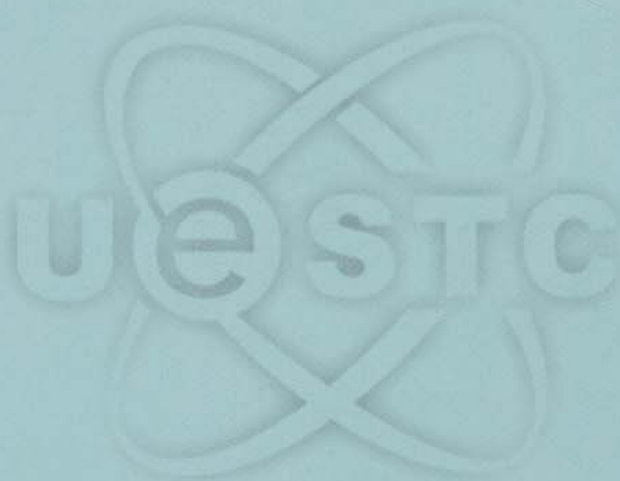


电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 硕士学位论文

MASTER DISSERTATION



论文题目     蚁群算法及其在 QoS 组播路由中的应用研究

学科专业     软 件 工 程

指导教师     邱会中   副教授

作者姓名     王庆美

班 学 号     200620606036

分类号

UDC <sup>注1</sup> \_\_\_\_\_

# 学 位 论 文

蚁群算法及其在 QoS 组播路由中的应用研究

(题名和副题名)

王庆美

(作者姓名)

指导教师姓名 邱会中 副教授

电子科技大学 成 都

(职务、职称、学位、单位名称及地址)

申请专业学位级别 硕士 专业名称 软件工程

论文提交日期 2009.4 论文答辩日期 2009.5

学位授予单位和日期 电子科技大学

答辩委员会主席 \_\_\_\_\_

评阅人 \_\_\_\_\_

2009 年 5 月 25 日

注 1：注明《国际十进分类法 UDC》的类号。

## 独 创 性 声 明

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

签名： 王庆东 日期：2009年6月1日

## 关于论文使用授权的说明

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

签名： 王庆东 导师签名： 邱林  
日期： 2009年6月6号

## 摘 要

随着网络技术的飞速发展,当前通信网络带宽和处理能力的提高使网络能够提供更多的多媒体业务,也使得支持“点到多点”或“多点到多点”的组播通信方式成为网络支持多媒体业务的必要形式。组播路由是网络层具备的功能,组播问题的关键在于组播路由的确定,寻找简单、高效、健壮的组播路由算法一直是网络界致力研究但未完全解决的问题。另一方面,许多分布式的多媒体应用对时延、时延抖动、带宽以及包丢失率有不同的要求,这需要当前网络能够传送具有这些 QoS 要求的实时多媒体信息。因此,作为 QoS 为中心的网络体系结构中不可缺少的组成部分,基于 QoS 约束的组播路由算法的研究成为网络研究领域的重要内容和热点问题。

蚁群算法是一种基于蚂蚁种群的新颖的自激励启发式算法,能够解决一系列复杂的组合优化问题。蚁群算法使用了正反馈机制、分布式计算以及结构化的贪婪启发式算法,具有较强的鲁棒性、寻径过程的并行性以及易于与其他启发式算法结合的优越性,在解决各种具有 NP 难度问题的实践中表现出了良好的效果,在优化组合问题的应用中表现出巨大的潜力。

论文主要研究优化的蚁群算法来解决具有 NP 难度的 QoS 组播路由问题。在蚁群算法中选择合适的参数能够获得良好的性能。论文对蚁群算法参数的选择进行了详尽的研究,包括理论分析、仿真实验、结果分析。提出了蚁群算法的参数最优组合方法,TSP 问题的实验结果证明了该方法的可行性和有效性。针对蚁群算法收敛速度慢和容易局部收敛的问题,对蚁群算法进行了优化,设计了新的蚂蚁转移规则和信息素更新机制。在信息素强度更新的策略中新引入了全局最优系数,通过判断全局最优系数来动态地更新全局信息素挥发系数和状态转移参数。分析了蚁群算法在解决复杂的组合优化问题上的优势,研究了 QoS 网络路由模型。对如何应用蚁群算法解决 QoS 组播路由问题提出了详细的分析和设计,内容包括数学模型的抽取、组播树的构造。最后,对 QoS 组播路由算法进行了探讨,将优化的自适应蚁群算法应用到解决 QoS 组播路由问题中。根据实际的网络模型进行了仿真,并对仿真结果进行了分析。

**关键字:** 组播路由, 服务质量, 蚁群算法

## ABSTRACT

With fast development of network technologies, increase of network bandwidth and processing power makes the network provide more multimedia applications, and also makes the multicast communication that supports “one-to-many”, or “many-to-many”, become an necessary mode of multimedia services fundamental issue in multicast communication is how to determine an efficient multicast routing, and finding simple, effective and robust multicast routing algorithms is unsolved problem in network fields. In addition, many distributed multimedia applications have various demands on delay, delay variation, bandwidth and packet loss, which requires network to transmit real-time multimedia information with these Quality of Service(QoS) constraints. So, as an indispensable component in a QoS-centric network architecture, research on multicast routing algorithms based on QoS constraint becomes an important part and hotspot issue of network research fields.

The characteristics of ant colony algorithm are distributed computing, positive feedback and constructive greedy heuristic. With inherent advantage such as robustness, parallelism of and easiness of combination with other heuristic algorithm, the ant colony algorithm is widely used to solve various different NP-hard problems in the last few years, which has empirically shown its excellent performance and huge potential of application in the fields of combinatorial optimization.

This thesis concentrates on researching multicast routing algorithm based on ant colony algorithm with QoS constraints, applies an improved colony algorithm to solve QoS multicast routing problems with NP difficulty. If the appropriate parameters are selected in the ant colony algorithm, a rather optimistic performance can be obtained. A detailed research of the selection of parameters is carried out in ant colony algorithm, including theoretical analysis, simulation, and analysis of results. The method of the optimal combination of the parameters is brought forward.

The ant colony algorithm is improved on in view of problem that its premature partial restrain and convergence. The new ant motion rules and mechanism of pheromone update are designed in the improved ant colony algorithm. A parameter of

global best tour in strategies of pheromone strength updating is mainly presented.

According to the judgment of a parameter of global best tour to dynamically update pheromone volatile factor and the transfer state factor, the algorithm ensures that global searching abilities are improved adaptively and avoid falling in local peak.

The actual network model is researched in detail based on the analysis of applying the ant colony algorithm to solve combinatorial optimization problem. Improved ant colony algorithm is applied to solve the QoS multicast routing problem. Simulation results demonstrate that this algorithm is effective, feasible and steady for solving QoS multicast routing problem with multiple constraints.

**Keywords:** Multicast routing, Quality of service, Ant colony algorithm

# 目 录

第一章 引 言 .....	1
1、1 课题背景以及选题意义 .....	1
1.1.1 QoS 组播路由概述 .....	1
1.1.2 蚁群算法简介 .....	1
1.1.3 研究 QoS 组播路由的意义 .....	2
1.2 蚁群算法研究现状 .....	3
1.3 论文主要内容和结构安排 .....	4
第二章 蚁群算法的原理和数学模型 .....	6
2.1 蚁群算法的原理 .....	6
2.1.1 蚂蚁的觅食策略 .....	6
2.1.2 蚁群算法的原理 .....	6
2.2 蚁群算法的基本思想 .....	7
2.2.1 真实蚂蚁和人工蚂蚁的区别 .....	8
2.2.2 蚁群算法的基本思想 .....	9
2.3 蚁群算法的算法模型 .....	10
2.3.1 蚁群系统 .....	13
2.3.2 蚁群算法的实现步骤及流程图 .....	14
2.3.3 蚁群算法的复杂度分析 .....	16
2.4 蚁群算法的意义及应用 .....	17
2.4.1 蚁群算法的意义 .....	17
2.4.2 蚁群算法的应用 .....	17
2.4.3 蚁群算法的优缺点分析 .....	18
2.5 本章小结 .....	19
第三章 参数对蚁群算法性能的影响 .....	20
3.1 信息素强度选择的研究 .....	20
3.1.1 信息素强度的选择范围 .....	20
3.1.2 信息素强度初始值的影响 .....	21
3.2 蚁群算法的蚂蚁数目与节点关系的研究 .....	24

