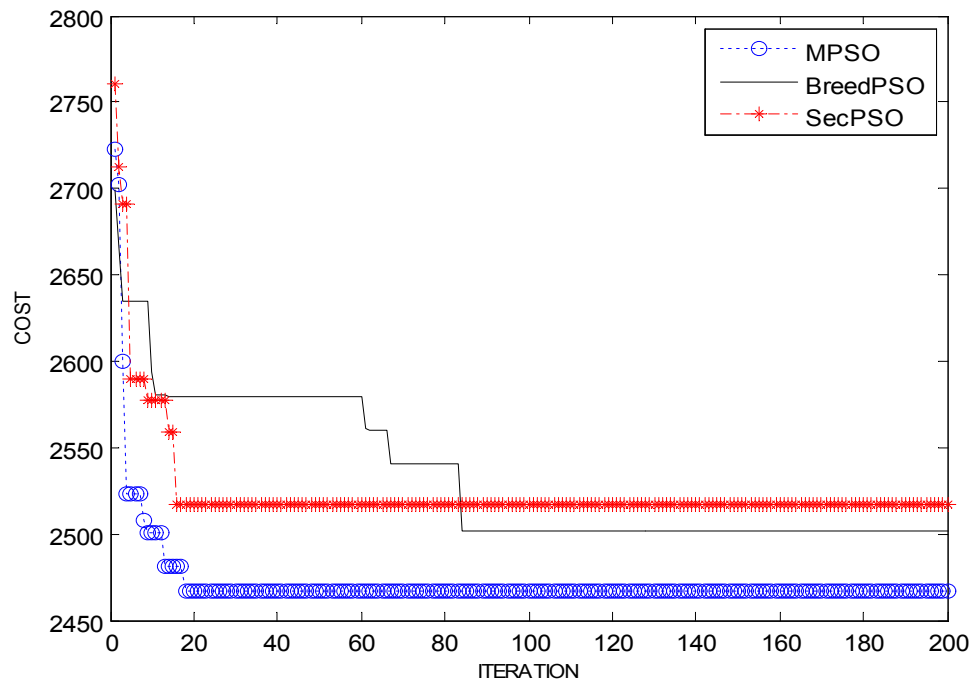


图 5-2b 三种算法的迭代图(请求数为 50)

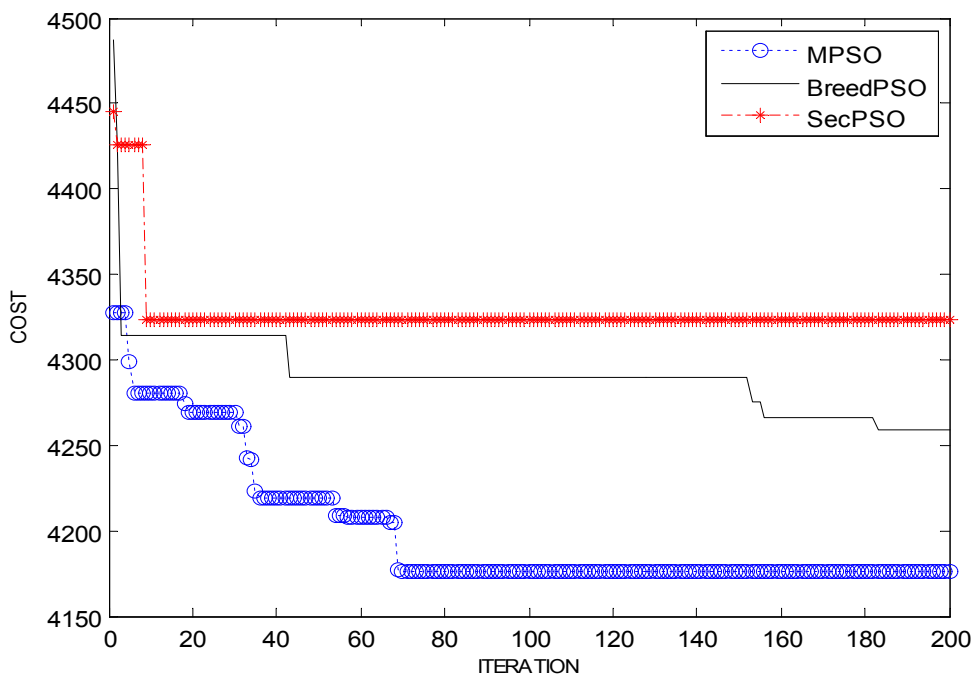


图 5-2c 三种算法的迭代图(请求数为 100)

