

分 类 号\_\_\_\_\_

学号 M200973427

学校代码 10487

密级\_\_\_\_\_

华中科技大学

# 硕士学位论文

网络服务选址问题研究

学位申请人： 孙东方

学 科 专 业： 管理科学与工程

指 导 教 师： 杨超 教授

答 辩 日 期： 2011 年 12 月 26 日

**A Thesis Submitted in Partial Fulfillment of  
the Requirements for the Master Degree of Management**

**A Research on the Web Service Location Problem**

**Candidate : Sun Dongfang**

**Major : Management Science and Engineering**

**Supervisor : Prof. Yang Chao**

**Huazhong University of Science and Technology**

**Wuhan, Hubei 430074, P. R. China**

**December, 2011**

## 独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的科研成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到，本声明的法律结果由本人承担。

学位论文作者签名：

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，即：学校有权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。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属    保密 $\square$ ，在\_\_\_\_\_年解密后适用本授权书。

不保密 $\square$ 。

(请在以上方框内打“√”)

学位论文作者签名：

日期：        年        月        日

指导教师签名：

日期：        年        月        日

# 华中科技大学硕士学位论文

---

## 摘 要

随着互联网技术的发展,网络服务(Web Service)整体响应时间(Response Time)中网络延时(Network Latency)所占比重越来越大。在网络服务应用范围不断扩大,人们对其响应时间要求却越来越高的情况下,如何对网络服务进行合理选址,从而降低总体响应时间,提高网络服务的服务水平显得尤为重要。

本文重点从响应时间的角度对网络服务选址问题进行了研究,因为网络服务体系中响应时间是决定服务水平的关键因素。研究中应用选址问题的相关理论并结合网络服务自身特点,提出了基于时间满意度的多服务带容量限制的网络服务选址模型。本文模型在国外相关研究的基础上,分析了网络服务体系的自身特点,结合这些特点进行了模型改进:网络服务需求点(Web Service Requester, WSR)只能选择一个网络服务提供商(Web Service Provider, WSP);增加了网络服务的容量约束;提出了“有效覆盖”的想法,强调了网络服务需求方可以根据自己的服务水平偏好自主选择网络服务。同时还将时间满意度引入目标函数,采用离散的时间满意度函数,与网络服务体系中的分级定价相关联,使得本文的模型不仅更好地贴合了实际应用,也具有了比较好的拓展性。本文问题是 NP 困难问题,因而采用贪婪算法思想求解,将问题划分为多个子问题后,根据子问题规模较小,使用了贪婪算法和动态规划法相结合的混合算法进行了求解,取得了比较好的求解效果。

**关键词:** 网络服务; 响应时间; 时间满意度; 贪婪算法; 动态规划

## Abstract

Along with the development of Internet Technology, network latency is taking up more and more proportion of overall response time of web service. And people's demand for response time is getting higher and higher while web applications based on web service is expanding constantly into people's life and work. Hence, it's particularly important to locate the web service properly so as to reduce the overall response time and improve the service level of web service.

In a network system based on web service, the response time is the key factor to impact the service level. So this paper focused to research the web service location problem from the point of response time. Given the features of web service, a brand new capacity-constrained multi-service web service location model based on time satisfaction was proposed by applying the classic location theory. After a review of related literature, location models for web service used before were improved with an analysis on the features of web service. Firstly, web service requester (WSR) can only get web service from one web service provider (WSP). Secondly, each type of web service has a capacity constraint. Thirdly, idea of "effective coverage" was put forward so that the WSR can choose the right WSP according to WSR's service level preference. Moreover, time satisfaction was introduced in to the final model, which not only makes the model close to practical application, but also gives it better expansibility. This model is a NP-Hard problem, so by using Greedy Algorithm, this problem was divided into some much smaller sub-problems. Finally, a hybrid algorithm combining the Greedy Algorithm and Dynamic Programming was used to solve the problem, and the result showed good efficiency and accuracy.

**Keywords:** Web Service; Response Time; Time-satisfaction; Greedy Algorithm; Dynamic Programming

# 华中科技大学硕士学位论文

---

## 目 录

摘 要.....	I
Abstract.....	II
目 录.....	III
1. 绪论	
1.1. 研究背景与意义.....	1
1.2. 国内外研究综述.....	5
1.3. 论文框架与研究内容.....	9
2. 一般选址问题相关理论综述	
2.1. 一般选址问题介绍.....	11
2.2. 基本覆盖选址问题及常用算法介绍.....	13
2.3. 覆盖选址问题中时间满意度的度量.....	错误!未定义书签。
3. WS 选址问题及模型构建	
3.1. WS 体系介绍.....	18
3.2. WS 选址问题基础模型.....	19
3.3. 基于时间满意度的带容量限制多服务 WS 选址模型.....	24
4. WS 选址模型求解	

# 华中科技大学硕士学位论文

---

4.1. 模型分析及算法设计 .....	32
4.2. 算例验证 .....	38
4.3. 结论 .....	42
<b>5 总结与展望</b>	
5.1 全文总结 .....	43
5.2 研究展望 .....	44
<b>参考文献.....</b>	<b>46</b>
<b>附录2 攻读学位期间参加的科研课题.....</b>	<b>50</b>

# 华中科技大学硕士学位论文

## 1 绪论

### 1.1 研究背景与意义

#### 1.1.1 研究背景

近年来,随着电子商务(E-commerce)的迅速崛起和“云计算”(Cloud Computing)概念的提出到开始逐步进入商务应用,基于网络的应用(Web-based Application)从集中式演变为分布式,从局部化演变为全球化,,使得需要强大处理能力的大规模的商务应用成为可能。WS(Web Service,后面简称WS)不仅是一种创新的web应用模式,也是一个全新的分布式计算模型,是web上数据和信息集成的有效机制,从其诞生之日起就受到了业内人士的追捧,而各种各样的WS技术也层出不穷。

表 1.1: www.webxml.com.cn 网站提供的部分 web service

服务名称	服务发现	服务描述
天气预报 Web 服务, 数据来源于中国气象局	~/WeatherWebService.asmx?disco	~/WeatherWebService.asmx?wsdl
IP 地址来源搜索 WEB 服务	~/IpAddressSearchWebService.asmx?disco	~/IpAddressSearchWebService.asmx?wsdl
Email 电子邮件地址验证 WEB 服务	~/ValidateEmailWebService.asmx?disco	~/ValidateEmailWebService.asmx?wsdl
中文简体字 <->繁体字转换 WEB 服务	~/TraditionalSimplifiedWebService.asmx?disco	~/TraditionalSimplifiedWebService.asmx?wsdl
中文 <-> 英文双向翻译 WEB 服务	~/TranslatorWebService.asmx?disco	~/TranslatorWebService.asmx?wsdl
火车时刻表 WEB 服务	~/TrainTimeWebService.asmx?disco	~/TrainTimeWebService.asmx?wsdl
中国股票行情数据 WEB 服务	~/ChinaStockWebService.asmx?disco	~/ChinaStockWebService.asmx?wsdl
即时外汇汇率数据 WEB 服务	~/ExchangeRateWebService.asmx?disco	~/ExchangeRateWebService.asmx?wsdl
腾讯 QQ 在线状态 WEB 服务	~/qqOnlineWebService.asmx?disco	~/qqOnlineWebService.asmx?wsdl
中国电视节目预告	~/ChinaTVprogramWebService.asmx?disco	~/ChinaTVprogramWebService.asmx?wsdl
外汇-人民币即时报价 WEB 服务	~/ForexRmbRateWebService.asmx?disco	~/ForexRmbRateWebService.asmx?wsdl
中国股票行情分时走势预览缩略图 WEB 服务	~/ChinaStockSmallImageWS.asmx?disco	~/ChinaStockSmallImageWS.asmx?wsdl
国内飞机航班时刻表 WS	~/DomesticAirline.asmx?disco	~/DomesticAirline.asmx?wsdl
中国开放式基金数据 WS	~/ChinaOpenFundWS.asmx?disco	~/ChinaOpenFundWS.asmx?wsdl

注: 其中为了节省空间已将网址缩写成“~/XXX.XXX”格式, 其中“~/”表示被省略的网址“http: www.webxml.com.cn/WebServices/”



# 华中科技大学硕士学位论文

---

经过近十年的发展,WS 几乎以及进入到所有的领域,所提供的服务范围已经包含了商业和贸易、通讯和通信、公共事业、图像和多媒体、科学计算和单位转换以及获取标准数据等等诸多方面。表 1.1 给出了 [www.webxml.com.cn](http://www.webxml.com.cn) 上所提供的部分 WS。从中,我们不难看出,WS 已经逐步深入到人们工作和生活方方面面。无疑,WS 的价值和潜力正在被人们所发现和开发,并且随着科学技术特别是信息技术的发展,WS 将会帮助人们完成更多的事情,创造更大的价值。关于 Web Service 的定义,虽然现在 WS 已经为人们所熟知,但 WS 确切定义却是很难得到。正如 W3C 所说的,“WS 对于不同的人,有着不同的含义”。但总体而言,WS 提供了一种标准化的方法,使得在不同平台和框架下运行的程序可以进行数据交互。其典型特征有互操作性和拓展性以及由于采用 XML 而具备的机器可识别性,并且可以被松散地组织在一起,来完成极其复杂的运算或者操作。

Web Service 本身具有的巨大潜力和价值:作为一种技术或者说是一种解决方案,Web 服务使用一系列的标准协议来对各种请求做出响应,使用 HTTP/TCP 等标准的网络协议完成底层的传输,以 XML 作为数据表示的基础,通过简单对象访问协议 SOAP(Simple Object Access Protocol) 协议在系统间交换信息,通过 Web 服务描述语言 WSDL(Web Service Description Language)等来描述和记录 Web 服务所产生和接收的消息,通过通用描述、发现和集成协议 UDDI(Universal Description, Discovery, Integration)来登记和寻找服务,这些技术构成了 Web 服务的支撑技术。

Fensel 和 Bussler (2002)提出 WS 将网络从一个信息的集合变成一个分布式的可以运算的设备。Hamadi 和 Benatallah(2003)也表示因特网正演变成 WS 的载体而不仅是信息的储藏室。许多公司或组织正倾向于将核心的商业竞争力至于因特网作为一种 WS 的集合。面对长久以来对于个人以及企业之间的快速而灵活的信息共享需求,WS 是一个很具有潜力的解决方案。Web 是为程序到用户的交互,而 WS 是为程序到程序的交互做准备。WS 使公司可以降低进行电子商务的成本、更快的部署解决方案以及开拓新机遇。

同时,WS 使转换基于互联网的商业模式成为可能,对于终端用户而言,WS 比起传统的技术更加容易获得。另外,由于 WS 是一个组件性的技术,支持质量导向

和服务定制，所以可以实现分级定价。这样对普通用户而言，可以降低服务的进入成本。甚至，越来越多的公司，为了减少维护成本和部署全球化应用，将其核心业务都采用 WS 的方式移植到网络上。

由于 WS 的巨大潜力和价值，当前 WS 已经大行其道，种类繁多，专门去定义 Web service 的概念已经没有意义。但仔细去看，主要存在两种 Web Service：一类是为了使得商业应用的分布式计算更加快捷，另一类则是为了加强不同程序应用间的互用性(Beznosov 等, 2005)。另外，根据不同的 WS 的实际应用背景和存在的主要问题，世界范围内的研究人员从不同的角度进行了相应的研究工作，也取得了一些成果。例如，WS 中的数据集成，WS 的组合，根据分布式数据库的分布式的优点研究如何提高 WS 质量，语义在 WS 中的应用，高效的 WS 发现技术和架构，针对 WS 应用中的安全及隐私问题，适合 WS 的安全和加密策略，P2P 环境中 WS 的有效表示和执行，网格计算与 web 服务的有机结合等(岳昆等, 2004)。

## 1.1.2 研究意义

WS 的兴起并被广泛应用已经成为一种不可扭转的趋势，随着高速宽带网络出现和计算机处理能力的大幅度提升，本地处理时间以及数据上传时间甚至网络中的排队等待时间都被大大的缩短，于是原先不起眼的网络延时变成影响响应时间的重要因素而得到重视。可以预见，在不久的将来，计算机处理能力将更加惊人，而不同的 WS 也由于技术的更新，WS 同质化程度会变得很高，这时 WS 的网络延时将更加重要，因而如何确定 WS 服务站的位置以及每个站点服务器数量(决定服务能力)以使网络延时更少具有特别的意义。

不仅如此，随着智能系统和分布式系统开发应用的深入，网络延时的重要性可能显得更为突出。随着“云计算”被提出并获得众多企业和组织的追捧，未来将不可避免的出现更加复杂的智能的分布式系统，在这些系统中，需要调用大量的 WS，而且对同一个 WS 可能需要调用很多次，这样，网络延时就会被成几何级数地放大。而响应时间在网络应用中，特别是电子商务和金融系统中的应用对于响应时间有着“严苛”的要求，即便是在一般的网络应用中(比如云邮箱，交通信息查询，在线地

# 华中科技大学硕士学位论文

---

图等等), WS 的响应时间也很大程度上影响着用户的满意度。所以, 对于 WS 选址问题进行深入研究, 从而降低整体的 WS 的响应时间变得尤为重要。

然而, 令人遗憾的是目前对于 WS 的研究多半集中于技术层面, 研究 WS 的架构、WS 的发现、语义识别等等, 而专门针对 WS 的选址问题的研究尚未受到学界重视。尽管目前对于传统选址问题的研究已经非常深入, 但对于 WS 的选址问题的研究却是少之又少。而由于 WS 与传统选址问题的应用环境不一样, WS 有着自身的特点, 所以我们需要专门的进行 WS 的选址问题的研究。

综上所述, WS 的发展和应用迫切需要降低响应时间提高服务水平, 而随着技术的进步, 网络延时在 WS 的响应时间中占有越来越多的比重, 而通过研究 WS 的选址问题可以很好的解决这一问题, 本文将本课题也是以此为出发点。本文将从企业应用的角度对 WS 的选址问题进行研究, 借鉴传统选址问题中的模型和算法, 并结合 WS 自身特点, 建立全新的 WS 选址模型。同时, 本文还将给出相应的求解算法, 以期促进 WS 的发展和推广应用, 为企业和政府机构提供有力的应用理论支撑。具体而言, 本文主要研究目的有:

1、本文将选址问题相关理论和方法引入到 WS 选址问题中, 提高 WS 的运行效率和可获得性也即覆盖尽可能多的客户, 从而获得效益的增加。

2、增加时间满意度的考量, 化传统的覆盖半径为阶梯状, 方便 WS 提供商根据不同的服务水平进行服务定价。传统文献研究中假设客户可以被多个 WS 提供商覆盖, 实际中由于 WS 自身的特点, 客户只能选择单个 WS 提供商, 而且通常会选择距离自己最近的, 从而获得更好的用户体验。