3.2.1 蚂蚁数目的影响 .....	24
3.2.2 蚂蚁初始化位置的影响 .....	25
3.3 蚁群算法的启发因子与信息素挥发因子选择的研究 .....	26
3.3.1 启发因子的影响 .....	26
3.3.2 全局信息素挥发因子的影响 .....	29
3.3.3 启发因子和信息素挥发因子组合的影响 .....	31
3.4 蚁群算法的蚁群系统状态转移参数的研究 .....	32
3.5 蚁群算法的参数最优组合原则 .....	34
3.6 本章小结 .....	34
<b>第四章 蚁群算法的自适应优化研究</b> .....	<b>35</b>
4.1 全局最优系数 .....	35
4.2 蚁群算法参数的自适应优化 .....	36
4.2.1 信息素挥发系数优化 .....	36
4.2.2 自适应的蚁群系统状态转移参数优化 .....	37
4.2.3 全局最优系数的影响 .....	37
4.3 基于自适应的蚁群算法描述 .....	38
4.4 在 TSP30、TSP50、TSP75 不同节点规模下验证 .....	39
4.5 本章小结 .....	41
<b>第五章 QoS 组播路由技术研究</b> .....	<b>42</b>
5.1 QoS 组播路由技术特点 .....	42
5.1.1 基本概念 .....	42
5.1.2 组播路由树型结构 .....	43
5.1.3 组播路由算法分析 .....	46
5.1.4 组播路由协议 .....	47
5.2 QoS 路由数学模型研究 .....	49
5.2.1 QoS 路由参数的选择 .....	49
5.2.2 QoS 路由的网络模型 .....	50
5.3 QoS 组播路由的研究 .....	51
5.3.1 QoS 组播路由技术面临的挑战 .....	51
5.3.2 QoS 组播路由技术设计的目标 .....	53
5.4 本章小结 .....	53
<b>第六章 基于自适应蚁群算法的 QoS 组播路由算法及其应用</b> .....	<b>54</b>



6.1 QOS 组播路由数学模型 .....	54
6.1.1 QoS 组播路由问题的描述 .....	54
6.1.2 实现过程 .....	55
6.2 QOS 组播路由算法 .....	56
6.2.1 算法设计的说明 .....	56
6.2.2 算法的描述 .....	56
6.2.3 算法特点分析 .....	57
6.3 仿真实验结果及比较分析 .....	58
6.3.1 仿真环境和参数设置 .....	58
6.3.2 实验结果及分析 .....	59
6.4 本章小结 .....	61
第七章 结束语 .....	62
参考文献 .....	64
攻读硕士学位期间发表的论文 .....	68
致谢 .....	69

# 第一章 引言

## 1.1 课题背景以及研究意义

### 1.1.1 QoS 组播路由概述

近年来,互联网的飞速发展伴随着许多新应用的产生,尤其是高带宽需求的多媒体应用,如视频点播、网络会议、远程教育、网络电话、交互式游戏、分布式多媒体数据库等<sup>[1]</sup>。这些新的应用导致网络带宽的急剧消耗和网络拥挤等问题日见显著,而采用传统的点到点的单播或广播通信方式很难满足当前网络信息传输的要求,不仅浪费大量的网络资源,而且效率也很低。为了缓解网络“瓶颈”问题,业内提出了几种解决方案,其中组播技术以其独特的优越性得到了足够的重视。因为采用组播路由技术能够从根本上减少整个网络带宽的需求。在单播技术下,如果有多个用户要求同一服务提供同一信息时,单播路由的带宽将会随着用户的增加而增加,而在组播情况下,由于是在共同的链路上传送同一信息的拷贝,所需带宽不会随着用户的增加而成倍的增加。因此,组播是最适合 Internet 多媒体网络业务的一种一对多(或多对多)的通信方式<sup>[2]</sup>。

另外伴随着网络的快速发展,当前的诸多网络要求既能实现常规的尽力而为服务,又能根据不同用户的要求传输具有一定服务质量(Quality of Service, QoS)的实时多媒体业务。而传统的尽力传输路由只关心网络的平均性能,各种数据流在网络中平等地分享网络资源且能沿多条路径传输,仅有这种特性的路由技术不能满足具有 QoS 要求的实时多媒体业务传送的需求。因此,基于 QoS 的组播路由算法的研究逐步发展起来。简单地说,基于 QoS 的组播路由是通过发现具有某种相关性能约束的最佳组播树,来更好地利用网络资源以支持应用的 QoS 需求。它是以 QoS 为中心的网络体系结构中不可缺少的组成部分,已成为网络研究领域的重要内容和热点问题。

### 1.1.2 蚁群算法简介

蚁群算法(Ant Colony Algorithm, ACA)<sup>[3]</sup>是 19 世纪初期意大利学者 M. Dorigo, V. Maniezzo 等人通过模拟自然界蚂蚁寻径的行为,提出的一种全新的算法,它被广泛用于解决各种具有 NP 难度的问题。

蚁群算法是一种基于自然界蚂蚁种群觅食行为提出来的新型启发式算法<sup>[4]</sup>。在实验观察中，蚂蚁在寻找食物的途中，会在所经过的路径上留下一种化学物质--信息素。由于蚂蚁在寻找食物的过程中能判断出信息素的存在并具有倾向于选择信息素强度高的路径，那么后来的蚂蚁选择该路径的概率也就越大。同时，路径上的信息素也会随着时间的流逝和环境的变化而不断的挥发，这种机制使得蚂蚁不完全受历史遗留信息的约束，有利于蚂蚁搜索新的路径。蚁群寻找到食物源到蚁巢的最优路径是蚁群相互协作的结果，信息素强度最高的路径即为最优路径。

蚁群算法是一种全新的启发式算法，使用了正反馈机制、分布式计算以及结构化的贪婪启发式算法，具有较强的鲁棒性、寻径过程的并行性以及易于与其他启发算法结合的优越性。蚁群算法已经由仅应用于旅行商问题<sup>[5]</sup> (TSP, Traveling Salesman Problem)到其他多个组合优化领域。

### 1.1.3 研究 QoS 组播路由的意义

现在因特网上的网络层采用了 IP(Internet Protocol)协议，因特网又称为 IP 网络。由于网络规模的不断扩大，网络的信息流量不断增加，这样直接导致网络发生拥塞。IP 网络采用了“尽力而为”的服务方式，所有数据报分组在网络中都被同等的对待，任何拥塞链路都会增加数据报分组传输时间，从而导致性能下降、数据抖动、或者数据报分组丢失，不能保证服务质量。当网络发生拥塞时，直接导致的后果是路由器采取了丢弃数据报的策略，将大量的数据包丢弃，同时向源节点发送反馈信息使其降低数据包的发送速度。采取丢弃策略并通过发送反馈信息来实现拥塞控制的策略减轻了网络的吞吐量，但是同时也减少了数据报的发送量，降低了网络的性能。

随着互联网的高速增长，大大改变了消费者对网络性能、安全性和服务质量的期望。随着对网络多媒体应用要求的不断提高，网络在提供尽力而为服务的同时还需要能够支持不同质量保证的服务。QoS 组播路由算法的研究是提高 QoS 服务研究的重要方面之一<sup>[6]</sup>。QoS 能够对数据包进行合理的排队，对含有内容标识的数据包进行优化，并对其中特定的数据包赋以较高的优先级，从而加速传输的进程。QoS 并没有创造带宽，只是根据应用需求和网络状况来管理带宽，它有一套度量指标，如带宽、延时、延时抖动、吞吐量和丢包率等。

QoS 单播路由算法研究的是如何找到一条能满足约束条件的最佳路径或是具有最大可能满足要求的路径，它的缺点在于忽略了由此带来的网络整体资源的开

销，可能会引起诸如网络资源综合利用率低下、出现网络瓶颈等问题，从而造成网络满足整体请求服务数量的下降，最终导致网络性能整体下降。QoS 组播路由技术的研究目标是寻找到一棵能满足约束条件的最小生成树来有效地利用带宽为多用户提供端到端的服务质量控制或保证。