从图中可以看出，三种算法中 MPSO 展现了较强的跳出前期局部优化的能力以及快速收敛能力。MPSO 算法在保持了各次迭代中种群优良性的情况下，不断和其他种群进行信息互换，不仅扩展了种群的多样性，且提高了最优解的质量。MPSO 算法在传统粒子群



优化算法的基础上引入振荡环节不仅能增强全局搜索能力,并且父代粒子杂交产生子代粒子的过程还进一步丰富了种群的多样性。在迭代过程中出现停滞时,MP SO 可以及时扩展种群的多样性和优良性,跳出局部最优,同时这些改变也是保持解质量前提下迭代速度进一步提高的原因所在。并且 MP SO 由于在局部搜索的过程中,不断扩展了最优候选值集合的范围,所以选取的最优值也比 BreedPSO 和 SecPSO 选取的最优值质量更好。但是,MP SO 快速收敛后难以找到更加优化的解,也体现出了其后期跳出局部最优能力的不足。MP SO 是在传统粒子群的基础上加入了多种改进方式,然而这些改进方式在前期搜索最优解的过程中能体现出非常好的效果,但是一旦搜索到最优解或者到达迭代过程的中后期,就很难有较好的跳出局部最优的表现。这也是下一步需要改进的。

#### 5.4 云服务调度实例仿真

面向批量云服务请求的调度模型与算法在实际应用中有着较为广泛的实例情况。由于云服务请求大多是需求功能繁多且个性化程度较高的服务请求,所在在本节中以各类转接线的制造服务调度为仿真背景,对本文提出的云服务调度模型与算法做了实例仿真。当前转接线制造所需的主要制造服务名称,服务重复 200 次的预计执行时间(单位:分钟)及服务编号如下:

表 5-3 服务细节列表

名称	服务编号	预计执行时间(/分)	名称	服务编号	预计执行时间(/分)
剥皮 1	A101	14	组装外壳	C025	63
剥皮 2	A102	14	组装弹片	C026	67
抽芯 1	A201	15	打螺丝	C045	57
抽芯 2	A202	15	测试 1	C001	47
焊接 1	A501	50	测试 2	C002	47
成型 1	B001	83	全检削进料点	C202	36
成型 2	B002	69	绕线	C220	40
焊接 2	A502	57			

假设有三类批量的服务请求,分别是服务请求 A 类,服务请求 B 类以及服务请求 C 类。同时我们假设对于三类服务请求的而言,需要主要的服务一致,不同在于 A10\*, A20\*, C00\*这三个部分不同的服务请求需要的服务选项不一致,除此之外,三种服务请求使用其他的选项是一致的。服务请求 A 类, B 类, C 类需要的各个单一服务的执行时间如下:

表 5-4a 服务请求 A 类

服务编号	预计执行时间(/分)	服务编号	预计执行时间(/分)
A101	14	C025	63
A102	0(无需此服务)	C026	67
A201	15	C045	57
A202	0(无需此服务)	C001	0(无需此服务)
A501	50	C002	47
B001	83	C202	36
B002	69	C220	40
A502	57		

表 5-4b 服务请求 B 类

服务编号	预计执行时间(/分)	服务编号	预计执行时间(/分)
A101	0(无需此服务)	C025	63
A102	14	C026	67
A201	0(无需此服务)	C045	57
A202	15	C001	47
A501	50	C002	0(无需此服务)
B001	83	C202	36
B002	69	C220	40
A502	57		

表 5-4c 服务请求 C 类

服务编号	预计执行时间(/分)	服务编号	预计执行时间(/分)
A101	0(无需此服务)	C025	63
A102	14	C026	67
A201	0(无需此服务)	C045	57
A202	15	C001	47
A501	50	C002	0(无需此服务)
B001	83	C202	36
B002	69	C220	40
A502	57		

在以各类转接线的批量制造服务调度为仿真背景下,服务管理平台将在线上集中各类信息,结合物流时间和服务请求执行时间进行整体调度。在本文的第三章中提出服务切换时间的概念,而在仿真环境下的服务切换就是指服务点之间的物流时间。在本例中我们随机生成各道服务点之间的切换时间。具体的服务点之间的切换时间(物流时间)表如下:

表 5-5 服务切换时间表

	A101	A202	A201	A202	A501	B001	B002	A502	C025	C206	C045	C001	C002	C202	C220
A101	0	21	6	18	19	0	12	8	15	27	1	23	7	20	18
A202	-	0	27	5	25	9	22	18	5	24	12	19	7	29	7
A201	-	-	0	13	13	30	19	24	29	24	6	19	25	5	24
A202	-	-	-	0	27	26	11	27	25	16	10	3	12	20	10
A501	-	-	-	-	0	24	27	19	15	17	13	24	6	11	3
B001	-	-	-	-	-	0	11	17	4	24	6	19	25	5	24
B002	-	-	-	-	-	-	0	27	17	25	8	23	9	30	18
A502	-	-	-	-	-	-	-	0	17	15	17	12	18	25	13
C025	-	-	-	-	-	-	-	-	0	26	28	29	29	3	30
C206	-	-	-	-	-	-	-	-	-	0	12	30	12	4	20
C045	-	-	-	-	-	-	-	-	-	-	0	15	13	0	7
C001	-	-	-	-	-	-	-	-	-	-	-	0	0	5	3
C002	-	-	-	-	-	-	-	-	-	-	-	-	0	5	30
C202	-	-	-	-	-	-	-	-	-	-	-	-	-	0	27
C220	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

在本文的实例仿真工作中，以上信息属于输入信息，根据每种类型的数量，经过编号后，需要分别输入到模型算法中的  $TimeTable_{m \times n}$  矩阵，同时将物流时间数据输入到服务切换时间矩阵  $\log_{n \times n}$ 。下面对三种类型的服务请求进行编号，同时为了方便展示调度结果，每种类型的连接数量设置为 10。

表 5-6 批量服务列表

服务请求类型	数量	编号范围	所需的服务总数
A	10	1-10	12
B	10	11-20	12
C	10	21-30	12

将以上信息输入到调度模型中，得到各个制造云服务请求的完成的最终时间：

表 5-7 调度结果表

编号	1	2	5	3	4	29	25	27	28	26
结束时间	779	862	945	1028	1111	1194	1277	1360	1443	1526

编号	30	18	20	24	22	19	23	21	15	14
结束时间	1609	1694	1777	1858	1941	2026	2107	2190	2275	2358

编号	16	13	17	11	10	8	9	12	7	6
结束时间	2441	2524	2607	2690	2771	2854	2937	3022	3103	3186

由于服务列表中的预计执行时间为该制造服务重复 200 次的执行时间,所以在本次实例中,相当于进行了 200 乘以 30 数量级别的服务请求执行,而仿真结果也说明了本文提出的模型算法能适用于大规模的云服务请求的执行。从表中可以看出服务请求序列总的执行时间是最后一个执行的服务请求的结束时间: 3186。

通过本文中提出的模型与算法所解出的云服务请求序列是在加入了服务切换时间(物流时间)影响因素的基础上得到的,符合实际的云服务环境下的批量服务请求的执行。但是在实际服务请求执行过程中,由于不同的服务切换时间和服务执行时间的比重不一样,所以若将所有的批量服务请求执行问题均采用本文的模型和算法进行求解,可能造成解的质量有所下降,具体来说:本文提出的模型试用于服务切换时间时间不大于服务执行时间的问题求解。

## 5.5 算法时间分析

本文提出的改进粒子群算法在动态参数优化,加入二阶振荡环节和杂家策略三个方面来改进基础粒子群算法。算法运行的时间开销展示了算法的执行效率,同时对算法进行理论时间分析,有助于进行实际问题的解决时间预估。在本节中,我们以 MPSO 算法解决最优云服务组合选择问题为例,分析 MPSO 的时间复杂度,并进行试验以验证分析。假设粒子群中共有  $N$  个粒子,并且进行  $m$  次迭代,则 MPSO 算法总的运算时间为  $T_{sum} = N \times m \times T$ 。其中  $T$  代表每个粒子在一次迭代中的时间开销,可以具体表示为:  
 $T = T_0 + p \times N \times T_1$ 。 $p$  表示杂交概率, $T_1$  代表每对粒子进行杂交的时间开销。 $T_0$  代表粒子进行速度和位置变换的时间开销。所以,综上可以得出 MPSO 总的运算时间:

$$T_{sum} = N \times m \times T_0 + p \times N^2 \times m \times T_1 \quad (5-1)$$