3、本文分析了主要的 WS 的网络类型, 包括注册中心, P2P 和索引三种主要类型。不同的类型具备自身特点, 传统 WS 选址研究中往往只关注于注册中心类型。并且通过转化, 最终融合了三种主要的 WS 的网络类型特点, 提出自己的 WS 选址模型

## 1.2 国内外研究综述

### 1.2.1 对 WS 相关问题的研究

WS 的出现并受到追捧源自于其对于分布式系统的良好支持和各种应用的便捷性和灵活性。特别是基于服务驱动架构(SOA, Service oriented Architecture), 基于服务的科学计算(Service Oriented Computing), QoS(quality oriented service)等理念的提出(Daniel 等, 2002; Papazoglou 等, 2003)。而且, WS 的产生也将网络从一个信息的集合变成一个分布式的可以运算的设备。这是一个巨大的变革, 因而在其出现不久, 就有不少相关研究发表了。为了充分发挥 WS 的潜力, 我们需要用合适的方法描述 WS, 为此, Fensel & Bussler(2002)定义了一个全面的 Web Service Modeling Framework (WSMF), 该框架给出了较好的理论模型以便开发和描述 WS 以及他们的组合体, 并提出借助可拓展的中间服务来获得最大耦合度。Turner 等(2003)提出了将软件作为服务(SaaS)的相关概念。随着网络及计算机技术的发展, 基于服务驱动架构(SOA)的 WS 使得企业内部和企业间的商业流程变得可以获得, 但为了能更好地支持动态的商业流程及其管理, 还需要做很多工作比如如何在商业流程的具体活动中使用 WS, 或者如何在 WS 中表达商业流程, 再者, 如何确定不同的商业模式完成不同的实际商业流程。Leymann(2002)等阐述了 WS 和商业流程管理的关系并给出了详尽的演示。Pauw 等(2005)提出了 WS 的领航员的概念, 帮助理解 WS。并用 IBM 的经验和实际项目应用, 给出了解决一些实际问题的方法。Wang(2003)等总结了当前框架下 web service 遭遇的一些主要问题比如安全问题, 并在文章最后给出了一些语义类 WS 的研究方向。

随着 WS 研究和应用的不断深入, 2005 年前后 WS 的研究已经取得了不少进展。Beznosov(2005)等总结了 WS 的发展和相关技术, 并重点阐述了 WS 的安全问题如 SOAP 信息安全。之后, 关于 WS 的研究应用进一步展开, 而随着 WS 的应用的增加, 对 WS 本身的优化逐渐见于各项研究中。对于 WS 的优化研究主要集中于 WS 的发现和筛选, WS 的描述, WS 重建以及语义 WS 等方面, 从而更好的发挥 WS 优势。

在 WS 发现过程中, 我们知道 WS 的体系中, 很关键的一步就是 WSP 发布 WS

# 华中科技大学硕士学位论文

---

并由 WSI 提供注册，Crasso(2008)提出虽然这些注册机制提供了查询方面的便利，但仍然不宜使用，而且常常需要用户花费大量的时间人工浏览和选择服务并且给出一种全新的 WS 搜索方法 WSQBE，该方法不仅能降低对查询语句的规范化要求，而且可以通过返回更加短小精准的候选服务列表来帮助用户发现需要的服务。与之前的服务搜索发放不同，WSQBE 的搜索过程是基于自动搜索内存优化机制(automatic search space reduction mechanism)使得该法更具效率。Rodriguez 等(2010)也提出了 WS 发现过程中一些可以阻止 WS 被发现的不好的做法，并提供了解决方案。并相应地给出一种全新的发现机制。

由于 WS 是一个独立的模块式软件组件，可以分布到整个网络上然后在一个远程的宿主上执行，具备较好的可拓展性，既能用于商业应用的早期开发又能保证后期的功能增加合成。然而与 WS 密切相关的网络通讯和业务处理上的开销也为其表现提出了挑战，所以 WSP 们常常在发布一个 WS 时，提出一个服务水平条款。该服务水平条款保证客户根据其支付得到相应质量的服务。本文研究两个提供相同功能 WS 的 WSP 的竞争问题，他们都为自己的 WS 设定一个基础的标准水平，同时为客户自选的服务设定相应价格从而保证服务质量(比如平均响应时间)。Zhang(2009)分析了 WSP 们同时设定服务水平和价格，然后拓展至一个序贯决策问题。最后考虑 WS 处理能力，探讨 WSP 们的战略策略的选择。

Cardinalea(2010)研究了 WS 的选择和组建的问题，主要考虑功能性要求和事务处理性质。对用户查询 WS 的事务处理特征以及构建 WS 定义进行建模。该文拓展了彩色说明网络(Colored Petri Net, CPN)方程以便加入事务处理的 WS 特征。提出的 CPN-WS 选择算法。

在语义 WS 方面，Maamar(2006)探讨了策略和语境在构建 WS 过程中的作用。语境支持 WS 的自适应能力的发展而政策确定 WS 在构建中的行为或表现，给出了基于语境的多层次政策(策略)，分为四个层次：组件层次，负责 WS 的定义和处理能力；重建层次(Composite Level)负责 WS 的发现和组合；语义层次(Semantic Level)负责处理 WS 之间语义冲突问题；源层次(Resource Level)负责 WS 整体性能。

在 WS 组合重建方面，因特网正演变成 WS 的载体而不仅是信息的储藏室。许

多公司或组织正倾向于将核心的商业竞争力至于因特网作为一种 WS 的集合。目前一个重要的挑战是如何组装这些 WS 从而创造新的含附加值的 WS 来构建新的 WS(电子商务类服务, 数据存储类服务, )服务, 也即 WS 组合重建问题。Hamadi & Benatallah(2005)提出一个 Petri-net-based 算法来对 WS 重建过程中必须的步骤—控制流建模。该算法的表达性良好足够捕获复杂 WS 组合的语义。在 Axel Martens(2005)对于 WS 组合重建的研究中, 兼容性和相互替代性受到了重视, 该文中提出了“可用性的”概念, 给出对特定的 WS 的直观的和可被证明的性能指标。

## 1.2.2 WS 选址问题

WS 进过多年的发展, 已经具备了相当的理论和技术基础(Beznosov, 2005; Zhang, 2009; Cardinalea, 2010), WS 也逐渐进入到实际应用的阶段。当然, 随着 WS 商用的深入发展, WS 服务提供商(Web Service Provider, WSP)们也开始面临越来越多的同行业的竞争。WS 本身作为一种基于位置的服务(Location Based Service, LBS), 其位置自然关系着 WS 响应时间这一决定 WS 质量的因素。而且, 随着近年来选址问题的研究深入, 研究范围得到不断地扩大, 在各行各业的应用也日趋广泛, 在 WS 方面的研究和应用也越来越多。加上 WS 本身蕴含的巨大潜力和价值, WS 选址问题也逐渐开始引起研究者的注意(Tang, 2005; Sun & Koehler, 2006)。

在包含 WS 的 Web 应用中, WS 的描述、发现、组合等问题要比相应时间来得重要, 而且, 在高速宽带网络出现和计算机处理能力大幅度提高之前, 网络延时只占到响应时间中的很小的部分。所以之前, 大量的研究都关注于 WS 的描述、发现、组合(Maamar 等, 2007)。在研究 WS 响应时间的文献中, 也更多的是考虑如何提高本地的处理速度, 减少排队等待时间和数据上传时间, 往往忽略了网络延时。但随着网络硬件的更新和软件技术的成熟, 其他因素的影响被逐渐减低, 网络延时却越发显得重要, Johansson(2000), Jamin(2001)等提出在高速宽带网络和计算机处理能力骤增的现在网络延时(Network Latency)是一个必须考虑的因素, Johansson(2000)更是指出网络延时与空间距离直接相关。Qiu 等(2001)也在降低网络延时方面做了相应的研究。

Tang 等(2005)等研究了 WSI 的 WS 选址和定价问题,提出 WS 可以是在网络中进行租借的一个业务功能模块,是一个时间敏感(Time-Sensitive)的网络功能组件,分别建立了基于“线性城市”选址模型和更为一般的基于“单元循环”的选址模型。Sun & Koehler (2006)提出的 WSI 的选址问题模型。该模型为 WSI 的服务器选址提供了一个框架,在该模型中,WSI 既作为一个 WSR 通用的接口又需要从独立的 WSP 处获得 WS,在其选址成本的考量中,网络延时是一个重要的部分。区别于服务器选址问题,该模型是一个三层的模型,底层的 WSR,中层的 WSI,以及上层的 WSR,并且假设了 WSR 必须通过 WSI 才能与 WSP 联系。模型的求解采用了 CPLEX 8. Zhang(2009)等指出 WS 的网络传输时间和 WS 的整合对基于 WS 的商务应用提出了很大的挑战,因而,WSP 在发布 WS 的同时会给出相应的“服务水平协议”,规定不同的 WS 服务水平对应不同的价格。其中决定服务水平最主要的因素便是 WS 的平均相应时间。随后 Aboolian(2009)等又从 WSP 的角度考虑 WS 的选址问题,客户信息收集以及每个设施上的服务器的数量(决定服务能力)对其性能和客户满意度有着重要的影响。他们继而提出一个双寡头垄断竞争市场中的分配选址(Location-allocation)模型,在该模型中,网络中每个节点都可以活动所有服务器上的服务,并且每个节点都可以增加或布置新的服务器。目标是最大化 WSP 的利润。

### 1.2.3 选址问题中的时间满意度

WS 作为互联网产业的产物,自然与时间有着密切的关系,Tang 等(2005)已经指出网络响应时间是决定 WS 服务水平的关键因素。自然在 WS 选址问题研究中时间因素是不可回避的。其实时间因素在选址问题中一直占有重要的位置,在很多选址问题中,时间因素决定了客户的服务选择。虽然决定客户选择的因素有很多:质量、价格、品种、外观,不同的产品和服务有着不同的侧重点。但总体而言,时间要么是决定服务质量的因素,要么是左右客户选择的重要因素,Spekman 等(1994)突出了产品或服务可获得性。无疑,时间因素是非常重要的,同样时间满意度对于顾客满意度也尤为重要因素,有时,如果顾客对某种产品或服务的时间满意度很高,即便对其它一些因素不是很满意,仍然有可能选择该公司的产品和服务。

# 华中科技大学硕士学位论文

马云峰(2005)在选址问题研究中引入了时间满意度的概念,并结合心理学等研究成果,给出了时间满意度度量的函数和时间满意度拟合的基本原则和方法。同时,提出了基于时间满意的最大覆盖问题,基于时间满意的集覆盖问题和基于时间满意的两点覆盖问题,并用于求解了巩义市康店伊洛河大桥选址问题。

## 1.3 论文框架与研究内容

本文构建了基于时间满意度的多服务带容量限制的 WS 选址模型,由于该类问题是 NP-困难问题,所以采用启发式算法进行求解,鉴于贪婪算法在求解覆盖选址问题中的综合性能,本文结合模型特点,构造了贪婪算法与动态规划相结合的混合算法进行了求解,论文框架可以参看图 1.1:

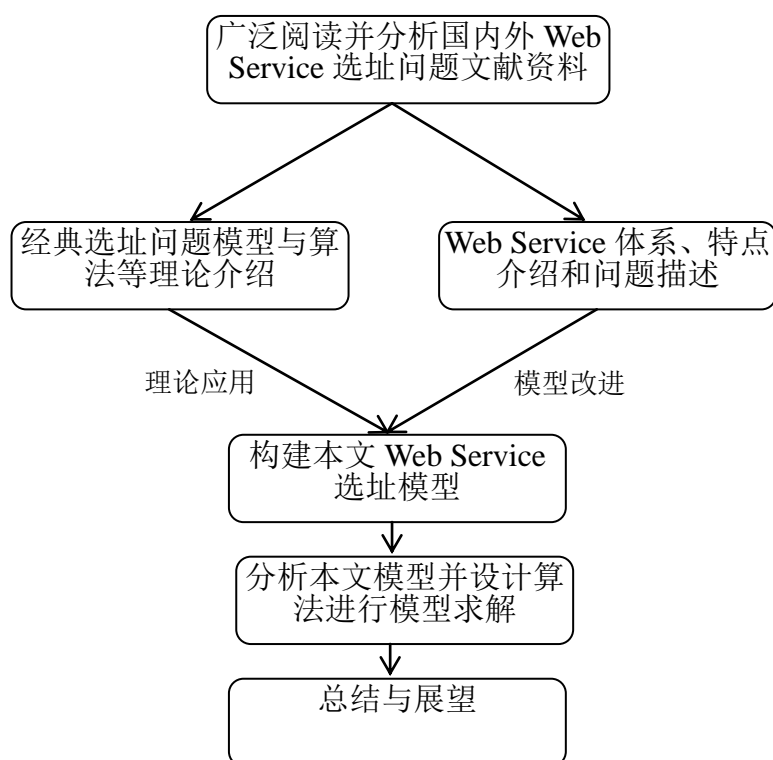


图 1.1 论文框架示意图



# 华中科技大学硕士学位论文

---

在上述框架下，本文被分为了五章，各章的主要内容如下：

第一章介绍课题的背景、研究意义，并对 WS 选址问题研究的国内外研究进展进行了综述，给出了本文的研究主要内容和方法。

第二章对概述了相关的理论，包括一般选址问题，覆盖选址问题及常用算法，覆盖选址问题中国的时间满意度的度量。

第三章主要是本文模型的构建，首先介绍了 WS 体系，接着结合选址问题相关理论建立基础模型，然后结合 WS 选址问题自身特点进行了模型改进同时引入时间满意度建立基于时间满意度多服务带容量限制的 WS 选址模型。

第四章主要是本文模型的求解，在对本文模型进行模型分析的基础上设计了贪婪算法与动态规划法相结合的混合算法，并用于模型求解，获得了比较好的求解效果。同时，本章还分析了算法求解的特点，和可以改进的地方。

最后一章对全文作总结，并提出下一步的研究方向。

## 2 一般选址问题相关理论综述

### 2.1 一般选址问题介绍

早在 17 世纪, Fermat, Torricelli 和 Cavallieri 等就提出基本欧氏空间中位问题(马云峰, 2005)。早期的研究中, 为了求解无权重的 1-中心问题, 早在 1885 年 Chrystal 就提出了著名的几何算法。1909 年, Alfred Weber 研究了单个仓库选址问题, 他需要找出多个客户到所建仓库间的总距离最小, 作为著名的 Weber 问题, 可以算作现代选址问题研究的开端。