综上所述，在多约束条件下的 QoS 组播路由算法对网络的研究和应用有多重要的作用。近年来，国内外许多学者采用了蚁群算法<sup>[7]</sup>、遗传算法<sup>[8]</sup>、神经网络算法、模拟退火算法等来对 QoS 网络组播路由进行研究，获得了许多重要的研究成果。

## 1.2 蚁群算法研究现状

1991 年，意大利学者 Doirgo. M 等根据自然界中真实蚁群的觅食行为提出蚁群算法，是最早提出的一种源于大自然的仿生类算法。1996 年，Doirgo.M 等在《IEEE Transactions on Systems, Man, and Cybernetics-Part B》上发表了论文《Ant system: Optimization by a colony cooperating Agents》，系统地阐述了蚁群算法的基本原理和数学模型，并对蚁群算法参数对性能的影响进行了初步探讨。1996 年，Schoonderwoerd 和 Holland 等人将蚁群算法用来解决路由问题，将其应用到英国电信电话网络系统一个模型中。1997 年，Doirgo. M 和 Luca Maria Gambardella 等在《IEEE Transactions on Evolutionary Computation》上发表文章《Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem》，提出蚁群系统（Ant Colony System），用于改善蚂蚁系统的性能。1998 年，Doirgo. M 等人在比利时布鲁塞尔组织召开了一次蚁群算法的国际研讨会。1999 年，Gutjahr W J 撰写学术报告和论文，对蚁群算法的收敛性进行了证明。2000 年，Doirgo. M 和 Bonabeau 等人在国际顶级学术刊物《Nature》上发表关于蚁群算法的研究综述，从而把这一领域的研究推向了国际学术的最前沿，使得蚁群算法逐渐成为一个备受关注的研究热点和前沿性课题。

在动态优化组合应用中，网络路由是一个典型的例子。在有向连接的网络中，国际上 Schoonderwerd 等人最先将蚁群优化算法应用解决路由问题，后来 White 等人则将蚁群优化算法应用于单点对单点和单点对多点的有向连接网络路由，Bonabeau 等人通过引入一个动态规则机制改善蚁群优化算法。在国内，用蚁群算法解决 QoS 路由问题，学者们也展开了大量的研究工作<sup>[7][8][43]</sup>，获得了丰富的研究成果。

### 1.3 论文主要内容和结构安排

蚁群算法是一种随机搜索寻优方法,是生物界群体启发式行为,现在已经逐渐应用到组合优化、人工智能、通讯等多个领域。蚁群算法的正反馈性和协同性使其可用于分布式系统,隐含的并行性更使之具有极强的发展潜力。从仿真结果来看,它比目前流行的遗传算法、模拟退火算法等有更好的适应性。但是蚁群算法是一种新兴的模拟进化算法,还缺乏坚实的数学理论基础,算法的参数选择等还有待进一步研究,而且算法本身的寻优能力也需要进一步提高,包括收敛的速度、收敛结果的最优性等。为了提高算法寻优能力,如何改进算法的一些根本性缺点,仍然很值得研究。

本文主要研究基于改进的蚁群算法来解决具有 NP 难度的 QoS 组播路由问题。主要研究工作和成果如下:

(1) 深入分析了蚁群算法国内外的研究现状和算法的优缺点,研究了蚁群算法的机制原理、特征和数学模型。

(2) 在蚁群算法中选择合适的参数能够获得较好的性能,论文对蚁群算法的参数进行了详细的研究,包括理论分析、仿真试验、结果分析,分析的参数包括信息素强度范围、蚂蚁数目和节点的关系,启发因子和信息启发因子,状态转移的参数范围等,提出了蚁群算法的参数最优组合方法,TSP 问题的仿真试验证明该方法的可行性和有效性。

(3) 针对蚁群算法的收敛速度慢和容易局部收敛的问题,对蚁群算法进行了优化,设计了新的蚂蚁转移规则和信息素更新机制,在信息素强度更新的策略中引入了全局最优系数,通过判断全局最优系数来动态地更新全局信息素挥发系数和状态转移参数。实验结果证明优化后的算法能够确保自适应地提高全局搜索能力,同时也避免陷入局部最优解。在节点规模增长时,优化后的算法仍然能够保持良好的性能。

(4) 基于对蚁群算法能够解决一系列复杂的组合优化问题的分析,研究了 QoS 网络路由模型。对如何应用蚁群算法解决 QoS 组播路由问题提出了详细的分析和设计,内容包括数学模型的抽取和组播树的建立。

(5) 对 QoS 组播路由问题进行探讨,将优化的自适应蚁群算法应用到解决 QoS 组播路由问题中,并对实际的网络模型进行了仿真与实际结果的比较分析。仿真结果证明:该算法在解决多约束 QoS 组播路由问题中是可行的、有效的、稳定的。

论文的结构和安排:

第一章阐述了 QoS 组播路由的意义, 介绍了蚁群算法的起源, 分析了蚁群算法的国内外研究现状。第二章分析了蚁群算法的原理与特点, 介绍了该算法数学模型结构, 并分析了该算法的优缺点, 为后面的改进提供了依据。第三章对蚁群算法参数选择进行了比较分析和研究, 并给出了参数最优选择的原则。第四章对蚁群算法的自适应进行了研究, 对蚁群算法进行了优化, 在信息素强度更新的策略中新引入了全局最优系数, 通过判断全局最优系数来动态地更新全局信息素挥发系数和状态转移参数。第五章介绍 QoS 组播路由的特点, 介绍了常见的组播路由算法与协议, 研究了 QoS 网络路由模型。第六章将基于优化的自适应蚁群算法应用在解决 QoS 组播路由中, 并进行了仿真实验, 与其他算法进行了比较实验结果, 得出分析结论。第七章对全文进行了总结, 并针对研究工作的不足指出了以后的工作方向。

## 第二章 蚁群算法的原理和数学模型

大自然不仅赋予了人类赖以生存的环境，同时又给予人类的智慧以启迪。自然界一直是人类创造力的源泉，人类认识事物的灵感也常源于自然。自然界中的许多自适应优化现象给人们以启示：生物体和自然生态系统可通过自身的演化就使许多高度复杂的优化问题得到完美的解决。蚂蚁群体就具有这样的生态系统，它们个体能力并不发达，但它们却能协同工作依靠群体力量发挥出超出个体的智能。

### 2.1 蚁群算法的原理

#### 2.1.1 蚂蚁的觅食策略

蚂蚁是一种几乎没有视力的昆虫，生物学家经过长期的研究发现：蚂蚁作为一个简单的个体，它的觅食行为是带有随机性的。但一群蚂蚁集体觅食时，它们常常能够快速、准确地找到食物源。原来蚂蚁在搬运食物回巢时，本身会分泌一种化学激素，我们称其为外激素(pheromone)<sup>[12]</sup>。蚂蚁个体之间通过外激素来传递信息，蚂蚁在所通过的路径上会留下这种激素，而蚂蚁本身又会通过这种信息的强弱来选择路径，从而达到互相协作，完成复杂任务。外激素在不断挥发，距离越短的路径上相对走过的蚂蚁数就会越多，该物质的强度就会越大。由于蚂蚁在运动中感知这种物质，并倾向于选择向该物质强度高的方向移动。这样就形成了一种信息正反馈现象，即某条路径上走过的蚂蚁越多，那么后来的蚂蚁选择这条路径的概率就越大。蚁群搜索食物源的过程是一个正反馈的过程，由此可以快速地找到从蚁巢到食物的最佳路径。

#### 2.1.2 蚁群算法的原理

蚂蚁寻找食物的过程是蚂蚁群体一个非常重要而且有趣的行为。在自然界中，蚂蚁的食物源根据研究发现，蚂蚁能够在没有任何提示的情况下找到一条从蚁巢到食物源的最短路径，并且能够随环境的变化而重新搜索新的路径。下面用形象化的图示来解释蚁群发现最短路径的原理和机制。如图 2.1：