MPSO 算法的速度变化公式的时间复杂度与粒子的维度呈线性关系。在本文的第四章中已经介绍了粒子的维度等于当前的子服务请求数,所以粒子进行速度与位置变换的时间复杂度为  $O(P_{dim}) = O(Task_{num})$ ,即  $T_0$  与子服务请求数目呈线性关系。在杂交池中进行杂交的各对粒子需要对粒子的每一维度的位置和速度进行杂交,即  $T_1$  与子服务请求数目也呈线性关系,所以在粒子种群数目与迭代数目不变的情况下, $T_{sum}$  与子服务请求数呈线性关系。同时从  $T_{sum}$  的表达式可以看出,对于迭代次数  $m$ ,若粒子数目与子服务请求数不变, $T_{sum}$  与

$m$  也呈线性关系。由于云计算环境具有节点数目多等复杂特性, 所以 MPSO 算法的运行时间开销与子任务数呈线性关系这一特点保证了算法在求解过程中不会因为子服务请求数目的增加造成运算时间上有突然性的大范围增加, 而增加过程是一个平稳的过程, 这对于算法应用于实际问题的适应度和可预测性有较大的帮助。

实验模拟得出了多种情况下 MPSO 算法处理云服务组合问题的全局最优 QoS 值所花费的运算时间。MPSO 算法的运算时间随问题规模基本呈线性增加, 证明了上述的结论。当子服务请求数目小于 200 时, MPSO 算法基本能在 4 秒以内完成最优云服务组合的选取。子任务数在 500 以内时, 运行 MPSO 算法的运算时间在 7 秒以内。对于一个算法而言, 能否适应大规模问题的求解是评估其优劣的一个重要因素, 所以使用 MPSO 算法求解云服务组合的全局最优 QoS 值, 在不同问题规模下(尤其是问题规模较大的情况下)的运算时间也是一个重要的评价标准。实验结果说明, 从算法的计算时间开销的角度看, MPSO 算法也适用于求解云环境下大规模的服务组合与调度问题。

表 5-8 不同问题规模情况下的 MPSO 运算时间

子请求数	供应商数	运算时间(/s)	子请求数	供应商数	运算时间(/s)
150	15	3.368	400	40	5.997
200	20	3.794	450	45	6.593
250	25	4.405	500	50	7.276
300	30	5.013	550	55	7.838
350	35	5.400	600	60	8.486

下图是 MPSO 算法在迭代次数逐渐增加的情况下解决三种规模的云服务组合问题所花费的运算时间对比图。从图中可以看出, 随着迭代次数的增加, MPSO 求解服务组合问题的时间开销基本呈线性增长趋势。在实际的问题求解过程中, 由于问题的规模较大, 可能会需要增加迭代次数以选取最优值。MPSO 算法在迭代次数增加过程中, 运算时间平稳的增加也说明了其能适用于实际的云服务组合问题, 且不会出现迭代次数增加造成运算时间激增的情况。