后来, 1929 年 Hotelling 研究了竞争环境下的服务站选址问题。1937 年 Weiszfeld 在研究了社区选址问题时, 给出“欧氏空间中位”算法。目前选址问题受到了研究着的高度重视, 掀起这一轮轮研究热潮的, 就是上世纪六十年代中期的 Hakimi, 他首先提出了网络 P-中位选址问题, 该模型被提出来主要用于解决通信网络交换机和高速公路警察局地点的选址问题(Hakimi, 1964)。差不多相同的时间, Cooper(1963,1964)研究并给出了选址一分配(Location-allocation)问题和相应的研究方法, 提出了两种算法进行求解类似问题: 精确极值算法和启发式算法。

#### 2.1.1 国外选址问题研究理论

##### (1) 基本选址模型

传统的选址问题有 P-中位问题、P-中心问题和覆盖问题。后来的选址模型基本都是从这三个经典选址问题发展而来, 而且 Daskin(1995)认为大多数选址问题都是 NP-Hard 问题。基本选址模型有:

##### P-中位问题(P-median Problems)

这类问题主要是研究如何选择 P 个配送中心使得物流网络上总的成本(运输费用, 配送中心建设费用)最低。近年来 P-中位问题仍然是研究的热点, 而且在此基础上产生了很多扩展模型, 如 Berman (2003)等研究了需求与设施的距离决定服务可靠度的问题。

## P-中心问题 (P-center problems)

也即 Minmax 问题, 讨论了如何选择  $P$  个服务中心, 使到某需求点的最近的服务中心的最大距离最小。Kariv & Hakimi(1979)证明了 P-中位问题为 NP-Hard 问题, Levin & Ben-Israel(2004)在算法上做了不少工作, 分析比较了一些著名的问题的基础上, 提出用启发式算法求解大规模 P-中心问题。

## 覆盖问题 (covering Problems)

这类问题又分两种, 也就是集覆盖选址问题(Set Covering Location Problems, SCLP)和最大覆盖选址问题(Maximal Covering Location Problems, MCLP), 两类问题在生活中实际应用非常广泛 (Farahani 等, 2011)。集覆盖问题最早是由 Roth(1969)和 Toregas(1971)等提出, 主要用于解决消防中心选址, 救护车选址等问题。集覆盖问题要求所有的需求点都被覆盖到, 多见于公共设施选址中。而与之不同的是最大覆盖问题, 这类问题考虑到资源有限不要求覆盖所有需求点, 主要考察在给定待建服务站的数目和服务半径的前提下, 如何选出  $P$  个服务站点, 覆盖尽可能多的需求点, 也即是覆盖的需求总量最大。除此之外, 还有部分覆盖 Farahani 等 (2011) 总结了部分覆盖的常见模型。

## (2) 拓展选址模型

扩展型问题是指在以上三种传统选址问题上发展而来, 因为不同的需求和约束衍生出来的一些模型, 常见的有带固定费用和容量限制的选址问题, 截流问题和竞争选址问题。

## 2.1.2 国内选址问题研究现状

### (1) 应用评价方法进行选址研究

后德君等(2004)采用了模糊聚类法用于求解区域物流配送中心的选址问题。何刚和魏连雨(2004)重点研究了第三方物流企业的物流中心选址问题。不仅建立了相关评价指标体系, 还提出了应用人工神经网络进行求解。何亚群和胡寿松(2003)将粗糙集理论运用到了选址问题研究中。此外, 张敏 (2006)应用 DEA 和 AHP 等这类评价方法进行物流选址研究, 主要分析了易腐品仓库的选址问题。

## (2)应用优化模型进行选址研究

主要集中在两大应用背景上：物流中心选址和公共设施选址。在国内研究的比较多的是物流中心选址问题。谢友才(2003)研究了快速反应中心的选址问题，模型采用贪婪算法进行求解。马正元和黄斌(2003)研究了物流配送中心的选址问题，模型采用人工神经网络求解。贺政纲等(2004)研究了综合考虑物流系统和社会经济效益的选址问题，文中借助地理信息系统的相关数据建立了中位模型。

在公共设施选址方面，国内学者也做了一些研究，比如王刊良等(1997)研究了有害物质填埋场的选址问题，建立了多目标选址模型，模型求解采用启发式算法。吕关锋(2001)研究了如何借助城市地理信息系统对消防站进行选址问题研究，模型也是采用启发式算法求解。

## (3)对时间约束考虑较多的选址研究

杨超等(2004)综合时间约束和容量约束研究了截流选址问题，提出了混合整数线性规划模型。李延晖和马士华(2004)建立了考虑时间约束的单源、P个中转点配送系统物流配送中心选址问题，建立了混合整数非线性规划模型(MINLP)，模型求解使用了启发式算法。

## 2.2 基本覆盖选址问题

在多种选址问题模型中，研究得最多的其中一类便是覆盖问题(Covering Problem)。最早的覆盖问题，可以追溯到 Hakimi(1965)提出的高速公路上收费站的选址问题。后来 Toregas 等(1971)，提出了覆盖选址问题的数学模型，应用于解决紧急服务设施(Emergency Service Facilities)选址问题。尽管覆盖问题不是新的模型，但由于其广泛的应用性，特别是在服务选址(医院，学校，银行网点等)和紧急设施(消防、120 急救等)选址问题中的应用，覆盖选址问题仍然受到研究人员的青睐。

在一般覆盖选址问题中，研究者会定义一个服务半径(Coverage Distance，有些研究中使用 Coverage Radius，为了统一，本文中都使用 Coverage Distance)。服务半径是一个预定义的数值，可能是时间，也可能是距离，或者是其他的一些可以度量

# 华中科技大学硕士学位论文

和描绘的属性(Fallah 等, 2009), 为了便于说明, 本文用“距离”统一指代。当客户到一个服务点的距离小于或等于该服务点的覆盖半径的时候, 就认为该客户被覆盖, 该客户选择接受在该服务点的服务。

传统的覆盖问题研究的假设中, 客户不定选址最近的服务站点, 而是在一定距离(这个距离是一些可以度量的关键参数比如时间或者空间距离和路径等等)范围的服务站点都可能被选择, 这个距离 $\gamma$ 就是覆盖问题中的覆盖半径。在一个实际问题中, 我们抽象出一个网络 $G(V, E)$ 中,  $E$ 为边集, 代表路径。 $V$ 为顶点集, 其中 $|V|=m$ 。

$I \in V$  和  $J \in V$  分别是需求点和候选服务站点集合的下标集, 下标  $i \in I$ ,  $j \in J$ 。在这个覆盖问题中, 一个服务站点可能覆盖多个需求点, 也就说说可能多个服务站都覆盖到一个需求点, 构成覆盖需求点 $i$ 的服务站点集  $\xi_i = \{c_{ij} = 1 | \forall j \in J\}$ ,  $\forall i \in I$ , 其中  $c_{ij}$

是 0-1 变量: 
$$c_{ij} = \begin{cases} 1 & \text{如果需求点 } i \text{ 可以被服务站点 } j \text{ 覆盖;} \\ 0 & \text{其他.} \end{cases} \quad (2.1)$$

一般, 服务站点要覆盖某一需求度 $i$ , 则需要让该点到该服务站点 $j$ 之间的距离不超过该服务站点的覆盖半径 $\gamma$ , 这时  $c_{ij} = 1$ , 反之, 该需求点没有被覆盖,  $c_{ij} = 0$ 。实际应用中, 服务站点对所有的需求点 $i \in I$ , 只有一个事先定义的覆盖半径 $\gamma$ 。

覆盖选址问题无疑是非常重要的, 因而覆盖选址的研究和应用也非常广泛, Schilling 等(1993)以及 Farahani 等(2011)分别对覆盖选址问题进行了详细的总结, 前者主要对 1991 年及以前的文献进行总结和归纳, 后者则在其基础上, 尝试总结截止到 2011 年的所有的覆盖选址问题模型。在综合介绍各种覆盖选址问题后, 重点研究了各种问题的求解方法和应用, 同时给出了未来的研究方向。一般而言, 覆盖选址问题主要有两大类: 集覆盖选址问题和最大覆盖选址问题。

## (1) 集覆盖问题

这类问题中, 不管需求量大还是小, 都需要覆盖, 建站数量不定义, 求解目标是在覆盖所有需求点的前提下, 找出建站总成本最小的服务站点集。假设  $f_j$  为候选

点  $j$  的服务站的固定建站费用，再定义决策变量  $X_j$  如下：

$$X_j = \begin{cases} 1 & \text{如果服务站点 } j \text{ 被选中;} \\ 0 & \text{其他.} \end{cases}$$

这样，我们可以得到集覆盖的基本模型：

$$\text{Min} \quad \sum_j f_j X_j \quad (2.2)$$

$$\text{s.t.} \quad \sum_j c_{ij} X_j \geq 1 \quad \forall_i \quad (2.3)$$

$$X_j = 0, 1 \quad \forall_j \quad (2.4)$$

(2.2)式是目标函数，用于最小化建站总成本；(2.3)式是为了保证所有的需求点至少被覆盖一次；(2.4)是对决策变量  $X_j$  的 0-1 约束。

## (2) 最大覆盖问题

最大覆盖问题时区别于集覆盖问题的一个重要分类，集覆盖问题的重要特征就是不考虑需求点的差别，主要是需求量，当然还有需求点的差异化，比如服务水平偏好等其他因素。有的需求点需求量很大，只被一个服务站点覆盖，有的需求量很小，却被多个服务站点所覆盖，不符合企业或者组织的经济效益。在现实生活中，很多时候因为资源有限，无法覆盖所有需求点，这时，我们就面临最大覆盖问题。

为了更好的说明最大覆盖问题模型，需要补充定义变量：

$h_i$  表示需求点  $i$  的需求量；

$P$  计划设立的服务点得数量；

$$Z_i = \begin{cases} 1 & \text{如果需求点 } i \text{ 被覆盖;} \\ 0 & \text{其他.} \end{cases}$$

变量  $x_j$  和  $c_{ij}$  的定义同前，则最大覆盖问题可以用下述模型描述：

$$Max \quad \sum_i h_i Z_i \quad (2.5)$$

$$s.t. \quad \sum_j c_{ij} X_j \geq Z_i \quad \forall i \quad (2.6)$$

$$\sum_j X_j = P \quad (2.7)$$

$$X_j = 0,1 \quad \forall j \quad (2.8)$$

$$Z_i = 0,1 \quad \forall i \quad (2.9)$$

(2.5) 式是使被覆盖的总需求最大；(2.6) 式是保证需求点至少被覆盖一次；(2.7) 式是确保设立服务站的个数为  $P$ ；(2.8) 式和 (2.9) 式是 0-1 约束，分别表示是否设立某服务站和某需求点是否被覆盖。

在覆盖问题的求解方面，我们都知道选址问题包括覆盖问题基本都是混合整数规划问题，这类问题都是 NP-困难问题(Daskin, 1995)，对这类问题的求解比较困难。对于小规模的问题，可以用分支定界法或者割平面法求解，但当问题规模变大，特别是是求解实际问题，常常有数千或者更多的约束和变量，用这类经典算法求解就变得不太现实。也就是说，在当前的计算技术下，用经典算法很难求得最优解。所以，在实际应用中，我们常常使用启发式算法求得最优解的近似解。因为，现实应用中，我们有时不要求得精确的最优解，只要问题的求解的精度达到一定的目标就可以了。所以通常我们需要使用启发式算法对问题进行求解。几种常用的求解覆盖问题的启发式算法有：贪婪算法(Greedy Heuristics)、拉格朗日松弛算法(Lagrangian Relaxation)、遗传算法(Genetic Algorithms, GA)、禁忌搜索算法(Tabu Search)、模拟退火算法(Simulated Annealing)、人工神经网络(Artificial Neural Networks)和蚁群算法(Ant System)等。

## 2.3 覆盖选址问题中时间满意度的度量

在心理学中称时间在人们心理上的综合反映为时间知觉，通过对消费者在时间知觉方面的研究得出了一些有用的结论(罗子明, 2002)，不同的事情，不同的心情，不同的个体都会造成时间体验的不同，一般而言愉快时时间体验加快，消极时时间

# 华中科技大学硕士学位论文

---

体验减慢。由于消费者的时间体验不同而导致消费者在消费产品和服务的过程中产生不同程度满意的现象称为时间满意度。罗子明（2002）将时间满意度分为三大类：可用服务的时间满意度、服务流程时间的满意度和商品使用过程的时间满意度。。

选址问题中的时间满意度一般是流程时间满意度，而客户的流程时间满意度跟时间是成负相关的，覆盖选址时间满意度的研究亦然。所以，需要尽量减少这种时间消耗，从而改善客户的时间体验(马云峰，2005)。为了度量时间满意度，学界提出了很多方法，马云峰(2005)总结了几种主要的时间满意度的度量函数，主要有连续度量函数，离散的度量函数和时间窗口下的度量函数。其中连续的度量函数常见的有线性时间满意度函数、凹凸时间满意度函数、余弦分布时间满意度函数)、降指数 Sigmoid 时间满意度函数、降半哥西分布时间满意度函数，这些度量函数各有各的特点，具体应用时需要根据问题的特点予以选择，当然最好能通过数值模拟的方法进行筛选。

在选址问题中，不同的问题背景，可能需要不同的时间满意度函数。在现实应用中，可以根据用户数据或者问卷调查的数据，通过拟合筛选来获取最适合的时间满意度函数。

在 WS 体系中，我们既要从技术的角度考虑时间满意度，只有满足了技术上的要求，客户才有选择的可能选择相应的 WS；有需要从用户消费心理上去考虑，在满足技术要求的前提下，不同的 WS 响应时间对应着不同的时间满意度。现在商用性质的 WS 已经非常多了，而 WS 的定价的一个重要因素便是 WS 的响应时间。所以在 WS 选址问题的研究中引入时间满意度显得非常有价值。



## 3 WS 选址问题及模型构建

### 3.1 WS 体系介绍

WS 体系结构基于三种角色：网络服务提供者(Web Service Provider, WSP)、网络服务中介(Web Service Intermediary, WSI)和网络服务请求者(Web Service Requestor, WSR)。图 3.1 可以简单表示出这三者之间的关系和相互间的操作：