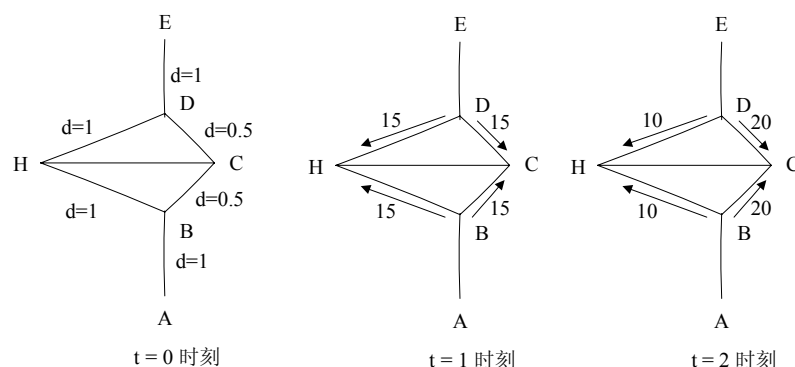


图 2.1 蚁群算法的原理示意图

如图 2.1 中，假设 A 是蚁穴，E 是食物源，HC 为一障碍物，由于障碍物的存在，蚂蚁只能经由 H 或 C 到达 E，或者由 E 到达 A。各点之间的距离如图 2.1 中所示。设每个时间单位有 30 只蚂蚁由 A 到达 B，有 30 只蚂蚁由 E 到达 D，蚂蚁经过后留下的信息素浓度为 1。假设信息素停留时间为 1。

初始时刻  $t=0$  时，由于路径 BH, BC, DH, DC 上均无信息素存在，位于 B 和 D 的蚂蚁可以随机选择路径。从统计的角度可以认为它们以相同的概率选择 BH, BC, DH, DC。

在  $t=1$  时刻，30 只蚂蚁在 B，30 只蚂蚁在 D；

在  $t=2$  时刻，15 只蚂蚁在 B，15 只蚂蚁在 D，30 只蚂蚁在 H；

此时，路径上信息素浓度的分布为  $AB=30$ ， $ED=30$ ， $BH=15$ ， $DH=15$ ， $BC=30$ ， $CD=30$ 。可知 BCD 路径上的信息量是 BHD 路径上的两倍，则将各有 20 只蚂蚁由 B 和 D 到达 C，各有 10 只蚂蚁由 B 和 D 到达 H。这样一来，随着时间的推移，蚂蚁就会以越来越大的概率选择 BCD 路径，直到最终选择 BCD 路径，实现寻优过程。由此可见，蚂蚁之间的信息交换是一个正反馈的过程。单个蚂蚁单个进行寻路，具有分布式并行的特点。

上面的图是一个简单的示意图，蚂蚁寻找食物的过程要比上述过程复杂的多。蚁群算法则是抽象了对蚂蚁寻找食物的过程，它的寻找全局最优解的过程包括两个基本阶段：适应阶段和协作阶段。在适应阶段，人工蚂蚁根据自身的状态信息，结合随机策略，不断调整路径上信息素的强度，同时人工蚂蚁趋向于信息素强的路径；在协作阶段，人工蚂蚁通过信息交流，避免陷入局部最优，向寻找最优解进一步迈进<sup>[10]</sup>。



## 2.2 蚁群算法的基本思想

### 2.2.1 真实蚂蚁和人工蚂蚁的区别

根据蚂蚁的这种正反馈的特性，Dorigo M 等在提出了一种叫做蚂蚁系统（Ant System, AS）的优化方法，它是理解蚁群优化算法的基础。它一方面是真实蚂蚁的行为特征的一种抽象，通过对真实蚂蚁觅食行为的观察，将其最重要的生物学特征赋予人工蚂蚁，另一方面，为了解决现实生活中的优化组合问题，给人工蚂蚁增加了一些真实蚂蚁不具备的特征，目的是为了使蚁群算法能更好的解决实际问题。与现实生活中的蚂蚁相比，人工蚂蚁主要有如下的几个不同点：

- (1) 两者都是一群相互合作的个体。虽然每个蚂蚁都能建立一个解决方案，但是高质量的解决方案是整个蚁群合作的结果。
- (2) 两者有共同的任务。它们的共同任务是寻找起点和终点之间的最短路径（最小代价）。两者都不能跳跃，只能一步一步沿着相邻区域移动。
- (3) 两者都使用信息素进行间接通讯。真实蚂蚁在经过的路径上留下信息素，人工蚂蚁在其所经过的路径上也会记录相应的数据来模仿信息素，蚁群的这种交流方式改变了当前蚂蚁所经过路径周围的环境。
- (4) 两者都采用了信息素挥发机制。人工蚂蚁所经过的路径也会模拟真实蚂蚁经过的路径的特性一样，存在一个信息素挥发机制。路径上的信息素会随时间和环境的变化而挥发，使得两者可以逐渐遗忘历史遗留信息，这样在搜索新的路径时不会受以前经验的限制。
- (5) 人工蚂蚁利用了真实蚂蚁觅食行为中的正反馈机制。当一条路径上的经过的蚂蚁越来越多时，该路径上的信息素就会随之增多，使得该路径上的信息素强度增大。由于蚂蚁具有选择信息素强度大的路径特点，那么后来的蚂蚁选择该路径的概率也会增加。正是这种正反馈机制使得问题的解能够向全局最优的方向不断进化，最终获得相对较优的解。但是，正反馈机制又存在一个缺点，就是容易产生早熟收敛，导致产生局部最优解，阻止了向全局最优的方向做进一步的搜索。
- (6) 两者都使用不可以预测未来的概率状态转移策略。两者都应用了概率决策机制沿着邻近状态移动。从而建立问题的解决方案。它们都只充分利用了局部信息，并没有利用全瞻性预测未来的状态。所应用的策略在时间和空

间是完全局部的。

由于人工蚂蚁是对真实蚂蚁进行的抽象，那么必然会具备一些比真实蚂蚁所不具备的特征。主要表现在以下五个方面：

- (1) 人工蚂蚁存在于一个离散的空间中，它们的状态实质上是由一个离散状态向另一个离散状态的跃迁。
- (2) 人工蚂蚁具有一个内部状态，能够记忆蚂蚁过去的行为。
- (3) 人工蚂蚁并不是随时都释放信息素，而是根据被解决问题的方案来进行的。
- (4) 人工蚂蚁释放信息素的多少是受问题空间特征的影响的，目的是为了更好的解决问题。
- (5) 为了提高算法的效率，需要对算法进行优化。人工蚂蚁被赋予了很多其他的本领，比如预测未来、局部优化、原路返回等，这些行为并不存在于真实的蚂蚁中。

### 2.2.2 蚁群算法的基本思想

在蚁群算法中，一个规模有限的人工蚂蚁群体相互协作来寻求解决优化组合问题较优解。人工蚂蚁根据解决问题的方案，从被选初始状态出发，建立一个可行解，该解可能是较差解。但是人工蚂蚁群体同时建立了很多不同的解，找出较优解则是群体的各个人工蚂蚁相互协作的结构。人工蚂蚁既能独立行动，又能通过信息素来相互交换信息指导同伴行动。它使用了一种结构上贪婪启发搜索办法，根据问题的约束条件找出一个解。由于蚁群算法是人工蚂蚁来模拟真实蚂蚁并增加一些额外有用的特征来实现的，那么需要对自然界中真实蚂蚁群体及其环境进行抽象模拟，并加入一些能提高蚁群算法效率的特征。

- (1) 首先是对真实蚂蚁个体进行抽象得到人工蚂蚁的概念，目的是为了有效地模拟抽取对算法有用的部分真实蚂蚁的生物学特征，同时去掉其他与算法无关的因素。每个蚂蚁具有记忆，用来存储当前路径的信息，这种记忆可以用来构造可行解、评价解的质量、路径反向追踪。然后抽象出问题空间的描述。自然界中真实蚂蚁是生活在一个三维的空间环境中，而对于求解的问题是针对于平面空间。由于真实蚂蚁是在一个连续的二维平面中寻找路径的，那么我们采用计算机来模拟时需要采用一组在平面离散的点来构成。人工蚂蚁在抽象出来的平面的点之间进行跳跃。