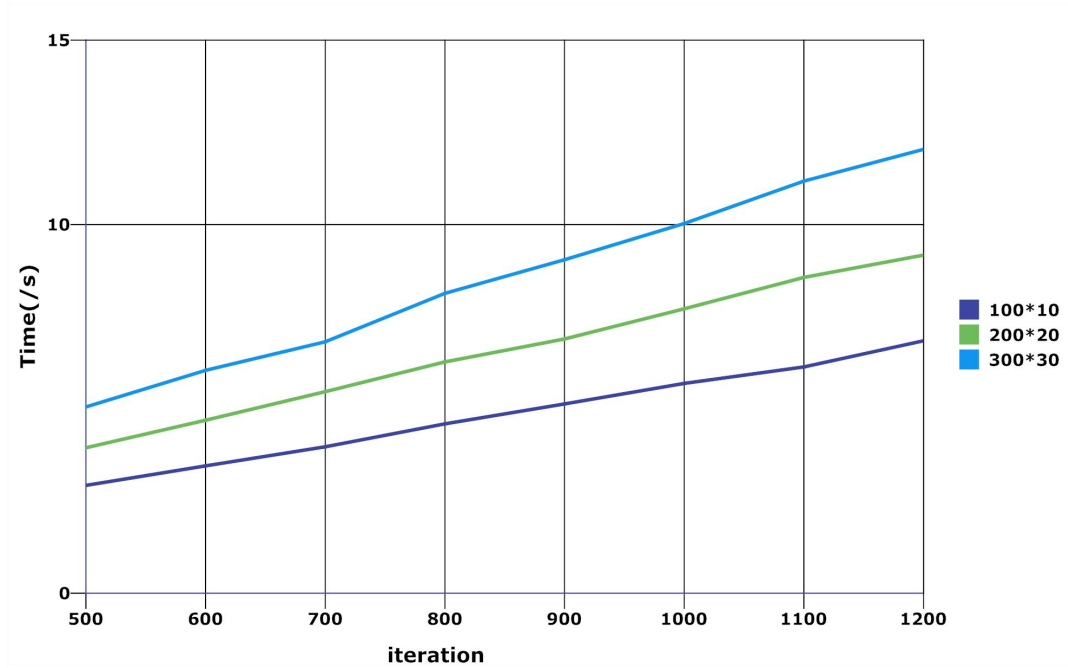


图 5-3 MPSO 运算时间与迭代次数对应关系图

## 5.6 本章小结

本章针对前两章提出的基于改进粒子群优化算法的云服务组合与调度优化算法主要做了两方面的实验仿真：(1)基于 QoS 度量值的最优服务组合选择仿真。(2)基于执行总时间最少的云服务调度仿真。然后本章完成了以各类转接线的批量制造服务调度为仿真背景的实例仿真。最后，针对改进的粒子群优化算法，本章进行了算法时间复杂度分析，并通过实验数据证明了分析结果。

## 第 6 章 总结与展望

### 6.1 总结

当今移动云计算等领域的云服务十分普及，云服务的请求与使用也更加频繁。由于云服务请求具有多功能复合需求的特点，所以云服务管理平台需要向服务请求提供高质量的云服务组合以满足其需求。同时为了保证服务请求的高效执行，还需要优化云服务调度策略。因此，云服务的管理常常要面对：如何在保证服务质量的前提下，依靠高效的调度策略满足云服务请求的执行需求。本文针对这一问题，建立了两阶段优化的云服务组合与调度模型，并将其转化为多目标优化问题，然后提出改进的粒子群优化算法(MPSO)求解该问题。论文主要研究了以下内容：

(1) 功能单一的云服务已经很难满足移动互联网用户提出的多功能需求的复合型云服务请求，所以需要选择云服务组合完成服务请求的执行。针对如何选择最优云服务组合这一问题，提出了基于全局 QoS 与局部 QoS 相结合的最优云服务组合选择模型。

(2) 移动云服务产业的极速发展使得云平台处理批量云服务请求的情况越来越多，然而云服务资源是有限的，所以有必要提出面向批量服务请求的云服务调度优化模型。针对这一需求，提出了以减少服务请求执行时间为目标的优化调度模型，同时通过使用各个服务请求对应的最优云服务组合，来保证服务请求的执行质量。在此基础上，给出了两阶段优化的云服务组合与调度的多目标优化的约束条件。

(3) 云服务组合与调度问题本质上是多目标优化问题，本文通过引入三种常见的有效改进方式对基础粒子群算法进行改进，并基于改进后的粒子群优化算法(MPSO)解决该问题。同时，在 5.5 节还给出了改进算法的时间复杂度分析与实验数据。

(4) 进行了基于 MPSO 的云服务组合与调度方法的仿真实验，并且证明该方法能在一定程度上提高解的质量，同时还能适应较大规模的问题求解。在此基础上，以各类转接线的制造服务调度为仿真背景，对本文提出的云服务调度模型与算法做了实例仿真，证明了其具有一定的实际应用价值。

### 6.2 下一步展望

本文在对云服务组合与调度的多目标优化研究中，有以下几点不足：

(1) 本文提出的面向批量服务请求的云服务调度模型还有待进一步的完善，在实际应用过程中有很多因素会影响云服务的调度进而影响云服务请求的执行，本文进行了容错机制方面的初步探索研究，但是还有更多的影响因素需要进行考虑并纳入本文提出的模型之中。

(2) 本文提出的结合全局 QoS 和局部 QoS 的最优云服务组合选择模型，考虑了多种云服务 QoS 属性且形成了全局 QoS 描述表达式，但是对局部 QoS 限制的情况只提出了两种。下一步可以具体针对 web 服务和制造云服务分别引入更多局部 QoS 限制。