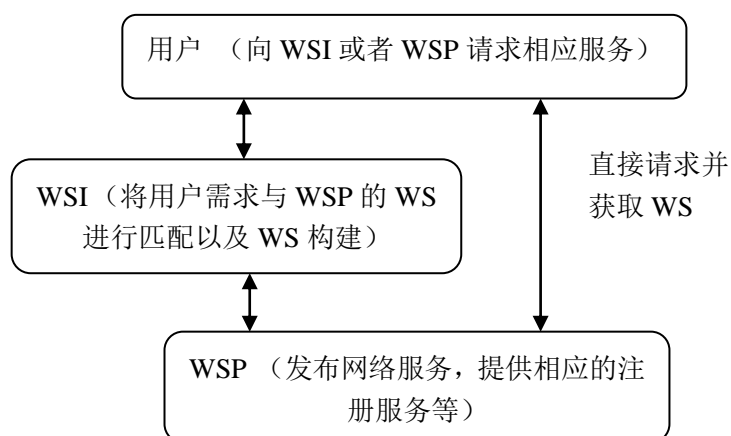


图 3.1 WS 角色及操作示例

其中，服务提供者，是指通过发布 WS 供其他的企业和个人使用的企业或者组织，一般不是个人；服务请求者，是指想要使用 WS 向 WSP 或者 WSI 发送请求，来获取 WS 服务的企业或者组织以及个人。服务注册中心，是介于 WSR 和 WSP 之间的企业或者组织，提供 WS 注册和查询及绑定等功能。

对于正常工作的 WS 体系，必须发生以下三个行为：发布 WS、查询 WS 以及根据 WS 描述绑定或调用 WS。一般而言，WS 的发现和查找主要有三种方式：Index 式，P2P 式，注册中心式，这三种方式各有各的应用背景和优缺点。WS 的这些行为可以单次或反复出现，但一般而言，服务一经绑定(通过注册中心发现或者购买的方式获得)，在相对比较长的一段时间内，WSR 便直接调用 WSP。也就是说可以将 Web Service 服务应用模型抽象为三个层次：WSP 发布 WS，WSI 负责查询，选择 WS，WSR 可以通过 WSI 获得 WS 或者直接从 WSP 处获得 WS。当然 WSI 的角色是相对

的，因为 WSI 在相应 WSR 请求的时候，也许需要调用或请求其他 WS，从 WSP 处或其他 WSI 处获得 WS。

## 3.2 WS 选址问题基础模型

### 3.2.1 问题描述及定义

通过上文的描述，可以发现，在整个 WS 所构成的商务应用中，主要分为两个阶段，一个阶段就是 WS 发现，不论是哪种类型的 WS 发现方式都需要 WSP 在发布 WS 时提供相应的描述以供人为和机器智能地去发现，因而现阶段有不少学者在 WS 的发现和语义描述(syntax description)方面做了不少研究(Zakaria Maamar, 2006)。

然而，但这不是本文的重点，因为 WS 的发现一般而言只是一次性的行为，在大部分商业应用中，有更多的方式(包括广告)让 WSR 知道 WSP 所提供的 WS，而对于 WSR 而言，在选择 WSP 的时候，并不是仅仅需要考虑哪些 WSP 更容易被发现就够了，相反，他们在选择的时候会更加注重 WS 的质量(响应时间，网络延时，稳定性，可靠性等等)，其中响应时间占有主要作用(Tang 等, 2005)。

而且，在现实环境下，一旦某个应用程序发现并且调用一个 WS，或者应用程序提供商购买了 WSP 的 WS，应用程序就会立即记录该服务的网络地址和调用参数等信息，第二次调用 WS 时便不再需要 WS 发现，如图 3.2 所示。

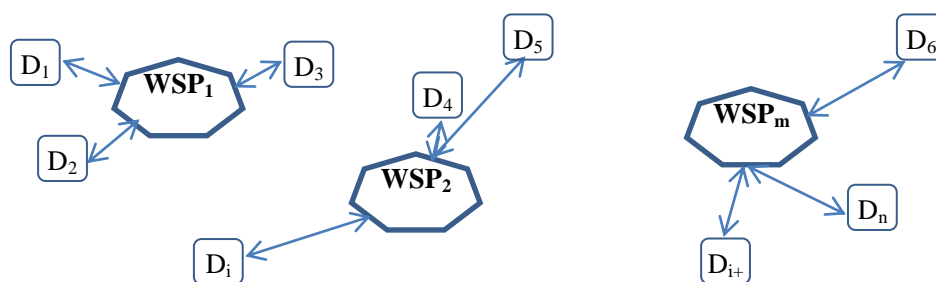


图 3.2 Indexes 类型的 WS 网络

注：在一个 WS 网络中，所有节点都是联通的，就像 Internet 一样，只要连接了 internet，那电脑就可以通过 Internet 连接到上面所有的电脑。又由于 WSR 在选择了可用的 WS 之后(系统选择或者人工选择)，WSR 便会放弃其他的 WSP 的服务，但是 WSP 可以同时为多个 WSR 提供服务。也即 WSR 对 WSP 之间是一对多的关系。

所以，一个 WS 如何被发现并不是 WSP 成败的关键，提供较高质量的 WS，并

让 WSR 在多种选择中选中才是 WSP 更应关注的问题。在影响 WS 诸多因素中，响应时间和网络延时是 WSR 最关注的因素，而这部分刚好是可以通过科学的选址来提高的。又由于本文考虑的是 WS 的选址问题，其重点在于在一个较长的时间内优化，所以暂时不考虑 WS 的发现时间。所以本文重点考虑的就是如何通过合理设置网络中服务器的位置，以减少整体网络响应时间和网络延时。

为了便于入手，我们先考虑一个比较简单的问题：某 WS(Web Service, WS)提供商 (Web Service Provider, WSP)决定在某一区域的网络上发布一个新的 WS，该网络中有众多的需求点，每个需求点对该 WS 都有一定的需求量。另外，该网络中有一些网络节点的基础设备比较完善：是处于区域性的网络中枢中，网络流量比较大，有比较好的网络设施可以假设 WS 服务器(Server)或者有现成的服务器可以租赁来发布 WS。每个服务器都可以为在一定有效服务半径(WSR 可以接受的总体响应时间)内，但不管是租赁别人的 WS 服务器或者是自己设置 WS 服务器(后文统称为设立服务站点)，都需要相当的投资。

为了便于建模，我们将上述网络抽象为一个无向连通图  $G(V,E)$ ， $V$  为各个网络中节点， $E$  是各个节点之间的“网络距离”。有需求点集  $D \subseteq V$ ， $D \neq \emptyset$  和备选设施点集  $F \subseteq V$ ， $F \neq \emptyset$  可以用下图表示：

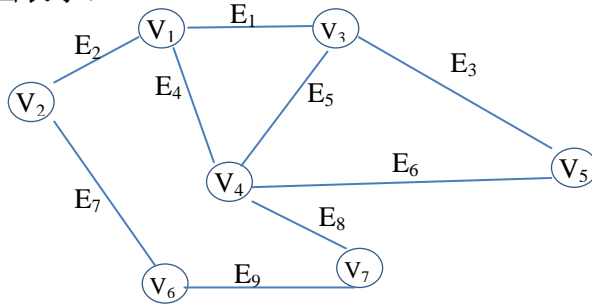


图 3.3  $G(V,E)$ 无向连通图网络示意图

由于本文研究的是 WS 选址问题，所以我们用响应时间来表示“网络距离”，为了计算响应时间，并基于 WS 中避免冗余和重复调用的现实，我们需要做几个假设：WSR 对 WS 的调用再一次应用中只会出现一次；WS 与 WS 之间是相互独立的，不存在相互调用。

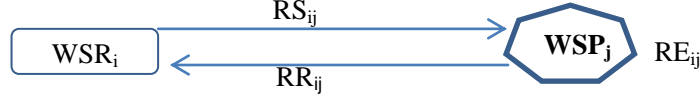


图 3.4 WS 调用示意图

注:上图描述了第  $i$  个 WSR 对第  $j$  个 WSP 发出请求并接受返回结果的过程和响应时间构成,  $RS_{ij}$  表示请求到达的时间,  $RE_{ij}$  表示请求在本地处理的时间,  $RR_{ij}$  表示请求返回的时间。

这样,一次 WSR 对 WS 的调用完整过程可以用图 3.4 表示,响应时间  $t_{ij}=RS_{ij}+RE_{ij}+RR_{ij}$ , 其中,忽略大数据量传输的情况(WS 架构体系中明确要求减少数据量的传递,数据型 WS 会有专门的高速网络,因而可以这样假设)  $RS_{ij}$  和  $RR_{ij}$  是可测的和已知的。而  $RE_{ij}$  对于相同的服务是一定的,且是可知的。

这样我们便将所面临的问题,转化为一个覆盖问题了:在给定的网络  $G(V,E)$  中,  $V$  为顶点集,  $|V|=m, E$  为边集。  $I \in V$  和  $J \in V$  分别用于表示需求点和候选服务站点集合的下标集,下标  $i \in I$ ,  $j \in J$ , 在不同的需求点  $i \in I$ , 具有不同的需求量  $h_i$ 。在这个覆盖问题中,一个服务站点在给定的覆盖半径  $\gamma$  内可能覆盖多个需求点,在不同的顶点  $j \in J$  (公司已经自有了服务器或者通过租借别人的服务器), 设立服务站点具有不同的成本  $f_j$ 。定义 0-1 变量  $c_{ij}$ , 决策变量  $X_j$ :

$$\text{其中, } c_{ij} = \begin{cases} 1 & t_{ij} \leq \gamma \\ 0 & t_{ij} > \gamma \end{cases}, \quad (3.1)$$

$c_{ij}$  等于 1 时表示被覆盖, 为 0 时表示没有被覆盖,

$$X_j = \begin{cases} 1 & \text{如果服务站点 } j \text{ 被选中;} \\ 0 & \text{其他.} \end{cases} \quad (3.2)$$

我们知道覆盖问题主要被分为两种, 集覆盖问题和最大覆盖问题, 这里为了区

别，我们需要从 WSP 的角度考虑，通常 WSP 可能有两种，一种以政府为代表的关注于社会公益的 WSP，另一种是一股份制公司为代表，追求公司商业利益的 WSP。这两类 WSP 在考虑 WS 的选址的时候可能选择不同的模型。

### 3.2.2 WS 的覆盖选址基础模型

#### (1) WS 集覆盖问题

关注社会公益的 WSP 所提供的 WS 需要覆盖某个国家或者地区，这时候我们可以得到一个集覆盖问题。WS 集覆盖问题模型可以表示为：

$$\text{Min} \quad \sum_j f_j X_j \quad (3.3)$$

$$\text{s.t.} \quad \sum_j c_{ij} X_j \geq 1 \quad \forall_i \quad (3.4)$$

$$X_j \in \{0,1\} \quad \forall_j \quad (3.5)$$

(3.3) 式是目标函数，用于最小化建站总成本；(3.4) 式保证所有的需求点至少被一个服务器覆盖一次；(3.5) 式是对所有决策变量的 0-1 约束。该模型解决的是要覆盖所有的需求点而需要设立的服务站点的最小数量。

当然，WS 选址问题从集覆盖问题的角度考虑，对于大部分的商业应用是不太合适的，首先，大部分商业应用中，企业是没有那么多的资金去设立足够多得服务器以覆盖所有的需求点的。其次，由于每个需求点可能具有的不同的需求量，这里的需求量的差距有时候是非常巨大的，比如我国东部沿海发达地区，互联网覆盖率高，互联网企业多，基于 WS 的网络应用多且复杂，而偏远地区本身可能互联网的覆盖率都不高，互联网企业也很少，基于 WS 的网络应用则更少，所以这两处的需求量的差距可能是跨数量级的。因而集覆盖中，为了覆盖所有点，有时需要为了覆盖需求量很小的需求点而去设立 WS 服务站点，这显然不具有成本效益、所以，对于大部分的商业应用而言，使用最大覆盖模型更具有实际应用价值。

## (2) WS 最大覆盖问题

WS 最大覆盖(MCLP)模型是考虑如何在已有的多个服务器上的  $P$  个服务器上部署 WS, 使得服务覆盖到的 WS 需求总量最大的问题, 其中  $P$  已知, 各个 WS 的覆盖半径 (这里是指 WS 本身的响应时间)。在描述 WS 最大覆盖问题模型之前, 首先定义下面几个变量:

$P$  计划设立的服务站点得数量;

$$Z_i = \begin{cases} 1 & \text{如果需求点 } i \text{ 被 WS 服务站点覆盖;} \\ 0 & \text{其他.} \end{cases} \quad (3.6)$$

其他变量的定义同前, 则按照传统的覆盖问题模型, 我们可以得到 WS 最大覆盖模型如下:

$$\text{Max} \quad \sum_i h_i Z_i \quad (3.7)$$

$$\text{s.t.} \quad \sum_j c_{ij} X_j \geq Z_i \quad \forall i \quad (3.8)$$

$$\sum_j X_j = P \quad (3.9)$$

$$X_j \in \{0,1\} \quad \forall j \quad (3.10)$$

$$Z_i \in \{0,1\} \quad \forall i \quad (3.11)$$

(3.7) 式是目标函数, 使被 WS 站点覆盖到的总的需求最大; 约束(3.8)是保证至少被覆盖一次的需求点才算被覆盖次; 约束式(3.9)是 WS 服务站的总数约束; 约束(3.10) 和约束(3.20)是对决策变量是否选中某服务站以及是否覆盖某需求点的 0-1 约束。对比两个模型, 加上之前的分析, 不难发现 WS 最大覆盖问题更具有研究价值, 所以后文将主要用 WS 最大覆盖模型解决 WS 的选址问题。

## 3.3 基于时间满意度的带容量限制多服务 WS 选址模型

### 3.3.1 考虑多种 WS 和服务容量限制

上节中，我们已经得到了一个比较简单的单服务最大覆盖模型，该模型说明了 WS 选址问题中的一些简单的规律，是经典最大覆盖模型在 WS 选址问题中的一个简单应用，但是在 WS 选址问题中，该模型是有着明显的缺陷的。首先， $\sum_j c_{ij} X_j \geq Z_i$  的约束保证了需求点被覆盖，却没有保证 WS 体系中一个需求点只能被一个服务站点接受服务的特征。其次，所有的服务器的容量都是不可能是无限的，而模型中显然没有考虑这一约束，现实中，如果 WSR 的请求超过了服务器的容量限制，对应的请求就需要等待处理而产生大量等待时间，这是 WS 的商务应用中所不允许存在的，故而，本文将考虑容量限制问题。也就是说，用户 WSR 一旦选择了一个 WSP，那么该 WSP 就需要保证提供相应的服务水平(主要是响应时间)，由于现实中服务器都是有容量限制的，也即一定时间只能正常处理一定数量的网络请求，超出的部分，必然会出现响应时间延时，这对于企业是无法容忍的，所以本文考虑“最坏”的情况，假设 WSR 的请求都是同时到达