- (2) 接着是将路径进行抽象。真实蚂蚁在觅食时对路径的选择是依照环境的信息素强度来决定其前进路线的。最初状态时, 真实蚂蚁是随机寻找路径的, 可以产生无数条杂乱无章的路径。人工蚂蚁则是在平面节点之间进行移动, 节点的规模和人工蚂蚁的数量是可控制的。
- (3) 再则是对信息素的挥发机制的抽象。自然界中蚂蚁在所经过的路径上不断留下信息素, 而同时信息素也随着时间和环境因素的影响则不断挥发。整个过程是一个连续性的过程, 计算机模拟则是通过离散性来实现, 它是通过人工蚂蚁从一个节点到另外一个节点后, 再来对该路径的信息素进行处理。虽然一个是离散状态, 一个是连续状态, 但是对信息素的机制的模拟是一致的而且有效的。
- (4) 最后是引入启发因子, 这个是对真实蚂蚁群体在初始状态的觅食行为存在着的缺陷的一个补充。该缺陷是真实蚂蚁群体在整个觅食过程中寻找到最短路径的演化过程需要很长时间。引入启发因子后, 可以根据求解问题的特征, 给蚁群算法中人工蚂蚁一个初始的引导, 极大的提高了算法的有效性。

从上面的基本思想可以看出, 蚁群算法是一种强大的自学习能力的算法, 能够根据环境的变化和自身过去的行为来对未来的行为提出有效的指导, 提高了算法的有效性。同时也必须指出, 蚁群算法作为一个概率算法, 本质上具有概率搜索的特征, 从数学上对它的可靠性与正确性进行证明还比较困难, 另外蚁群算法模拟的机理的复杂性和环境变化的不确定性也增加了结果的不可预测性。

## 2.3 蚁群算法的算法模型

算法可解问题在实践中不一定是可解的, 因为求解该问题的算法可能需要极长的运行时间与极大的存贮空间, 以至根本不可能在现有计算机上实现。算法对时间和空间的需要量称为算法的时间复杂性和空间复杂性。问题的时间复杂性是指求解该问题的所有算法中时间复杂性最小的算法的时间复杂性<sup>[11]</sup>空间复杂性的定义与时间复杂性的定义类似。按照计算复杂性理论研究问题求解的难易性, 可把问题分为 P 类、NP 类和 NP 完全类。NP 完全问题是 NP 类中最难的问题, Cook 定理奠定了 NP 完全理论的基础<sup>[12]</sup>。

为了说明蚁群算法模型, 首先引入旅行商问题(TSP)。旅行商问题是一个经典的 NP-hard 问题, 蚁群算法最初是在求解旅行商问题(TSP)时而提出的, 它可以

来求解组合优化中的经典 NP-hard 问题。下面以求解  $n$  个城市的 TSP 问题介绍蚁群算法的基本模型。

旅行商问题就是：给定  $n$  个城市和两两城市之间的距离，从某个城市出发，要求确定一条经过各城市当且仅当一次，最后回到出发点的最短路径。其图论描述为：给定图  $G=(V, E)$ ，其中  $V$  为顶点集， $E$  为各顶点相互连接的边的集，各顶点之间的距离是给定的，要求确定一条长度最短的 Hamilton 回路，即遍历所有顶点当且仅当一次的最短回路。

TSP 的已知数据包括一个有限完全图中各条边的权重，其目标是寻找一个具有最小总权重的 Hamilton 圈。对于  $n$  个城市规模的 TSP，则存在  $(n-1)!/2$  条不同的闭合路径。求解该问题最完美的方法应该是全局搜索，但当  $n$  较大时，用全局搜索法精确地求出其最优解几乎是不可能的。蚁群算法来求解 TSP，得到了较好的结果。

首先来假设一些参数： $m$  表示蚁群中蚂蚁的数量； $d_{ij}$  表示两个城市  $i$  和  $j$  之间的距离； $b_i(t)$  表示  $t$  时刻位于城市  $i$  的蚂蚁的个数，其中  $m = \sum_{i=1}^n b_i(t)$ ； $\eta_{ij}$  表示边  $(i, j)$  的能见度，反映由城市  $i$  转移到城市  $j$  的启发程度； $\tau_{ij}$  表示边  $(i, j)$  上的信息素轨迹强度； $\Delta\tau_{ij}$  表示蚂蚁  $k$  在边  $(i, j)$  上留下的单位长度轨迹信息素量； $P_{ij}^k$  表示蚂蚁  $k$  的转移概率， $j$  是尚未访问的城市。

其次，每个蚂蚁作为一个主体具有如下特征：

- (1) 在完成从城市  $i$  到城市  $j$  运动后或是完成一次循环，蚂蚁在边  $(i, j)$  上释放信息素，同时更新路径上的信息素。
- (2) 在本次循环过程中选择下一城市时，不能选择已经经过的城市，这个通过禁忌表  $tabu_k (k=1, 2, \dots, m)$  来控制。
- (3) 根据路径上的信息素强度和启发因子的作用，以相应的概率来选择将要访问的下一个城市。

初始时刻，将  $m$  只蚂蚁按照某种分布的方式放到  $n$  个城市上，各条路径上的信息素量相等，设  $\tau_{ij}(0) = C$ 。（ $C$  为常数）。蚂蚁  $k(k=1, 2, \dots, m)$  在运动的过程中根据各条路径上的信息素和路径的启发信息来决定转移方向。这里使用禁忌表  $tabu_k (k=1, 2, \dots, m)$  来记录蚂蚁  $k$  当前所走过的城市。当该蚂蚁在该次的循环结束后，禁忌表会被清空。蚂蚁系统使用的状态转移规则被称为随机比例规则，它给出了位于城市  $i$  的蚂蚁  $k$  选择移动到城市  $j$  的概率。在  $t$  时刻，蚂蚁  $k$  在城市  $i$  选择城市  $j$  的转移概率  $P_{ij}^k(t)$  为：

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)}, & j \in allowed_k \\ 0, & j \notin allowed_k \end{cases} \quad (2.1)$$

式中  $allowed_k = \{0, 1, 2, \dots, n-1\}$  表示蚂蚁  $k$  下一步允许选择的的城市,  $\alpha$  表示信息启发因子, 表示轨迹的相对重要性, 反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起的作用, 其值越大, 则该蚂蚁越倾向于选择其他蚂蚁经过的路径, 蚂蚁之间的协作性越强;  $\beta$  为期望启发式因子, 表示能见度的相对重要性, 反应了启发信息在蚂蚁选择路径中受重视程度, 其值越大, 则该状态转移概率越接近于贪心规则。 $\tau_{ij}$  表示节点  $i$  和节点  $j$  之间的信息素强度,  $\eta_{ij}$  表示由节点  $i$  转移到节点  $j$  的启发程度, 其表达式:  $\eta_{ij} = 1/d_{ij}$ , 式中  $d_{ij}$  表示节点  $i$  和节点  $j$  之间的距离。

经过  $n$  个时刻, 当所有蚂蚁各自都完成了一次循环后, 才对各路径上的信息素进行更新, 信息素的调整如下:

$$\tau_{ij}(t+1) = \rho * \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (2.2)$$

其中,

$$\Delta\tau_{ij}(t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (2.3)$$

$\Delta\tau_{ij}^k(t, t+1)$  表示第  $k$  只蚂蚁在时刻  $(t, t+1)$  留在路径  $(i, j)$  的信息素量。 $\Delta\tau_{ij}(t, t+1)$  表示此次循环中路径  $(i, j)$  的信息素量的增量,  $(1-\rho)$  为信息素轨迹的衰减系数, 通常设置系数  $\rho < 1$  来避免路径上信息素强度无限的增加。

根据具体算法的不同,  $\Delta\tau_{ij}$ ,  $\Delta\tau_{ij}^k$ ,  $P_{ij}^k(t)$  表达形式可以不同。M.Dorigo 曾给出三种不同的模型<sup>[12~13]</sup>, 分别是蚁量系统(Ant-Quantity System)、蚁密系统(Ant-Density System), 蚁周系统(Ant Cycle System)。