(3) 粒子群优化算法还有一定的改进空间，本文提出的改进方法虽然可以快速收敛，但是跳出后期局部优化的能力还有待提升。



## 参 考 文 献

- [1] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.
- [2] Jula A, Sundararajan E, Othman Z. Cloud computing service composition: A systematic literature review[J]. Expert Systems with Applications, 2014, 41(8):3809 - 3824.
- [3] Jia-liang L, Jun-li S, Li-qun J. A QoS Evaluation Model for Cloud Computing[J]. Computer Knowledge & Technology, 2010.
- [4] Li Q, Guo Y, Li Q, et al. Optimization of Resource Scheduling in Cloud Computing[A]. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing[C]. USA:IEEE press, 2010:315 - 320.
- [5] Martin D, Paolucci M, Mcilraith S. Bringing Semantics to Web Services:The OWL-S Approach [J]. Semantic Web Services and Web Process Composition,2004:26-42.
- [6] 李 祯, 杨放春, 苏森. 基于模糊多属性决策理论的语义 Web 服务组合算法[J]. 软件学报,2009,20(3):583-596.
- [7] 王杰生, 李舟军, 李梦君. 用描述逻辑进行语义 Web 服务组合[J]. 软件学报,2008,19(4):967-980.
- [8] Haddad J E, Manouvrier M, Ramirez G, et al. QoS-driven Selection of Web Services for Transactional Composition[A]. IEEE International Conference on Web Services[C],USA:IEEE press,2008:653 - 660.
- [9] 许晓曼. 基于蚁群算法的 QoS 最优 Web 服务组合方法研究[D]. 南京邮电大学,2013.
- [10] 彭勇行. 管理决策分析[M]. 北京: 科学出版社,2000,147-285
- [11] 马林, 朱明, 王雷等. 支持 QoS 保障的可信服务组合调度算法[J]. 小型微型计算机系统,2013,34(2):193-198.
- [12] 叶枫, 王志坚, 徐新坤等. 一种基于 QoS 的云负载均衡机制的研究[J]. 小型微型计算机系统,2012,33(10):2147-2152.
- [13] 祝家钰, 肖丹, 王飞. 云计算下负载均衡的多维 QoS 约束任务调度机制[J]. 计算机工程与应用,2013:85-89.
- [14] 王鹏, 黄焱, 李坤等. 云计算集群相空间负载均衡度优先调度算法研究[J]. 计算机研究与发展,2014,51(5):1095-1107.
- [15] 张振中. 支持 QoS 的 Web 服务选择[D]. 北京邮电大学, 2006.
- [16] BEA,IBM,Microsoft,SAP and Siebel,Business Process Execution Language for Web Services Version 2.0[R], April 2007, <http://docs.oasis-open.org/wsbpel/2.0/>

- [17] Huang C, Chen D, Hu H. Intelligent agent-based scheduling mechanism for grid service[A]. International Conference on Machine Learning and Cybernetics[C].USA:IEEE press, 2004:16 - 21.
- [18] 马军, 罗国富, 路迪等. 制造资源云集成元建模及云 Agent 服务调度研究[J]. 中国机械工程,2014,25(7),917-923.
- [19] Chen Y. A service scheduler in a trustworthy system[A]. Proceedings of the 37th annual symposium on Simulation[C]. USA:IEEE Computer Society press,2004:89-96.
- [20] Yang Y, Zhou X, Yang G, et al. Trustworthy Service Scheduling Framework for QoS guarantee of composite service[A]. International Conference on Information Science and Engineering[M].USA:IEEE press, 2010:1 - 5.
- [21] Pan Q, Wang L, Sang H, et al. A High Performing Memetic Algorithm for the Flowshop Scheduling Problem With Blocking[J]. IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING,VOL.10,NO.3,JULY 2013
- [22] Onwubolu G, Davendra D. Scheduling flow shops using differential evolution algorithm[J]. European Journal of Operational Research,2006,171(2):674-692.
- [23] Liao C J, Tseng C T, Luarn P. A discrete version of particle swarm optimization for flowshop scheduling problems[J]. Computers&Operations Research, 2007, 34(10):3099-3111.
- [24] Naderi B, Tavakkoli-Moghaddam R, Khalili M.Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan[J]. Knowledge-Based Systems, 2010, 23(2):77-85.
- [25] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. Systems,Man,and Cybernetics,Part C:Applications and Reviews,IEEE Transactions on, 1998, 28(3):392-403.
- [26] Laili Y, Tao F, Zhang L,et al. The optimal allocation model of computing resources in cloud manufacturing system[A]. 2011 Seventh International Conference on Natural Computation[C], USA:IEEE press 2011,4:2322-2326.
- [27] M. Dakshayini D, H. S. Guruprasad D. An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment[J]. International Journal of Computer Applications, 2011,32(9):23-29
- [28] Bossche RVD, Vanmechelen K,etc. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds[J]. Future Generation Computer Systems, 2013, 29(4):973 - 985.
- [29] Fang Y, Wang F, Ge J. A task scheduling algorithm based on load balancing in cloud computing[M].Web Information Systems and Mining.Germany: Springer press, 2010:271-277.
- [30] 单志广, 戴琼海, 林闯等. Web 请求分配和选择的综合方案与性能分析[J]. 软件学