而且，很多时候，我们所面临的问题可能更加复杂。一个企业可能提供的服务有多达上百个，比如 [www.webxml.com.cn](http://www.webxml.com.cn) 公司，其服务器上可以部署的 WS 有很多个，而相同的需求点对于不同服务的需求量也是不尽相同，为了更好的贴近实际应用，我们需要考虑一个带容量限制的多服务的最大覆盖问题。经过适当的抽象，我们实际面对的是下面的问题：

在给定的网络  $G(V, E)$  中， $V$  为顶点集， $|V|=m$ ， $E$  为边集。 $I \in V$  和  $J \in V$  分别用于表示需求点和候选服务站点集合的下标集。增加下标集  $K$ ，表示 WSP 提供的 WS 种类，下标  $i \in I$ ， $j \in J$ ， $k \in K$ 。在不同的 WS 具有需求量  $h_{ik}$ 。增加容量约束  $Cap_{jk}$  表示服务站点  $j \in J$  所能提供的第  $k$  种 WS 最大的容量，在这个覆盖问题中，一个服

# 华中科技大学硕士学位论文

务站点在给定的覆盖半径 $\gamma$ 内可能覆盖多个需求点, 但一个需求点对某个特定的服务的需求只能被最“适合”(距离该需求点时间最短)的服务站点所覆盖。定义新的 0-1 变量 $C_{ijk}$ , 表示需求点 $i \in I$ , 对第 $k \in K$ 种服务的需求被第 $j \in J$ 处的服务站所满足:

$$\text{其中, } C_{ijk} = \begin{cases} 1 & t_{ijk} \leq \gamma, t_{ijk} = \min_j \{t_{ijk}\} \\ 0 & t_{ijk} > \gamma \end{cases}, \quad (3.12)$$

和决策变量 $X_j$ :

$$X_j = \begin{cases} 1 & \text{如果服务站点} j \text{被选中;} \\ 0 & \text{其他.} \end{cases}$$

$$\text{Max} \quad \sum_i \sum_j \sum_k h_{ik} C_{ijk} \quad (3.13)$$

$$\text{s.t.} \quad \sum_j C_{ijk} \leq 1 \quad \forall i, \forall k \quad (3.14)$$

$$\sum_i h_{ik} C_{ijk} \leq \text{Cap}_{jk} \quad \forall i, \forall k \quad (3.15)$$

$$\sum_j X_j = P \quad (3.16)$$

$$C_{ijk} - X_j \leq 0 \quad \forall i, \forall k \quad (3.17)$$

$$X_j \in \{0, 1\} \quad \forall j \quad (3.18)$$

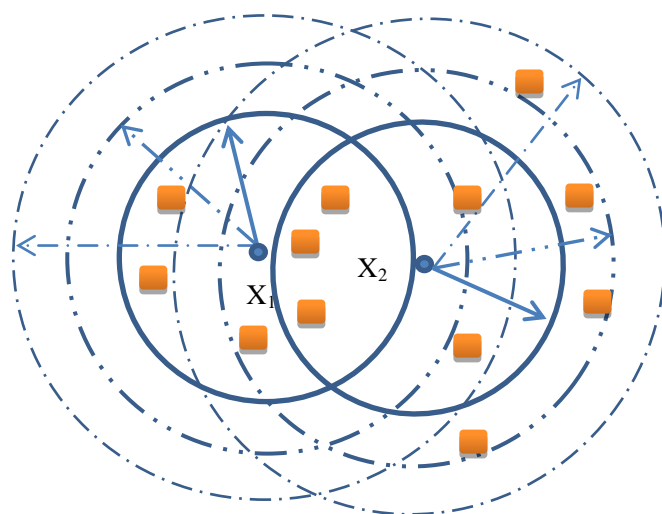
(3.13)式表示目标函数, 使被 WS 服务站点覆盖到的需求总量最大; (3.14)式是保证 WS 的需求点 WSR 最多只能被一个服务站点所覆盖; (3.15) 式, 是容量限制, 每个服务站点和每种 WS 都有一定的容量限制; (3.16) 式是要求部署 WS 的服务站点总数为 P; (3.17)式表示只有部署了 WS 的服务站点才能用于覆盖需求点 WSR; (3.18)式是对决策变量都是 0-1 约束, 代表在某个服务站点是否部署 WS。



## 3.3.2 考虑时间满意度构建最终模型

前文中，我们已经得到带容量限制的多服务 WS 选址模型，该模型本质上是一个最大覆盖模型，属于最大覆盖问题(MCLP)的研究范畴。MCLP 的关于覆盖半径的假设有两个主要的缺点，一是在一个最大覆盖问题中只有一个覆盖半径，这忽略了在需求差异化已然成为当今的主流，各种企业和个人都在强调个性差异化以突出自己的服务水平；另一个缺点是覆盖半径固定了，覆盖半径内的都被覆盖，覆盖半径之外都没有被覆盖，这一假设很不人性化，而且略显牵强，这样的问题假设下，求解的结果会与现实有着比较大的误差。特别是在 WS 体系中，每个商务应用对于响应时间的要求是非常不一样的，就拿股票数据相关的 WS 来说，注重即时性金融产品或者产品分析等应用，对于其响应时间要求非常高，但对于一般的模拟炒股等等娱乐性应用而言，响应时间则显得没有那么重要，而且相同的需求点，其不同阶段的商务应用对 WS 时间的要求也是不同的，比如之前所说的模拟炒股这种娱乐性应用，随着其用户的增加，他可能需要开始关注用户体验，自然对于 WS 的响应时间会提高要求。

所以，本文将引入基于时间满意的最大覆盖问题(TSBMCLP)，来克服这两大假设的不足。图 3.5 说明了传统的最大覆盖问题(CMCLP)问题和 TSBMCLP 之间的差别。



# 华中科技大学硕士学位论文

图 3.5 CMCLP 与 TSBMCLP 覆盖示意图

在图 3.5 中，勾画的距离是一种抽象意义上的时间满意度，越靠近 X1 和 X2 的时间满意度越高，为了方便说明，本文将时间满意度定义成离散的，即需求点对时间的满意度水平划分为非常不满意、不满意、一般、满意、非常满意等五种等级，图 3.5 中为了便于说明就只划出了三个离散的覆盖半径，由短到长分别为满意，一般满意，不满意。图中有 12 个需求点和两个候选服务站点 X1, X2，如果只考虑非常满意的覆盖情况，就是覆盖半径只有一个，则会选 X1 点，因为它覆盖 6 个需求点。但如果我们再考虑一般满意的情况多，也就是有两个覆盖半径，则我们会选择 X2 因为它不仅覆盖了 5 个非常满意需求点，覆盖了 4 个一般满意的需求点这样，总的需求的时间满意度会有所增加。所以本文将在模型中引入时间满意度对模型进行进一步的优化，以期获得更好的解。

在上文中，我们已经建立带容量限制的多服务 WS 选址模型，在 Sun 等(2006)和 Aboolian 等(2009)的基础上，增加了容量限制，同时，为了更加贴合 WS 本身的特点，增加了新的约束，一个需求只能被一个 WS 服务站点服务。当一个需求被多个服务站所覆盖的时候，需求点可以选择对自己最“合适”的服务，在之前的模型

$$C_{ijk} = \begin{cases} 1 & t_{ijk} \leq \gamma, t_{ijk} = \min_j \{t_{ijk}\} \\ 0 & t_{ijk} > \gamma \end{cases}$$

该式实际上包含了一个假设，就是客户会选择时间最短的 WS，这是一个理想的情况，在实际的商务应用的情况下，客户的选择的服务水平会复杂许多，包含品牌，价格，时间，稳定性等诸多因素。而且不同的客户对于服务水平有着不同的期望，有的客户对服务水平要求高，愿意付出比较高的成本；也有的在乎成本，愿意选择比较低价的服务，也就是说，在经济高度繁荣的今天，客户需求的差异化程度非常高。在传统的覆盖问题中，往往忽略了这种需求差异化，忽略了客户根据自己需求差异化(对服务水平要求的不同)，来自主选择服务的情况。

WS 作为信息技术产品，其中最主要的因素便是价格和时间。在上节内容中，我们讨论了时间满意度在客户满意度方面的关键作用，刚好在决定 WS 服务价格的诸

# 华中科技大学硕士学位论文

多因素中时间因素是最重要的一个因素。而且，随着互联网产业的深入发展，企业间竞争日趋激烈，而且产品同质化程度越来越高，时间因素由于地理上的客观存在，不会随着这个趋势而改变，但除了时间外地其他因素可能都趋于同质化。所以，本文本着抓住关键因素又简化问题的想法，我们将客户的时间满意度作为服务水平的决定因素。

也就是说，客户在可获得的 WS 的服务站点中做选择的时候，会根据时间满意度作判定。而下文在对客户需求差异化进行描述的时候也会使用时间满意度。这样本文定义“有效覆盖”的概念，在服务站点覆盖的基础上，客户再根据各自的时间满意度的需求差异化水平(主要是技术上对响应时间的要求)进行选择，只有两者最终匹配才算是有效覆盖。

这样，我们可以得到新的问题，在给定的网络  $G(V, E)$  中， $V$  为顶点集， $|V|=m$ ,  $E$  为边集。 $I \in V$  和  $J \in V$  分别用于表示需求点和候选服务站点集合的下标集。增加下标集  $K$ ，表示 WSP 提供的 WS 种类，下标  $i \in I$ ， $j \in J$ ， $k \in K$ 。在不同的需求点对第  $k$  种 WS 服务具有需求量  $h_{ik}$ ，而且，由于需求差异化，需求点  $i$  对于不同的服务的时间满意度的要求  $d_{ik}$ ， $d_{ik} \in D_l$ ， $l \in L$ ， $D_l \subseteq \{(0, D_1], (D_1, D_2], (D_2, D_3], \dots\}$  是正实数空间  $\mathbb{R}^+$  的一个分割。容量约束  $Cap_{jk}$  表示服务站点  $j \in J$  所能提供的第  $k$  种 WS 最大的容量，在这个覆盖问题中，一个服务站点在给定的覆盖半径  $\gamma$  内可能覆盖多个需求点，但一个需求点对某个特定的服务的需求只能被满足其时间满意度的服务站点所覆盖，定义为有效覆盖。时间满意度函数  $F(t_{ijk})$  定表示客户对特定时间  $t_{ijk}$  的满意度。另外，定义决策变量  $X_j$  和变量  $C_{ijk}$ ，并作如下调整：

其中，

$$C_{ijk} = \begin{cases} 1 & \text{在 } j \text{ 处的服务站覆盖 } i \text{ 点对 } k \text{ 种服务的需求} \\ 0 & \text{其他} \end{cases},$$

$$X_j = \begin{cases} 1 & \text{如果服务站点} j \text{被选中;} \\ 0 & \text{其他.} \end{cases}$$

另外，考虑到客户的自主选择性，我们定义 0-1 变量  $\delta_{ijk}$ ，表示客户选择，表示客户根据自己的时间满意度的水平选择服务站，即当该服务站的时间满意度与用户的时间满意度的需求处于同一水平时，才会被 WSR 选用。可能有多个服务站满足需求，由于这些服务站点在对客户的时间满意度上的效用是相同的，因而这些服务站点对于该需求被认为是无差异的。而在实际的 WS 商务应用中，用户购买了 WS 服务之后，具体的分配是由 WSP 完成的，用户只是根据 WSP 提供的参数进行 WS 的绑定，因而可以从这些服务站点中任意分配给用户。

$$\delta_{ijk} = \begin{cases} 1 & j \in \{j | F(t_{ijk}) \in D_l, d_{ik} \in D_l; \forall l \in L\} \\ 0 & \text{其他} \end{cases} \quad \forall i, \forall k$$

这样我们可以得到新的模型的目标函数

$$\text{Max} \quad \sum_i \sum_j \sum_k h_{ik} F(t_{ijk}) C_{ikj} \delta_{ijk}, \text{ 该目标函数以在有效覆盖的前提下获得最大}$$

的客户时间满意度为目标，因为最大的时间满意度意味着可以带来相应的商业利益，该目标函数既反映了服务站点对需求点的覆盖，也反映了需求点对服务站点的选择。

为了保证每个具体需求只被一个服务站服务，所以加上约束：

$$\sum_j C_{ijk} \delta_{ijk} = 1 \quad \forall i, \forall k$$

由于这是一个带容量限制的模型，故而需要容量约束，所有在同一服务站接受的某一特定服务的总量小于该服务站点所能提供的该服务的总量：

$$\sum_i h_{ik} C_{ijk} \delta_{ijk} \leq \text{Cap}_{jk} \quad \forall i, \forall k$$

准备建立的服务站的数量

$$\sum_j X_j = P$$

由于网络都是互通的，在考虑时间满意度的情况下，所有需求点理论上都是可以接受服务的。但是如果某设施候选点没有部署 WS 服务器，那就不是服务站，自然所有连接到该服务站的需求的时间满意度都将为 0。

# 华中科技大学硕士学位论文

$$C_{ijk} - X_j \leq 0 \quad \forall i, \forall k$$

在时间满意函数的选择上，由于本文的重点不在于找出最好的时间满意度函数。加之 WS 定价过程中，常常主要根据时间要素所决定的服务水平进行定价(Tang 等，2005)，所以本文使用离散的时间满意函数，根据马云峰(2005)的表示，该时间满意度函数可以表示如下：

$$F(t_{ijk}) = \begin{cases} 1 & t_{ijk} \leq T_k^1 \\ S_k^m & t_{ijk} \in (T_k^{m-1}, T_k^m], m \in \{2, 3, \dots, M\} \\ 0 & t_{ijk} > T_k^m \end{cases}$$

其中  $i \in I$ ,  $j \in J$ ,  $k \in K$ 。  $S_k^m$  是第  $k$  种服务的第  $m$  个时间满意度区间的时间满意度值。从而，我们可以得到基于时间满意度的多服务带容量限制的 WS 选址模型如下：

$$\text{Max} \quad \sum_i \sum_j \sum_k h_{ik} F(t_{ijk}) C_{ikj} \delta_{ijk} \quad (3.19)$$

$$\text{s.t.} \quad \sum_j C_{ijk} \delta_{ijk} \leq 1 \quad \forall i, \forall k \quad (3.20)$$

$$\sum_i h_{ik} C_{ikj} \delta_{ijk} \leq \text{Cap}_{jk} \quad \forall i, \forall k \quad (3.21)$$

$$\sum_j X_j = P \quad (3.22)$$

$$C_{ijk} - X_j \leq 0 \quad \forall i, \forall k \quad (3.23)$$

$$X_j \in \{0, 1\} \quad \forall j \quad (3.24)$$

$$\delta_{ijk} = \begin{cases} 1 & j \in \{j | F(t_{ijk}) \in D_l, d_{ik} \in D_l; \forall l \in L\} \\ 0 & \text{其他} \end{cases} \quad \forall i, \forall k \quad (3.25)$$

$$F(t_{ijk}) = \begin{cases} 1 & t_{ijk} \leq T_k^1 \\ S_k^m & t_{ijk} \in (T_k^{m-1}, T_k^m], m \in \{2, 3, \dots, M\} \\ 0 & t_{ijk} > T_k^m \end{cases} \quad (3.26)$$