在蚁量系统模型中,

$$\tau_{ij}^k(t+1) = \begin{cases} Q, & \text{若第} k \text{只蚂蚁在本次循环中经过路径}(i, j) \\ 0, & \text{否则} \end{cases} \quad (2.4)$$

在蚁密系统模型中,

$$\tau_{ij}^k(t+1) = \begin{cases} \frac{Q}{d_{ij}}, & \text{若第} k \text{只蚂蚁在本次循环中经过路径}(i, j) \\ 0, & \text{否则} \end{cases} \quad (2.5)$$

以上两个模型中, 前者一只蚂蚁从  $i$  向  $j$  运动的过程中路径  $(i, j)$  上释放的信息

素强度为每单位长度  $Q$ ，与路径  $(i, j)$  的长度无关；而后者一只蚂蚁从  $i$  向  $j$  运动的过程中路径  $(i, j)$  上释放的信息素强度为每单位长度为  $Q/d_{ij}$ ，它是与  $d_{ij}$  成反比的，也就是说在蚁密系统中短路径对蚂蚁有更大的吸引力。

在蚁周系统模型中，

$$\tau_{ij}^k(t+n) = \begin{cases} \frac{Q}{L_k}, & \text{若第 } k \text{ 只蚂蚁在本次循环中经过路径 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (2.6)$$

其中  $L_k$  为第  $k$  只蚂蚁在本次循环中所走的路径长度。

它与上述两个模型存在的区别有两点：

- (1) 蚁量系统模型和蚁密系统模型中每个蚂蚁每走一步都对该路径进行信息素进行更新，而蚁周系统模型是每个蚂蚁完成一个循环才对所经过的路径一起进行信息素更新。
- (2) 蚁量系统模型和蚁密系统模型每次进行信息素更新使用的局部信息，而蚁周系统模型利用的是全局信息。

在求解 TSP 时，蚁周系统模型性能较好，因此通常采用此模型作为蚁群算法的基本模型。

### 2.3.1 蚁群系统

Dorigo 和 Gmabardella 在 1996 年提出的蚁群系统(Ant Colony System, ACS)，有效的改进了蚂蚁系统的性能。蚁群系统对蚂蚁系统主要做了三个方面的改进：

- (1) 状态转移规则为更好更合理地利用新路径和利用关于问题的先验知识提供了方法。

在蚂蚁系统中，蚂蚁完全依赖概率进行路径选择，使用的了随机比例规则，有倾向性地对新路径进行了搜索。而在蚁群系统中，为了避免停滞现象和搜索新的路径，使用了状态转移规则，它采用确定性选择和随机性选择相结合的选择策略，并在搜索过程中动态地调整状态转移概率。状态转移规则，也称为伪随机比例规则，它给出了一只位于节点  $r$  的蚂蚁  $k$  选择的下一个城市  $s$ 。规则如下：

$$s_k = \begin{cases} \arg \max_{u \in allowed_k} \{ [\tau(r, u)]^\alpha [\eta(r, u)]^\beta \}, & q \leq q_0 \\ S, & q > q_0 \end{cases} \quad (2.7)$$

其中， $s_k$  表示蚂蚁  $k$  所选择的下一个节点， $q$  是在  $[0, 1]$  之间的一个随机数， $q_0$  是一个参数 ( $0 \leq q_0 \leq 1$ )，可以调节探索新路径的程度和是否使蚂蚁的搜索活动集中于

最优解的空间邻域内。式(2.7)中的S根据式(2.1)来计算出下一个访问节点。 $\text{allowed}_k$ 表示蚂蚁k下一步允许选择的节点的集合。 $\alpha$ 表示信息启发因子，表示轨迹的相对重要性，反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起的作用，其值越大，则该蚂蚁越倾向于选择其他蚂蚁经过的路径，蚂蚁之间的协作性越强； $\beta$ 为期望启发式因子，表示能见度的相对重要性，反应了启发信息在蚂蚁选择路径中受重视程度，其值越大，则该状态转移概率越接近于贪心规则。 $\tau_{ij}$ 表示节点i和节点j之间的信息素强度， $\eta_{ij}$ 表示由节点i转移到节点j的启发程度。

(2) 使用了全局更新规则，即在每次循环过程结束，只允许全局最优的蚂蚁释放信息素。在蚂蚁系统中，全局更新规则对系统中的所有蚂蚁经过的路径进行信息素更新，这样会降低搜索最优路径的效率，降低了收敛速度。采用了新的全局更新规则后，由于信息素挥发机制的存在和只有最优蚂蚁所经过的路径进行了信息素更新，使得非最优秀路径上的信息素减少。这样使得蚂蚁更倾向于选择最优路径，缩短了收敛于最优解的时间。全局更新规则如下：

$$\tau(r,s) \leftarrow (1-\rho) \times \tau(r,s) + \rho \times \Delta\tau(r,s) \quad (2.8)$$

其中，

$$\Delta\tau(r,s) = \begin{cases} \frac{1}{L_{gb}}, & \text{if } (r,s) \in \text{全局最优路径} \\ 0, & \text{否则} \end{cases} \quad (2.9)$$

$\rho$ 表示全局信息素挥发参数， $L_{gb}$ 为所有蚂蚁完成一次寻径后的全局最优路径。

(3) 使用了局部更新规则，降低蚂蚁经过的路径上的信息素的强度，避免蚂蚁过早的收敛于同一解。在每只蚂蚁每次循环结束后，对所有的路径进行一次全局更新。局部更新规则如下：

对于第k只蚂蚁，如果节点r，S是它所选择路径上的两个相邻节点，则有：

$$\tau(r,s) \leftarrow (1-\alpha) \times \tau(r,s) + \alpha \times \tau_0 \quad (2.10)$$

其中，

$$\tau_0 = (nL_m)^{-1} \quad (2.11)$$

$\alpha$ 表示局部信息素挥发参数， $n$ 表示节点数量， $L_m$ 是由启发式邻近算法得到最短的路径长度。

本文后面章节所使用的蚁群算法均是以该模型为基础的，所做出的改进、优化也是在它的基础之上进行的。

### 2.3.2 蚁群算法的实现步骤及流程图

从蚁群算法模型可知，该算法的寻找全局最优路径是一个反复迭代的过程，以 TSP 为例，蚁群算法的实现具体步骤如下：

- (1) 初始化参数，其中  $C$  表示节点集， $s$  表示  $tabu_k$  的列数，初始化  $s = 1$ ；循环次数  $N_c = 0$ ，设置最大的循环次数  $N$ ，蚂蚁个体  $k$ ；
- (2) 循环次数  $N_c \leftarrow N_c + 1$ ；
- (3) 将源节点置于禁忌表  $tabu_k$  表中的每行中的第一列， $m$  为行数( $k=1,2,\dots,m$ ,  $m$  为蚂蚁总数)，最初  $s=1$ ；
- (4) 蚂蚁个体  $k \leftarrow k + 1$ ；
- (5) 蚂蚁  $k$  根据伪随机比例规则，计算概率选择下一个节点  $j$ ，其中  $j \in allowed_k$ ， $allowed_k$  表示蚂蚁  $k$  下一步可以选择的节点集，即  $j \in \{C - tabu_k\}$ ，就是节点集  $C$  除去禁忌表  $tabu_k$  已经包含的节点。禁忌表  $tabu_k$  是用来保存第  $k$  只蚂蚁的行走路径的表，即禁忌表每行都是保存某个蚂蚁的行走路径；
- (6) 将蚂蚁  $k$  移动到新的节点  $j$ ，应用局部更新规则计算该路径上的信息素强度，然后修改禁忌表  $tabu_k$ ，将节点  $j$  加如禁忌表第  $k$  行第  $s$  列， $s$  的值加 1；
- (7) 如果  $\{C - tabu_k\}$  不为空，即为遍历完整个节点集合  $C$ ，则跳转到第(5)步，否则继续执行下面的步骤；
- (8) 蚂蚁  $k$  完成一次循环结束，应用全局更新规则，更新当前全局最优路径上的信息素。如果  $k \leq m$ ，则跳转到第(4)步，否则继续执行下面的步骤；
- (9) 如果循环次数  $N_c < N$ ，则清空禁忌表  $tabu_k$ ，跳转到第(2)步，否则结束循环输出计算结果。