- 报,2000,12:355-366.
- [31] 姜文超, 金海, 王述振等. 一种基于多优先级队列和 QoS 的服务调度策略[J]. 小型微型计算机系统,2008,29(3):450-454.
- [32] 高昂. Google 构建我们身边的云计算[J]. 程序员, 2008, (11):54-55.
- [33] Christensen E, Curbera F, Meredith G, et al. Web services description language (WSDL) 1.1[J]. 2001.
- [34] Curbera F, Duftler M, Khalaf R, et al. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI[J]. Internet Computing IEEE, 2002, 6(2):86 - 93.
- [35] Xu X. From cloud computing to cloud manufacturing[J]. Robotics and computer-integrated manufacturing, 2012, 28(1):75-86.
- [36] 何炎祥等, 动态 web 服务组合关键技术与性能分析[M]. 北京: 清华大学出版社,2011.6,58-59.
- [37] Yu Q, Liu X, Bouguettaya A, et al. Deploying and managing Web services:issues,solutions,and directions[J]. The VLDB Journal, 2008, 17(3):537-572.
- [38] Wang H, Tong P, Thompson P. QoS-based web services selection[A]. IEEE International Conference on e-Business Engineering[C]. USA:IEEE press, 2007: 631-637.
- [39] Shangguang Wang, Qibo Sun, Zou Hua, et al. Particle swarm optimization with skyline operator for fast cloud-based web service composition[J]. Mobile Networks and Applications. 2013, 18(1):116-121.
- [40] Fanjiang, Yong-Yi, Yang Syu. Semantic-based automatic service composition with functional and non-functional requirements in design time:A genetic algorithm approach[J]. Information and Software Technology, 2014, 56(3):352-373.
- [41] Mardukhi F, NematBakhsh N, Zamanifar K, et al. QoS decomposition for service composition using genetic algorithm[J]. Applied Soft Computing, 2013, 13(7):3409-3421.
- [42] Wu Q, Zhu Q. Transactional and QoS-aware dynamic service composition based on ant colony optimization[J]. Future Generation Computer Systems, 2013, 29(5):1112-1119.
- [43] 温涛, 盛国军, 郭权等. 基于改进粒子群算法的 Web 服务组合[J]. 计算机学报,2013,36(5):1031-1046.
- [44] 李昌兵, 杜茂康, 付德强. 基于层次粒子群算法的非线性双层规划问题求解策略[J]. 系统工程理论与实践,2013,33(9):2292-2298.
- [45] 李松, 刘力军, 翟曼. 改进粒子群算法优化 BP 神经网络的短时交通流预测[J]. 系统工程理论与实践,2012,32(9):2045-2049.
- [46] 王廷梁, 王理, 夏国平. 基于混合粒子群算法的装运机械组合优化方法[J]. 系统工程理论与实践,2012,32(10):2262-2269.
- [47] 黄伟, 罗世彬, 王振国. 基于动态参数的杂交粒子群优化算法[J]. 计算机科学, 2010, 37(12):165-166.
- [48] Chandrasekaran S, Miller J A, Silver G S, et al. Performance Analysis and Simulation of Composite