本文的模型较之于以往的模型有几点主要的不同，第一：引入了时间满意度的概念，使得覆盖范围得到了扩充，结合 WS 自身的时间特性，也使这种覆盖范围上

# 华中科技大学硕士学位论文

---

的增大有了现实意义。第二：根据 WS 的商业应用自身特点增加了一个需求只能被一个服务站服务的约束，更加贴近实际应用。第三，充分考虑需求点的对于服务水平(时间满意度)的差异化，并让需求点根据自身的需求特点(对时间满意度的偏好)自动选择服务站，避免了传统覆盖问题中，客户被动覆盖的情况。第四，本文模型具有比较好的拓展性，该模型中引入了时间满意度函数以贴近实际生活，但本文的时间满意度函数只是以时间为唯一参考量，现实应用时，可以构建稍微复杂的时间满意度函数以求进一步符合实际情况；其次，现实生活中，可能有些 WS 对网络资源占用非常小，也就是说几乎不存在服务站点的容量限制，那只需要将本模型的约束式(3.21)去掉便可以了；再者，本文考虑到客户的自主选择性，定义了 0-1 变量  $\delta_{ijk}$  用以模拟 WSR 的自主选择，本文为了简化问题，就只是要求网络响应时间小于 WSR 技术上的要求便可，实际应用中可能更加复杂，可以通过修改式(3.25)来定义新的 WSR 选择模式。

综上所述，本文所建模型不仅贴合了实际问题，而且保证了模型有较好的拓展性，便于实际的应用或参考。

## 4 WS 选址模型求解

### 4.1 模型分析及算法设计

#### 4.1.1 模型分析

观察本文模型，可以发现跟传统最大覆盖模型有所不同，本文中的模型事先假设需求点都被“覆盖”了，只是覆盖的程度不一样，这样更加符合互联网及 WS 体系的特点，因为 WSP 常常会对 WS 进行分级定价，刚好可以与客户的时间满意度相对应。在传统最大覆盖问题中，是不考虑完全覆盖所有需求点的，而且覆盖半径是固定的，通常是一个数值；而且，本文添加了  $\delta_{ijk}$  变量，给用户(需求点)一个主动选择的权利，根据覆盖后的时间满意度的差别，让客户可以自己“选择”自己合适的服务站点，使问题更加贴近现实情况。本文的模型试图最大化覆盖到的需求点的时间满意度：

$$Max \quad \sum_i \sum_j \sum_k h_{ik} F(t_{ijk}) C_{ikj} \delta_{ijk}$$

式中  $F(t_{ijk})$  为 0，也即时间满意度为 0 时，也就等于是没有覆盖上，所以也类似于传统最大覆盖问题。为了更加贴近传统最大覆盖问题，我们令  $f_{ijk} = F(t_{ijk})$ ，则模型目标函数可以表示为：

$$Max \quad \sum_i \sum_j \sum_k h_{ik} f_{ijk} C_{ikj} \delta_{ijk}。$$

这样就很像 P-中位问题了，只不过 P-中位问题是最小化值，上述模型是求最大化目标值。因而，可以在目标函数前加上一个负号，就可以把基于时间满意度的多服务带容量限制的选址问题转化为 P-中位问题，也就是说 TSBMCLP 是 CMCLP 更一般的情况。特别是，在满意度划分的时候，如果只考虑时间满意度最大的情况下，那这个问题便成了传统最大覆盖选址问题了(CMCLP)。

#### 4.1.2 算法设计

如前文所述，加入了基于时间满意度的最大覆盖模型最终可以转化为 P-中位问题，所以本文的模型也是 NP 困难问题，因此，无法借助商业优化软件进行求解，为

# 华中科技大学硕士学位论文

了求解规模较大的问题，本文决定采用启发式算法进行求解。

在覆盖问题的求解上，Grossman 和 Wool(1997)曾进行了大量的数值实验，用于对比九种常用于求解集覆盖问题的启发式算法，其中随机贪婪算法(R-Gr)、简单贪婪算法(S-Gr)和转换贪婪算法(Alt-Gr)在几乎所有问题中都是最好的前四种算法之一，所以本文相信使用贪婪算法可以获得比较好的求解效果。故而本文将使用贪婪算法对该问题进行求解。

## (1)基于贪婪算法的模型求解算法设计

贪婪策略：每次都增加一个能使目标函数增加量最大的服务站点。也即将整个选址问题分为多步完成，每次的目标就是从待选服务站点中选出一个给目标函数带来提升最大的，由此本文设计了贪婪算法如下：

**Step 1** 初始化参数： $k$  种服务的本地处理时间；在不同需求点  $i$  对第  $k$  种 WS 服务的需求量  $h_{ik}$ ，以及相应对时间满意度的需求  $d_{ik}$ ；需求点  $i$  到服务站点  $j$  的网络延时(network latency)；服务站点  $j$  对第  $k$  种服务的容量限制  $Cap_{jk}$ ；WS 技术上要求划分  $d_{ik} \in D_l, l \in L, \mathbb{Z} = \{(0, D_1], (D_1, D_2], (D_2, D_3], \dots\}$ ；

**Step 2** 定义已选服务站点下标集  $F = \phi$ ，未选服务站点下标集  $F' = J$  和记录已被覆盖需求点的三维矩阵  $Cov_{ijk}$ ；计算各种服务的阈值  $T_k^m$  及对应的时间满意度；计算  $i$  到服务站点  $j$  的关于第  $k$  种服务的网络响应时间  $t_{ijk}$  及相应时间满意度  $f_{ijk}$ ；计算所有未选服务站点  $j, j \in F'$  的有效覆盖子集  $\omega_j$

**Step 3** 根据未被覆盖的需求点集  $F'$  及未选服务站点  $j, j \in F'$  的有效覆盖子集  $\omega_j$  计算各个服务站点  $j$  的最大覆盖的子问题，记录下所有子问题的最优解包括最大函数值  $\max_j$  和其最优解  $solution_j$  (该服务站点最终有效覆盖到的需求点集)。

**Step 4** 对  $\max_j$  进行排序，选择其中对目标函数影响最大的服务站点  $j$ ，并归入已选服务站点集  $F$ 。同时更新  $Cov_{ijk}$ ， $F'$ ，同时检查  $solution_j$ ，如果有影响，则需要

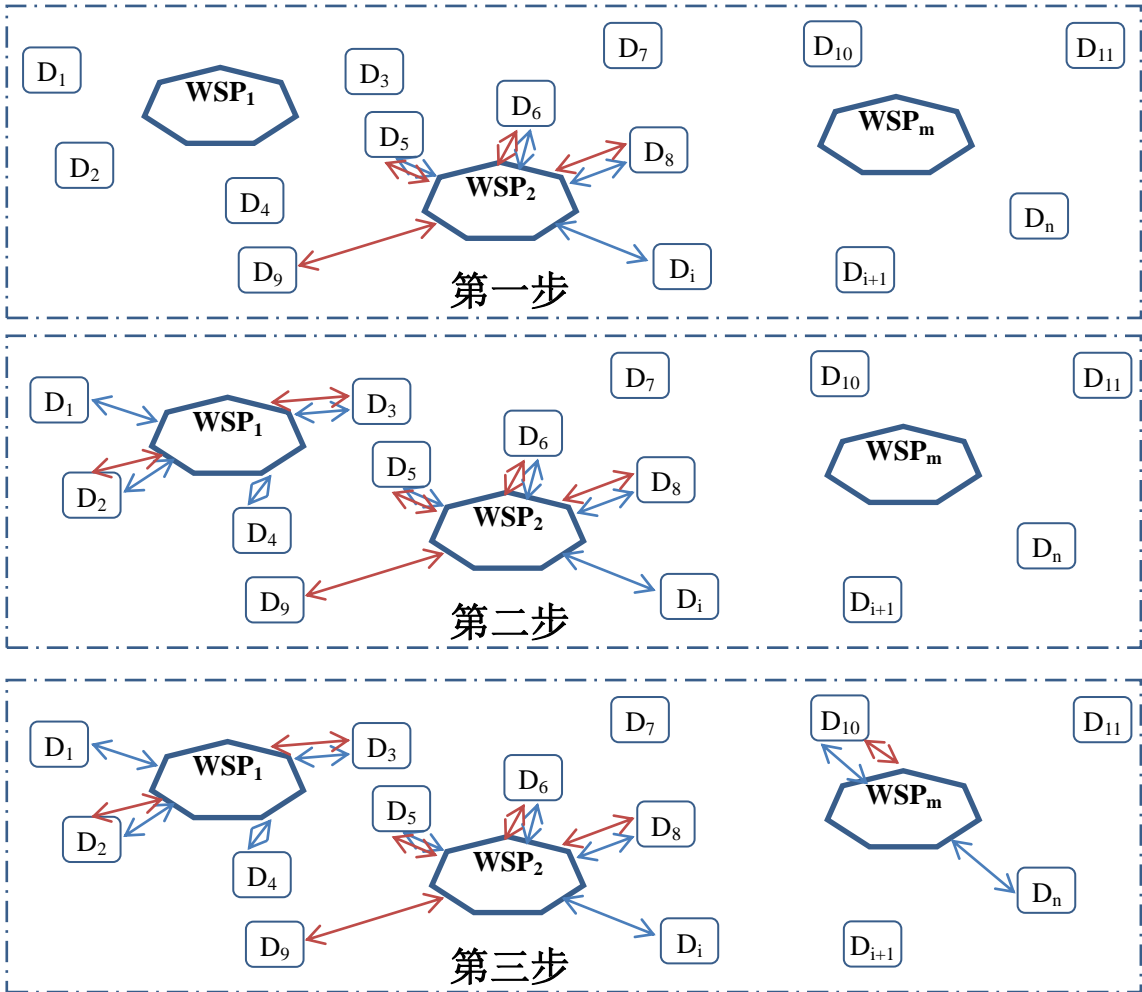


重新计算相应的最优化问题。并更新  $j, j \in F'$  的有效覆盖子集  $\omega_j$  计算各个服务站点  $j$  的最大覆盖的子问题。

**Step 5** 判定是否终止，达到如下任一条件，即终止迭代，输出结果：已选的服务站点数达到了  $P$  个，或者所有的需求点都已经被覆盖到。

**Step 6** 若没有到达终止条件，转到 step4.

根据本文的贪婪策略及相应的算法设计，其实是一个逐步覆盖的过程参见图 4.1:



**图 4.1 贪婪算法逐步覆盖示意图**

说明：图中 WSP 表示 WS 服务站点(服务器所在位置)，D 表示需求点，每个需求点包含多种 WS 服务需求(这里是指两种 WS 服务)。蓝色的双箭头连线表示该需求点的第一种服务被相应

# 华中科技大学硕士学位论文

服务站点覆盖；红色的双箭头连线表示第二种服务被相应的服务站点覆盖。

本文所用的贪婪算法思想非常简单，就是将  $P$  个服务站点的部署看成  $P$  次每次部署一个服务站点，如图 4.1 所示，每一步都添加一个设施使得新增加的有效覆盖最大。

## (2) 子问题的求解

在 step3 中，计算根据未被覆盖的需求点集  $F'$  及未选服务站点  $j, j \in F'$  的有效覆盖子集  $\omega_j$  计算各个服务站点  $j$  的最大覆盖的子问题。当考虑容量限制的时候，这个子问题其实是一个背包问题(Knapsack problem)：对于服务站点  $j, j \in F'$ ，我们可以将它看成背包，而需求点的需求量  $h_{ik}$  和相应的时间满意度  $f(t_{ijk}) * h_{ik}$  则看成商品的重量和价值。

背包问题(Knapsack problem)是一种组合优化的 NP 完全问题。它是在 1978 年由 Merkel 和 Hellman 提出的，可以表示为：

$$\begin{aligned} \max Z &= \sum_{j=1}^n p_j x_j \\ \text{s.t.} \quad &\begin{cases} \sum_{j=1}^n w_j x_j \leq W \\ x_j \geq 0 \text{ 且为整数 } (j=1,2,\dots,n) \end{cases} \end{aligned}$$

根据是否限定物品选择数量还可以将背包问题分为有界背包问题或者无界背包问题。如果限定物品  $j$  最多只能选择一定数量，则问题称为有界背包问题；反之，如果不限定每种物品的数量，则问题称为无界背包问题。

作为一类经典的 NP-困难问题，已经有了很多相关的研究。在对背包问题的研究中，人们已提出了许多解法，例如：分支界限法(Branch and Bound)、动态规划法(Dynamic Programming)、近似(Approximate Solutions)、遗传算法：(Genetic Algorithm) 等等。其中分支定解法和动态规划法可以求得精确解。由于本文中只要求解规模较小的子问题，所以选用动态规划法进行求解。

# 华中科技大学硕士学位论文

---

王乐等(2006)研究了 0-1 背包问题的动态规划法求解, 并通过在 Matlab 6.5 环境下对动态规划法求解背包问题进行测试并与其他方法进行了对比, 对比结果证明了动态规划法求解背包问题时有着较不错的表现。然而, 该文中的动态规划算法只能求解整数问题, 对于存储空间的浪费也比较大。由于本文中所遇到的问题可能不是整数, 所以本文决定设计自己的算法, 也就是用于求解子问题(背包问题)的动态规划算法:

动态规划是一种思想, 将多维的问题划分为多个一维的问题, 或者将大规模的问题划分为一个个小规模的问题。在使用动态规划求解的时候, 关键的就是对于阶段和状态转移方程的设定。在背包问题中, 我们可以将问题的求解看成决定每一个物品是否要放入的过程。每次决定一个物品  $i$  是否要放入看成一个阶段  $i$ , 只是根据 WS 本身的特点, WS 一个需求点只能被覆盖一次, 相当于物品最多只被放入一次, 这样可以得到状态转移方程:

$$\begin{aligned} f_k(y) &= \max_{x_k \in \{0,1\}} \{c_k x_k + f_{k-1}(y - a_k x_k)\} \\ &= \max \{f_{k-1}(y), c_k + f_{k-1}(y - a_k)\} \quad k \in \{2, 3, \dots, n\} \\ f_1(y) &= c_1 x_1, \quad x_1 \in \{0, 1\} \end{aligned}$$