以 TSP 为例，蚁群算法的程序结构流程图如图 2.2 所示。



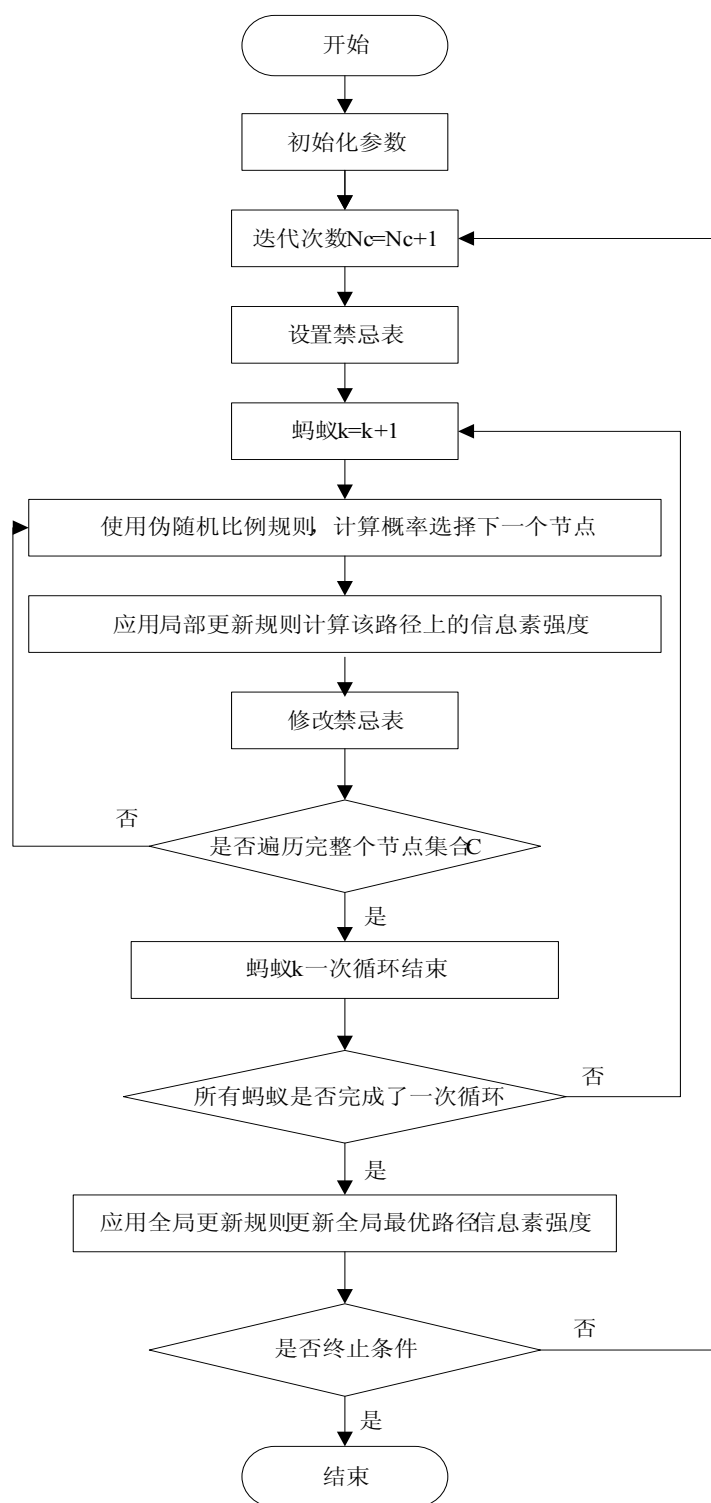


图 2.2 蚁群算法流程图

### 2.3.3 蚁群算法的复杂度分析

蚁群算法的复杂度可表示为问题规模为  $n$  (TSP 中城市的数目) 的函数, 时间复杂度记为  $T(n)$ 。设 TSP 的规模为  $n$ , 蚁群算法所采用的蚂蚁数目为  $m$ , 循环变量为  $N_c$ , 最大循环次数为  $N_{cMax}$ , 下面来分析蚁群算法的时间复杂度。

表 2.1 蚁群算法复杂度分析表

步骤	内容	主要组成部分	时间复杂度
(1)	初始化参数	路径上信息素的初始化, 蚂蚁随机放在节点上	$O(n^2 + m)$
(2)	初始化禁忌表	将源节点置于禁忌表	$O(m)$
(3)	所有蚂蚁完成 $N$ 次循环后	这个部分包含三层嵌套的循环过程, 第一层循环是迭代次数 $N_c$ , 第二层循环是蚂蚁数目 $m$ , 第三层循环是遍历 $n$ 个节点, 在第三层内部还包括三步, 第一步是选择下一节点时计算转移概率并与禁忌表内的节点比较, 第二步是更改禁忌表, 第三步是采用局部更新规则, 第一步是 $O(n)$ , 第二、三步均为 $O(1)$ , 第三层结束返回第二层时, 需要计算全局最优路径使用全局更新规则, 复杂度为 $O(1)$ 。所以整个过程是 $N_c * (m + 2) * (n * (n + 2))$	$O(N_c * m * n^2)$
(4)	输出结果	输出全局最优路径	$O(1)$

通过上表可知, 此计算过程的时间复杂度为  $T_n = O(N_c * m * n^2)$ 。

## 2.4 蚁群算法的意义及应用

### 2.4.1 蚁群算法的意义

仿生优化算法是人工智能研究领域中的一个重要的分支, 其中就包括模拟生物解选择和遗传机制的遗传算法、模拟蚂蚁种群觅食行为的蚁群算法、模拟基于物理中固体物质的退火过程的模拟退火算法等。目前, 蚁群算法已经成为人工智能领域中倍受关注的研究热点, 近年来国内外许多学者获得了许多重要的研究成果。

### 2.4.2 蚁群算法的应用

蚁群算法从 1991 年提出至今, 不到二十年的时间, 但它已经被应用于研究和解决多个领域各类优化组合问题。组合优化问题包括了两类: 一类是静态的组合优化问题, 如旅行商问题(TSP), 二次分配问题(Quadratic Assignment Problem ,

QAP)<sup>[14]</sup>, 作业调度问题 (JSP, Job-shop Scheduling Problem)<sup>[15]</sup>, 车辆路径问题 (VRP, Vehicle Routing Problem)<sup>[16]</sup>, 有序排列问题 (SOP, Sequential Ordering Problem)<sup>[17]</sup>, 图着色问题 (CP, Graph-coloring Problem)<sup>[18]</sup>, 频率分配问题 (FAP, Frequency Assignment Problem)<sup>[19]</sup>以及数据挖掘(DM, Data Mining)<sup>[20]</sup>等; 另一类是动态的组合优化问题, 蚁群算法在这方面的应用主要是通信网络的路由问题。

### 2.4.3 蚁群算法的优缺点分析

蚁群算法本质上是进化算法中的一种新型启发式优化算法, 虽然是从求解旅行商问题 (TSP) 而提出的, 但是它在求解多种组合优化问题中获得了广泛的应用。蚁群算法作为一种新型的进化算法, 其正反馈机制有利于提高发现较优解的能力, 分布式计算能够避免早熟收敛和陷入局部最优解, 贪婪启发式算法有助于在搜索过程的早期阶段找到满足条件的解, 减少了搜索时间。

作为一种随机搜索算法, 它的优点是:

- (1) 蚁群算法是本质一种并行分布式算法。搜索的过程中各个蚂蚁的路径是各自的, 这样虽然容易陷入局部最优, 但通过蚂蚁个体之间不断的信息素的交流和传递有利于发现较好解。在实际的应用过程, 蚁群算法可以分并行运算和串行运行之分。并行运行就是蚂蚁在各个节点之间同时进行, 各个节点需要保存蚂蚁传递的信息, 它可以不需要知道整个节点的拓扑图, 但是蚂蚁在移动过程中会同时更新信息素, 细致的规则需要制定。串行运行需要知道这个节点的拓扑图, 各条路径上的信息素更新易控制。
- (2) 蚁群算法具有正反馈特性, 蚂蚁在搜索的过程中会在路径上留下信息素, 对后来的蚂蚁具有指导作用。
- (3) 蚁群算法的具有较强的鲁棒性。
- (4) 易于与其它算法结合。