- Web Services.[J]. Electronic Markets, 2003, 13(2):120-132.
- [49] Murata T. Petri nets:Properties,analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4):541-580.
- [50] Ajmone Marsan M. Stochastic Petri nets:An elementary introduction[J]. Lecture Notes in Computer Science, 1989, 424:1-29.
- [51] Narayanan S, McIlraith S A. Simulation,verification and automated composition of web services[A]. Proceedings of the 11th international conference on World Wide Web[C]. USA: ACM Press, 2002:77-88.
- [52] 张卫, 潘晓弘, 刘志等. 基于云模型蚁群优化的制造服务调度策略[J]. 计算机集成制造系统,2012,18(1):201-207.
- [53] 陈国斌. 一种分布式动态负载均衡调度算法[J]. 广西师范学院学报: 自然科学版,2014,(1):56-59.
- [54] 郭平, 李涛, 李琪. 一种云计算环境下的负载调度算法 [J]. 系统工程理论与实践,2014,34(s1).269-275
- [55] 郑卉, 郭平, 李琪等. 基于带宽约束的云计算负载调度算法[J]. 西南师范大学学报: 自然科学版,2014,(7):121-128.
- [56] 施进发, 焦合军. 面向多维度 QoS 的云资源在线调度优化研究[J]. 计算机工程与设计,2013,34(12):4299-4303.
- [57] 丁宇光. 基于 QoS 约束的云计算调度算法的研究[D]. 辽宁大学,2012.
- [58] 刘文娟. 基于 QoS 约束的云 workflow 任务调度模型与算法研究[D]. 中国科学技术大学,2012.
- [59] 朱泽民, 张青. 基于多维 QoS 和云计算的资源负载均衡调度研究[J]. 计算机测量与控制,2013,21(1):263-265.
- [60] Liu A, Li Q, Huang L, et al. QoS-Aware Scheduling of Web Services[A]. International Conference on Web-age Information Management[C]. USA:IEEE press, 2008:171 - 178.
- [61] Kennedy J, Eberhart R. Particle swarm optimization[A]. IEEE International Conference on Neural Networks[C]. USA:IEEE press, 1995:1942 - 1948.
- [62] Zheng X. A modified particle swarm optimization with differential evolution mutation[A]. International Conference on Natural Computation[C]. USA:IEEE press, 2010:2530 - 2534.
- [63] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报,2007,18(4):861-868
- [64] Li N, Zhu S. Modified particle swarm optimization and its application in multimodal function optimization[A]. Proceedings International Conference on Transportation Mechanical & Electrical Engineering[C], 2011:375 - 378.
- [65] Arumugam M S, Rao M V C, Chandramohan A. A new and improved version of particle swarm optimization algorithm with global-local best parameters[J]. Knowledge and Information systems, 2008,

16(3):331-357.

- [66] Shi Y, Eberhart R. A modified particle swarm optimizer[A]. Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on[C]. USA:IEEE press, 1998:69-73.
- [67] 胡建秀, 曾建潮. 二阶振荡微粒群算法[J]. 系统仿真学报, 2007, 19(5):997-999.
- [68] Lovbjerg M, Rasmussen T K, Krink T. Hybrid particle swarm optimiser with breeding and subpopulations[A]. Proceedings of the Genetic and Evolutionary Computation Conference[C]. San Francisco, USA, 2001:469-476.

## 致 谢

三年时间，一晃而过，转眼就要毕业了。三年的研究生学习期间，在老师和同学的帮助下，我不仅强化了专业技能，同时也培养了更加严谨的科研态度。在这期间，我要对多位曾经帮助过我的老师，同学及亲人表达我最为真挚的感谢。

首先要感谢我在研究生阶段的两位导师：张美玉和简琤峰老师。两位老师在生活上对我给予了极大的关心，在学习上帮助我养成了良好的科研习惯。正是这些帮助，让我能在科研方面取得一点点小小的成绩以及顺利完成毕业论文。

同时，我还要感谢数字媒体技术实验室的陶萌等同学对我的关心与帮助，我们的并肩协助使得研究生阶段的科研学习任务能够顺利地完成。三年的时间里，实验室的同学们互帮互助，大家如一家人般地愉快相处。祝愿大家在以后的工作学习中一切顺利！

最后，我要感谢父母及女友，感谢你们在我本科和研究生的学习阶段对我的大力支持。结束了人生中的学习阶段，我即将要进入社会开始工作，面对未知的将来，我将鼓足勇气，为自己加油！

## 攻读学位期间的科研项目和学术论文

### 参加的科研项目

- [1] 浙江省科技厅公益计划项目:支持移动 WEB 手势识别的草图协同创意设计平台。  
(No.2014C31081), 2014.01-2015.12
- [2] 企业项目:北京赛尔教育工业产品设计云服务关键技术开发。(No.RJ-[2012]051@),  
2013.01-2014.12

### 已发表的论文

- [1] Jian Chengfeng, Wang Yekun, Tao Meng, Zhang Meiyu. Time-Constrained Workflow Scheduling In Cloud Environment Using Simulation Annealing Algorithm[J]. Journal of Engineering Science and Technology Review, 2013, 6(5): 33-37.
- [2] Jian Chengfeng, Wang Yekun. Batch task scheduling-oriented optimization modelling and simulation in cloud manufacturing[J]. International journal of simulation modelling, 2014, 13(1): 93-101.