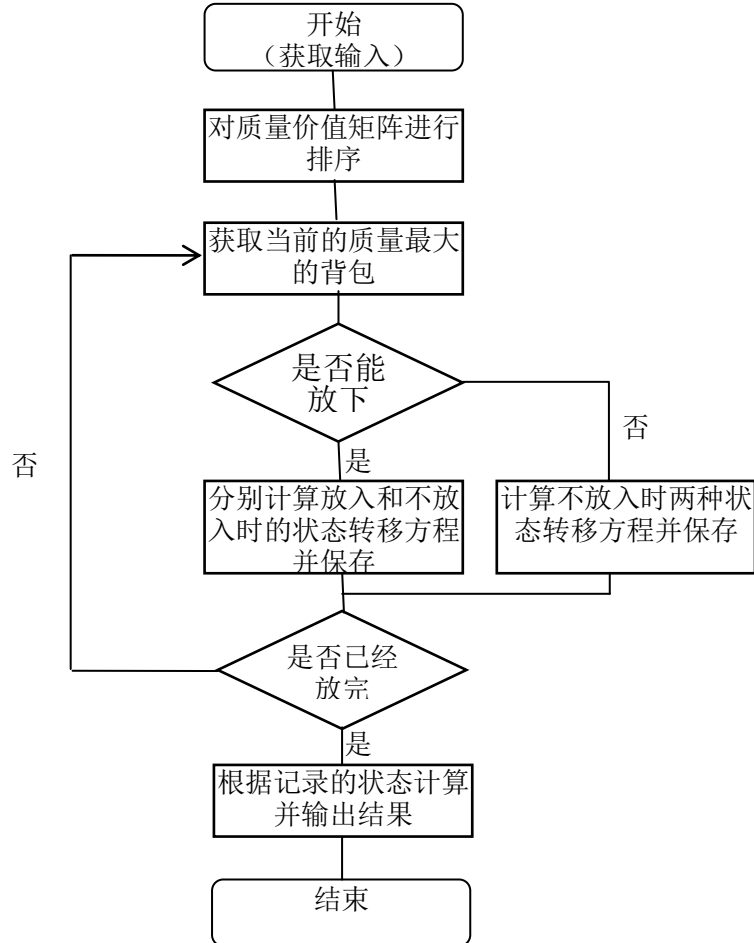


图4.2 动态规划求解背包问题算法流程图

为了表示各个不同的阶段，本文定义元胞数组  $\text{Stage}\{i\}$ , 对应算法中的阶段  $i$ ;

为了表示状态转移方程，本文定义数据结构  $\mathbf{b}_{il} = [w_{il}, v_{il}, \text{status}_{il}, \text{index}_{il}]$ ，其中  $w_{il}$  表示当前背包的剩余容量， $v_{il}$  表示当前背包的价值，0-1 变量  $\text{status}_{il}$  表示该物品是否装入(0 表示不装入，1 表示装入)， $\text{index}_{il}$  表示该状态对应的前一个状态。通过这种设计，就可以完全实现动态算法的求解过程，而且避免了不必要的计算和存储空间的浪费。同时，为了提高算法的效率，本文算法中将先对各种物品(对应 WS 需求点 WSR)按照质量(对应 WSR 的 WS 需求量)进行排序,具体算法流程如下:

## 4.2 算例验证

某国内电子商务巨头，在多处设有服务器，现在整个集团开始推行服务化，不仅仅是对外界(电子商务网站上的所有应用)的应用推行服务化，集团内部的研发和测试平台也要进行服务化改造。也就是说，集团对内对外提供的各种互联网服务都需要以 Web Service 的方式实现。该集团目前提供者上万中服务，其中主要服务有数百种，这些服务现在每天要被调用的平均次数为 1.1 亿次。在这样的数量级下面，如何选择服务器的部署位置，就显得格外重要。为了进行这种选址部署，集团对主要服务进行了响应时间的测试：

表 4.1 某互联网集团 web service 响应时间测试

WS 名称	场景名称及考察项							
	pojo 场景 (ms)		1k string 场 景(ms)		50k string 场景(ms)		200k string 场景(ms)	
	TPS	RT	TPS	RT	TPS	RT	TPS	RT
Dubbo1(hessian2 序列化+mina)	10813.5	0.9	11940	0.8	1962.7	5.1	324.2	30.8
Dubbo2(hessian2 序列化+netty)	11994	0.79	14402	0.64	1293	5.03	362.92	27.49
Dubbo2(dubbo 序列化+netty)	13620	0.67	15096	0.6	1966	7.68	569.5	17.51
Rmi	2461.79	4	11136.02	0.81	3349.88	2.9	1031.28	9.61
Hessian	2417.7	4.1	11426.83	0.79	1925.33	5.13	628.06	15.83
Http(json 序列化)	8179.08	1.15	8919.27	1.04	3247.1	3	1011.97	9.79

注：其中 TPS(Transaction per second)是指平均每秒钟请求次数，RT(Response time)是指平均响应时间。

从表 4.1 可以看出，一个 Web service 的响应时间跟服务类型有很大的关系，根据网络请求所需传递的数据量，本地处理(local processing)时间都有关系。但 Response time(RT)基本上都是从几毫秒到几十毫秒的级别，其中包括本地处理时间和网络传输时间。其中，本地处理时间跟服务本身有关系，而网络传输时间(Network latency)则由 WS 传递的数据量和传递的距离有关系。由于本文的重点在于，根据 WS 的响应时间，为 WS 的服务器选址提供解决方案，而且，在实际应用中，针对特定的 WS，其本地处理时间和网络传输时间是可以通过实验检测出来的，所以本文算例中不再关注究竟是哪些 WS，以及他们的响应时间如何。

# 华中科技大学硕士学位论文

在此基础上，我们可以构造我们的算例，用以验证算法的有效性。本文的算例中，我们先考虑  $I$  个需求点,  $J$  个候选服务站点,  $K$  种 WS,  $P$  个最终部署 WS 的服务站点:

- 其中需求点对各个服务的需求量为 100~400(个请求数)之间的随机分布;
- 备选的服务站点的对不同服务的容量为 1000~3000(个请求)之间的正态分布;
- 不同服务的本地处理时间为 5-10(ms)之间的正态分布;
- 不同需求点到服务站点的网络传输时间为 5~20(ms)的随机分布;
- 时间满意度函数使用凹凸函数，并假设的阈值分别为 WS 本地处理时间的 1.3 到 3.0 倍;
- 同时进行两组实验，一组含容量约束，一组不含，每次每组实验 100~200 次。

为了将问题简化，不同需求点对于不同服务的选择直接参考时间，每个需求点对一种服务都设定一个最大响应时间，该时间服从 10~25(ms)的随机分布。在这个最大响应时间内的 WS 需求点则接受，反正则不接受。通过软件 MATLAB 2009a 进行数值实验，使用电脑 CPU I3 390，内存 2G，效果如表 4.2 所示:

表 4.2 本文算法求解与枚举法求解结果的对比情况

序号	问题规模				含容量约束			
	I	J	K	P	Max	Min	Mean	P(>0.95)
1	5	3	2	2	1	0.9348	0.9970	0.9703
2	6	3	2	2	1	0.9219	0.9927	0.9710
3	7	3	2	2	1	0.9361	0.9933	0.9733
4	6	4	2	2	1	0.9315	0.9932	0.9703
5	7	4	2	2	1	0.9099	0.9892	0.9307
6	8	4	2	2	1	0.9194	0.9903	0.9505
7	9	4	2	2	1	0.9025	0.9918	0.9604
8	10	5	2	2	1	0.9211	0.9900	0.9307
9	10	5	2	3	1	0.9056	0.9832	0.9520
10	10	6	2	3	1	0.9201	0.9829	0.9420
11	11	6	2	3	1	0.9001	0.9809	0.9220
12	12	6	2	3	1	0.9134	0.9849	0.9232
13	13	6	2	3	1	0.9064	0.9864	0.9434
14	14	6	2	3	1	0.8871	0.9895	0.9611
15	15	6	2	3	1	0.8973	0.9895	0.9315
16	15	6	3	3	1	0.9071	0.9806	0.9410
17	15	7	3	3	1	0.8971	0.9895	0.9364

说明：表中 Max, Min, Mean 和 P(>0.95)分别代表与枚举法求解结果的最大相似度，最小相

# 华中科技大学硕士学位论文

似度，平均相似度和相似度大于 0.95 的概率。

从表 4.2 中可以看出本文设计的算法求解效果还是比较理想的：**Max** 的值都为 1，说明算法求解可以经常与枚举法求解相似度为 1，也即得到相同的结果；**Mean** 的值基本都在 0.98 左右，说明算法求解可以保持与枚举法求解保持较高的平均相似度；**P(>0.95)** 的值基本都在 0.9 以上，说明算法求解与枚举法求解相似度在 0.95 以上的概率比较高。

综上所述，本文算法求解效果比较好，但是不可否认的是，由于本文算法在处理子问题的时候，采用了精确算法——动态规划法求解，在追求精确解的同时，牺牲了大量的时间，算法求解的效率随着问题规模的上升而扩大。也即，本文算法的效率受到子问题的规模的影响很大，只能用于求解子问题规模比较小问题。求解精度和时间消耗都跟子问题的规模相关，子问题规模越小，精度越高，时间消耗约少。因而控制子问题的规模非常关键。

本问题中，使用了贪婪算法对问题进行求解，求解精度在子问题规模确定的情况下，还与各个子问题之间的交叉程度有关。交叉程度越低，算法求解精度越高，反之则求解精度表现不好。本文算法中，在规模不变的情况下，每个需求点的最大响应时间越大，有效覆盖半径越大，则问题的交叉越高，子问题规模也响应变大，为此，我们在保持问题规模不变的情况下，通过改变有效覆盖半径来看求解精度的变化情况。实验结果见图 4.3 和图 4.4。

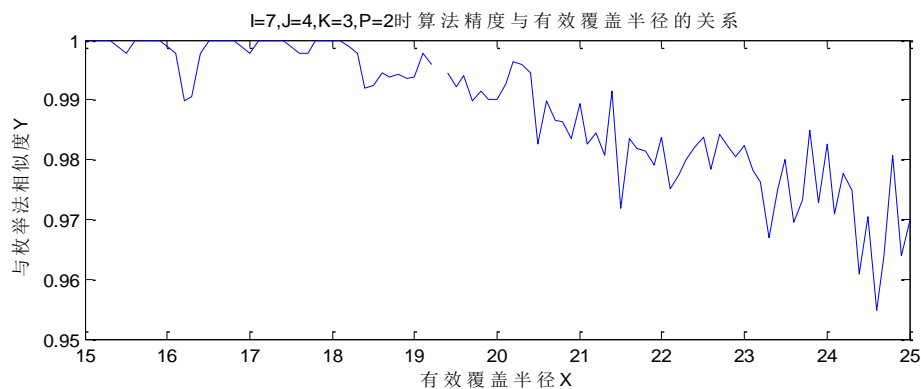


图 4.3  $I=7, J=4, K=3, P=2$  时算法精度与有效半径的关系

# 华中科技大学硕士学位论文

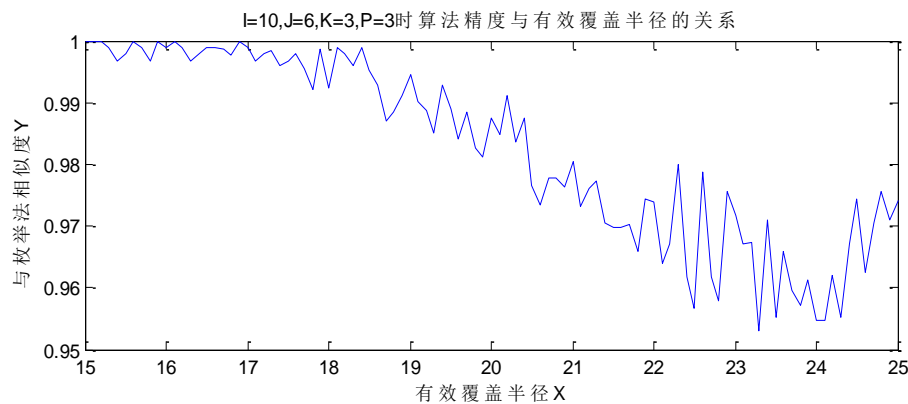


图 4.4  $I=10, J=6, K=3, P=3$  时算法精度与有效半径的关系

从图 4.3 和图 4.4 中我们可以发现，在相同规模下，随着有效覆盖半径的增大，求解精度总体上出现了下滑(图中的锯齿状的图形的出现，是由于每组实验的实验次数不够大所致)，从而佐证了之前的分析。

而之前数值试验中发现的另一个重要问题就是随着子问题的规模的扩大，因为子问题使用精确求解的方式——动态规划法求解的，所以当子问题规模大到一定程度，本文算法将不再有效。所以，本文考虑将子问题的解法也改用已被证明求解背包问题效果比较好的贪婪算法(王乐等，2006)，以求对子问题规模比较大的问题获得较好的求解效果和求解效率。子问题的贪婪策略是：性价比最大和价值最大两种策略综合运用，取最大值。于是通过类似于上述数值实验的比较，获得结果：

表 4.3 子问题用贪婪算法求解与动态规划法求解情况对比

序号	问题规模				子问题用贪婪算法求解					子问题用动态规划法求解				
	I	J	K	P	Max	Min	Mean	P(>0.95)	T(s)	Max	Min	Mean	P(>0.95)	T(s)
1	5	3	2	2	1	0.9137	0.9930	0.9537	0.0011	1	0.9348	0.9970	0.9703	0.0009
2	6	3	2	2	1	0.9339	0.9955	0.9323	0.0012	1	0.9219	0.9927	0.9710	0.0009
3	7	3	2	2	1	0.9170	0.9946	0.9409	0.0012	1	0.9361	0.9933	0.9733	0.0009
6	8	4	2	2	1	0.9004	0.9893	0.9320	0.0016	1	0.9194	0.9903	0.9505	0.0015
7	9	4	2	2	1	0.8522	0.9899	0.9417	0.0017	1	0.9025	0.9918	0.9604	0.0017
8	10	5	2	2	1	0.9037	0.9930	0.9623	0.0019	1	0.9211	0.9900	0.9307	0.0023
11	11	6	2	3	1	0.8528	0.9745	0.8658	0.0030	1	0.9001	0.9809	0.9220	0.0037
12	12	6	2	3	1	0.8607	0.9709	0.8176	0.0030	1	0.9134	0.9849	0.9232	0.0045
13	13	6	2	3	1	0.8322	0.9722	0.7547	0.0032	1	0.9064	0.9864	0.9434	0.0055
14	14	6	2	3	1	0.8845	0.9739	0.7736	0.0031	1	0.8871	0.9895	0.9611	0.0082
16	15	6	3	3	1	0.8437	0.9541	0.6038	0.0045	1	0.9071	0.9806	0.9410	0.0198
17	15	7	3	3	1	0.7973	0.9520	0.6226	0.0052	1	0.8971	0.9895	0.9364	0.0235
18	20	10	5	5	--	--	--	--	0.0161	--	--	--	--	--
19	50	20	10	10	--	--	--	--	0.1098	--	--	--	--	--
20	1000	40	10	20	--	--	--	--	0.3210	--	--	--	--	--



# 华中科技大学硕士学位论文

21	500	2000	20	100	--	--	--	--	9.9402	--	--	--	--	--
22	1000	200	20	100	--	--	--	--	19.3877	--	--	--	--	--
23	2000	500	20	200	--	--	--	--	122.7512	--	--	--	--	--

从表 4.3 可以看出,若使用启发式算法(贪婪算法)求解子问题,可以有效的避免因为子问题规模扩大而出现子问题无法求解的情况,而且其求解精度表现也不错,如表 4.3 所示,其求解的结果与枚举法的求解结果的平均相似度基本都在 95% 以上。但由于子问题采用了启发式算求解,只能得到近似解,这使得整个问题的解出现了波动。求解结果与枚举法的相似度的下限(最差情况)和保持相似度在 95% 以上的概率在规模不断扩大的时候出现了明显的下降。而在求解子问题规模比较小的时候,本文算法不仅求解精度高,求解效率也与子问题也使用了启发式算法的算法不相上下。

## 4.3 结论

本文模型总体上,属于 P 覆盖问题,因而无法求得精确解。为了求解比较好的近似解,本文采用了覆盖问题中广泛采用的贪婪算法进行求解。在贪婪算法的每一步的计算中,抽象出了 0-1 背包问题(带容量容量约束的时候),同时为了获得较高精度的解,使用了动态规划法对子问题也即 0-1 背包问题进行求解。从表 4.2 的数据可以发现,本文算法可以获得质量较高的解。但也发现,由于采用了精确算法求解子问题,因而子问题的规模严重影响算法的效率。当子问题的规模比较小时,本文算法不仅求解速度快,而且求解精度高;但当子问题规模扩大后,虽然求解精度依然比较高,但求解效率下降很多。后来改用启发式算法求解子问题,虽然在求解规模上有了突破:求解了 2000 个需求点,500 个候选点,20 种服务,选择 200 个服务站点的的问题,当问题规模在扩大的时候,MATLAB 的内存已经不足。但是子问题采用启发式算法求解时,求解精度不经波动大,而且随着问题规模的扩大而下降,因而我们认为本文的算法的综合表现仍然是最好的。

考虑到,本文应用背景是 WS 选址问题,而 WS 体系往往是一种星状网络,因而子问题的规模基本还是可以控制的,因而本文的算法还是有效的。

## 5 总结与展望

### 5.1 全文总结

在 Web Service 技术和实际应用逐步成熟,而用户对服务水平要求不断提高的趋势下,本文讨论了将选址问题的理论和研究成果应用到了 WS 选址问题中,并根据 WS 本身的特点对模型进行了改进,提出了基于时间满意度的多服务带容量限制 WS 选址问题模型,并采用了贪婪算法与动态规划法相结合的混合算法进行求解,并取得了比较好的求解效果。主要工作如下:

(1) 本文回顾了选址问题(特别是覆盖选址问题)的主要研究成果,并将之应用到了 WS 选址问题中,填补了国内 WS 选址问题的空白。在模型构建上,本文参考了国外相关 WS 选址问题的研究,并结合 WS 自身应用特点对其模型进行了改进。根据 WS 体系的特点,本文增加服务需求点只能选择一个 WS 提供商和每种服务都存在容量的约束,提出了有效覆盖的概念给予 WS 需求方 WSR 自主选择的权利,另外,本文将时间满意度概念加入到了 WS 选址问题模型中,使得模型具实用性,同时保持了较好的拓展性。从而建立了基于时间满意度的多服务带容量限制 WS 选址问题模型。

(2) 本文对覆盖问题的常用算法进行了介绍,并采用了已被证明了求解效果的贪婪算法进行求解,同时在对子问题的求解过程中,考虑到子问题的规模不大,本文采用了动态规划进行求解,最终提出了贪婪算法与动态规划法相结合的混合算法。通过大量算例的求解,本文的算法获得了比较好的求解效果。

## 5.2 研究展望

在完成本文的过程中，虽然花费了很多时间和精力，也查阅了很多相关资料，但是由于时间和自身学术水平的原因，本文的研究还是存在着一定的局限性，存在需要扩展的和改进的地方。对未来的研究，我觉得可以从以下方面进行改进。

(1) 本文强调了响应时间对于 WS 的服务水平的重要性，因而用响应时间作为服务水平唯一考量，现实应用中可能还有其他因素影响服务水平，将来的研究中可以详细研究下 WS 服务水平的影响要素以及响应时间在其中的作用。

(2) 本文在优化的目标中引入了时间满意度，希望最大化覆盖到的需求点的总体时间满意度。但并没有将时间满意度与收益的确定关系描绘出来，未来可以尝试建立时间满意度与收益的关系，也可以用实验或者调查的方法，选出 WS 时间满意度最好的计算方式。

(3) 本文在模型求解中使用了贪婪算法与动态规划法相结合的混合算法求解，该算法在求解子问题规模不大的时候可以获得很好的求解效果。但当子问题规模变大的时候，该算法因为子问题的求解采用了精确算法求解而效率不高。所以以后可以优化子问题的求解，在争取求解精度的同时保持较好的求解效率。或者用新的启发式算法对本文模型进行求解。

# 华中科技大学硕士学位论文

---

## 致谢

本论文是在导师杨超教授的指导下完成的，论文的完成也标志着我研究生生活即将结束。感谢杨老师从论文的开题到最后的定稿中的每一过程都严格、认真、负责地把关，倾注了大量的时间和精力，让我很受感动。在我攻读硕士期间，杨老师给了我全方位的悉心教导，不论治学还是做人，都给了我很多帮助和提点。在这里我特别向杨老师表示感谢，感谢杨老师近三年来给予我莫大的帮助与关心，我也将把杨老师的教导谨记在心，在以后的人生旅途中，踏踏实实地走好每一步，用杨老师的教诲走出属于自己的光彩人生。

在此，我还要感谢我的家人，是你们一直以来的支持才让我完成了研究生阶段的学习。我迷茫的时候不仅有杨老师指导我，更有你们让我安心。是你们让我有了直面任何困难的勇气和信心，谢谢你们。

同时，还要感谢管理科学与决策研究所的师兄姐妹们，在我有疑问的时候与我一起讨论，有不开心时候陪我打球，让我顺利地完成本论文。特别是黄松师兄，在我的模型上给了关键性的建议，让我少走了不少弯路。

最后还要感谢所有教导过我的老师们，是你们无私的奉献，让我不断成长。还要感谢与这篇文章的完成有着密切联系的科研工作者们，是他们的科研成果为我本文的研究做了铺垫。

感谢各位教授专家在百忙中对本文的评阅。

## 参考文献

- [1] Aboolian R., Sun Y., Koehler G. J. A location-allocation problem for a web services provider in a competitive market. *European Journal of Operational Research*, 2009, 194: 64-77
- [2] Berman O., Drezner Z., Wesolowsky G.O. Locating Service Facilities Whose Reliability is Distance. *Computers and Operations Research*, 2003, 30(11): 1683-1695
- [3] Berman O. The p Maximal Cover: p Partical Center Problem on Networks. *European Journal of Operation Research*, 1994, 72: 432-442
- [4] Beznosov K., Flinn D. J., Kawamoto S., Hartman B. Introduction to Web services and their security. *Information Security Technical Report*, 2005, 10: 2-14
- [5] Cardinale Y., Haddad J. E., Manouvrier M., et al. Web service selection for transactional composition. *Procedia Computer Science*, 2010, 1: 2689-2698
- [6] Charnes A., Cooper M. M. Measuring the efficiency of decision making units. *European Journal of Operational Research*, 1978(2): 429-444
- [7] Church R., ReVelle C. The maximal covering location problem. *Papers of Regional Science Association*, 1974, 32: 101-118
- [8] Crasso M., Zunino A., Campo M. Easy web service discovery: A query-by- example approach. *Science of Computer Programming*, 2008, 71: 144-164
- [9] Daniel A., Menascé Ruan H., Gomaa H. QoS management in service-oriented architectures. *Performance Evaluation*, 2007, Vol. 64, Issues 7-8, 646-663
- [10] Daskin M. S. *Network and Discrete Location: Models, Algorithms, and Applications*. New York: Wiley Inter-science, 1995.
- [11] Fallah H., NaimiSadigh A., & Aslanzadeh,. *Covering problem In Facility location: Concepts, models, algorithms and case studies*. Heidelberg, Germany: Physica Verlag,

2009.

- [12]Farahani R. Z., Asgari N., Heidari N. et al.Covering problems in facility location: A review.Computer & Industrial Engineering(2011),doi:10.1016/j.cie.2011.08.020
- [13]Fensel D., Bussler C. The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, 2002, 1(2):113-137
- [14]Grossman T., Wool A. Computational experience with approximation algorithms for the set covering problem. European Journal of Operational Research, 1997, 101:81-92
- [15]Hamadi R., Benatallah B. A Petri Net-based Model for Web Service Composition. Proceedings of the 14th Australasian database conference. Australia, Adelaide, 2003. 17:191-200
- [16]Jamin S., Jin C., Kurc A.R., et al. Constrained mirror placement on the internet in: Proceedings of IEEE INFOCOM. Anchorage. AK. 2001
- [17]Johansson J. M. On the impact of network latency on distributed systems design. Information Technology Management, 2000, 183–194.
- [18]Kariv O., Hakimi S. An Algorithmic Approach to Network Location Problem. The p-Medians, SIAM Journal on Applied Mathematics, 1979, 37(2):539-560
- [19]Levin Y., Ben-Israel A. A Heuristic Method for Large-Scale Multi-Facility Location Problem. Computers & Operations Research, 2004, 31:257-272
- [20]Leymann F., Roller D., Schmidt M. T. Web services and business process management. IBM SYSTEMS JOURNAL, 2002, 41:198-211
- [21]Maamar Z., Benslimane D., Thiran P., et al. Towards a context-based multi-type policy approach for Web services composition, Data & Knowledge Engineering, 2007, 62:327–351
- [22]Martens A. Analyzing Web Service based Business Processes. Proceedings of 8th International Conference on Fundamental Approaches to Software Engineering,2005, 3442: 19-33

- [23]Papazoglou M.P., Georgakopoulos D. SERVICE-ORIENTED COMPUTING. COMMUNICATIONS OF THE ACM , 2003, 46:25-28
- [24]Pauw W. D., Lei M., Pring E., et al. Web Services Navigator:Visualizing the execution of Web Services. IBM SYSTEMS JOURNAL, 2005, 44(4):821-845
- [25]Qiu L., Padmanabhan V. N., Voelker G. M. On the placement Of Web server replicas. Proceedings of IEEE INFOCOM, Anchorage, 2001, AK
- [26]Rodriguez J. M., Crasso M., Zunino A. et al. Improving Web Service descriptions for effective service discovery. Science of Computer Programming, 2010,75: 1001-1021
- [27]Roth R. Computer Solutions to Minimum Cover Problems. Operation Research, 1969, 17:455-465
- [28]Schilling D. A., Jayaraman, V., Barkhi R. A review of covering problem in facility location. Location Science, 1993, 1(1), 25–55.
- [29]Spekman R, Salmond D, Kamauff J. At last procurement becomes strategic.Long-Range Planning, 1994, 27(2): 76-84
- [30]Sun Y., Koehler G. J., A location model for a web service intermediary. Decision Support Systems, 2006, 42:221-236
- [31]Tang Q.C., Cheng H.K. Optimal location and pricing of Web services intermediary. Decision Support System,2005,40:129-141
- [32]Toregas C., Swaim R., ReVelle C., et al. The location of Emergency Service Facilities. Operation Research, 1971, 19:1763-1373.
- [33]Turner M., Budgen D., Brereton P. Turning Software into a Service. Computer,2003, 36(10): 38-44
- [34]Wang H., Huang J. Z., Qu Y. et al.Web services: problems and future directions Journal of Web Semantics ,2004,309-320
- [35]Zhang Z., Tan Y., Dey D. Price competition with service level guarantee in web services. Decision Support Systems, 2009, 47: 93-104
- [36]后德君, 肖秋霞, 张坤. 基于模糊聚类的 RDC 选址的研究. 微机发展, 2004,

# 华中科技大学硕士学位论文

---

- 14(2):43-45
- [37]何刚,魏连雨.基于人工神经网络第三方物流企业的物流中心选址研究.物流科技,2004,27(103):4-7
- [38]何亚群,胡寿松.基于粗糙集的空军航材供应点的偏好选址.系统工程理论与实践,2003,7:95-99
- [39]贺政纲,帅斌,廖伟.GIS在物流中心选址中的应用.交通标准化,2004,129:17-20
- [40]李延晖,马士华.基于时间约束的单源/P个中转点配送系统的MINLP模型.中国管理科学,2004,12(3):86-90
- [41]罗子明.顾客时间满意度.商业研究,2002,11:27-29
- [42]吕关锋.利用城市GIS中数据信息进行消防站的选址.消防科学与技术,2001:3:28-28
- [43]马云峰.网络选址中基于时间满意的覆盖问题研究:[博士学位论文].保存地点:华中科技大学图书馆,2005.
- [44]马正元,黄斌.Hopfield神经网络在物流配送中心选址优化中的应用.组合机床与自动化加工技术,2003,3:24-26
- [45]王刊良,许寅峰,汪应洛.一个有害物品填埋场选址的决策支持系统.系统工程理论与实践,1997,11:60-65
- [46]王乐,王世卿,张静乐.基于Matlab的0-1背包问题的动态规划方法求解.计算机技术与发展,2006,16:89-92
- [47]谢友才.快速反应中心的选址问题及动态解法.宁波大学学报(理工版),2003,16(3):302-304
- [48]杨波.多品种随机数学模型的物流配送中心选址问题.中国管理科学,2003,11(2):45-49
- [49]杨超,马云峰,杨珺.一类带容量限制的服务站选址问题.系统工程,2004,22(1):19-23
- [50]张敏.易腐物品物流WS设施选址问题研究:[博士学位论文].保存地点:华中



科技大学图书馆，2006.

## 附录 2 攻读学位期间参加的科研课题

- [1] 国家自然科学基金项目“基于需求多元化的网络设施选址研究”
- [2] 华中科技大学研究生创新基金项目“基于客户行为视角的供应链运作优化研究”