但是该算法缺乏坚实的数学理论基础, 也存在一些缺点和不成熟的方面:

- (1) 算法初期求解慢。
- (2) 解决大规模问题时性能下降, 需要更多时间和更容易陷入局部最优解。
- (3) 作为一种贪婪启发式算法, 它也存在着搜索时间长的缺点。
- (4) 寻找最优解易于过早的陷入局部最优, 出现停滞现象, 即搜索进行到一定程度后, 所有个体所发现的解完全一致, 不利于对解空间进行进一步搜索, 以发现更好的解。

在搜索过程中，尽力加快收敛速度又尽量避免陷入局部最优解，是值得研究的问题。针对这些问题，可以通过对算法规则进行一些调整，从参数的选择、路径选择的概率、信息素挥发等方面对蚁群算法进行优化来改善算法的性能。

## 2.5 本章小结

本章对蚁群算法原理、模型以及应用进行了介绍。首先介绍了的生物学特征，并阐述了蚁群算法的基本思想和原理，然后介绍了蚁群算法的三类基本的数学模型，并介绍了基本蚁群算法的改进，给出了算法的实现步骤和其流程图。最后谈到了蚁群算法的意义及其应用，并分析了其优缺点，为下面两章对蚁群算法的研究与优化提供切入点。

### 第三章 参数对蚁群算法性能的影响

在蚁群算法中，参数是影响其求解性能和效率的关键因素，信息素启发因子  $\alpha$ ，期望启发式因子  $\beta$ ，蚂蚁数目  $m$ ，信息挥发因素  $\rho$ ，信息素初始强度  $Q$  都是非常重要的参数，其选取方法和选取原则直接影响到蚁群算法的全局收敛性和求解效率。但是由于蚁群算法参数之间的关联性，如何确定组合优化组合使蚁群算法<sup>[21~23]</sup>求解性能最佳一直是一个极其复杂的优化问题，大多情况下都是根据经验而定。本章暂时采用一些文献传统的方法得出参数的一般合理范围，然后在此基础上进一步优化。通过仿真，对蚁群算法的参数选择进行了深入研究，选取 Oliver30 城市 TSP 问题，提出了参数组合原则。

#### 3.1 信息素强度选择的研究

##### 3.1.1 信息素强度

在蚁群算法中，初始时刻各条路径上的信息素强度是相等的，蚂蚁会根据各条路径上的信息素来选择下一个节点。那么，初始时刻信息素强度的是值得探讨的问题之一。蚂蚁在寻找路径的过程中，各条路径上的信息素强度也会发生变化。如果在更新信息素强度后，某条边上的信息素强度过高或是过低都会影响蚁群算法的性能，有必要对其信息素强度给出一个范围限定，或者从逻辑上证明算法可以保证各路径上的信息素强度会限制在一个范围<sup>[23~24]</sup>。

蚂蚁在寻找路径时，如果一条路径上的信息素强度远高于其他边，则其他蚂蚁更倾向于这条路径，那么正反馈机制使得该路径上信息素强度更强，导致搜索停滞的现象，陷入局部最优的状态；反之如果一条路径上的信息素强度远低于其他边，其他蚂蚁倾向于其他路径，也会造成陷入局部最优这种现象的发生。蚁群算法在全局更新规则和局部更新规则中能够在逻辑上保证任何路径上的信息素强度在限制在一个范围内。

在路径的构造过程中蚂蚁每走一步，需要进行局部更新，使用局部更新规则公式(2.10)

$$\tau(r,s) \leftarrow (1-\alpha) \times \tau(r,s) + \alpha \times \tau_0$$

其中,  $\tau_0 = (nL_{mn})^{-1}$ ,  $L_{mn}$  是随机选择一个节点, 由最近邻距离方法获得的路径长度。使用该规则使得蚂蚁走过的路径上信息素强度降低, 换句话说就是进行局部信息素挥发, 目的是为了后来的蚂蚁搜索到同一路径上来形成局部最优解。处于节点  $r$  到节点  $s$  之间的路径  $\tau(r,s)$  在  $t$  时刻和  $t+1$  时刻的信息素强度分别为  $\tau_{rs}(t)$  和  $\tau_{rs}(t+1)$ 。根据局部更新规则的要求, 路径上信息素强度降低, 则必须有  $\tau_{rs}(t+1) \leq \tau_{rs}(t)$ , 那么局部更新规则中有  $\tau_0 \leq \tau(r,s)$ 。

在蚂蚁循环一次结束后, 对全局最优路径的信息素进行更新, 使用全局规则公式(2.8)

$$\tau(r,s) \leftarrow (1-\rho) \times \tau(r,s) + \rho \times \Delta\tau(r,s)$$

其中  $\Delta\tau(r,s) = (L_{gb})^{-1}$ ,  $L_{gb}$  为全局最优路径。全局更新规则的目的是加强全局最优路径上的信息素强度, 即  $\tau_{rs}(t+1) > \tau_{rs}(t)$  则有  $\tau(r,s) \leq \Delta\tau(r,s)$ 。

通过局部更新规则和全局更新规则的分析可知, 处于节点  $r$  到节点  $s$  之间的路径  $\tau(r,s)$  在任意时刻, 其范围在  $\tau_0 \leq \tau(r,s) \leq \Delta\tau(r,s)$ , 也就是  $(nL_{mn})^{-1} \leq \tau(r,s) \leq (L_{gb})^{-1}$ 。这与另外一种蚁群优化算法之一的最大最小蚂蚁系统<sup>[25]</sup>有相似之处。将所有路径上的信息素强度限制在一个范围内, 防止信息素强度过高而导致局部最优, 也防止信息素强度过低而降低收敛速度。

### 3.1.2 信息素强度初始值的影响

根据上一小节的推理可知, 蚁群算法能够保证所有路径上的信息素强度限制在一个范围内。在初始时刻, 需要对各条路径上的信息素强度进行初始化。假设信息素强度参数名称为  $Q$ , 任意路径上在  $t=0$  时刻,  $\tau_{rs}(0) = Q$ , 其中  $Q$  为常数。关于  $Q$  的取值问题, 是个值得研究的问题。

采用 Oliver 30 城市 TSP 问题作为仿真对象。通过上一节的分析可知, 处于节点  $r$  到节点  $s$  之间的路径  $\tau(r,s)$  在任意时刻, 其范围在  $\tau_0 \leq \tau(r,s) \leq \Delta\tau(r,s)$ 。先分析最小值的定位, 由于  $\tau_0$  通过给出的问题即可计算得出,  $\tau_0 = (nL_{mn})^{-1}$ ,  $L_{mn}$  是随机选择一个节点, 由最近邻距离方法获得的路径长度, 那么  $(nL_{mn})^{-1} \leq Q$ ; 再分析最大值的定位, 由于  $L_{gb}$  在信息素强度初始化过程中是未知的, 所以可以利用  $L_{mn} \geq L_{gb}$  这个关系来计算  $Q$  的范围。根据  $\Delta\tau(r,s) = (L_{gb})^{-1}$  和  $(L_{mn})^{-1} \leq (L_{gb})^{-1}$  可知,  $Q \leq (L_{mn})^{-1}$  是合理的。那么  $Q$  的范围:  $(nL_{mn})^{-1} \leq Q \leq (L_{mn})^{-1}$ 。

实验的硬件环境: CPU 为 AMD 3800+, 内存 1G。实验的软件环境: 操作系统为 Windows XP, 开发工具为 VC++, 开发语言 C++。后面的实验环境与此相同。

实验可设置的参数如下：城市节点数目  $n = 30$ ，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 3$ ，全局信息素挥发参数  $\rho = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目  $m = 30$ ，状态转移规则中的  $q_0 = 0.1$ ， $L_{mn}$  的计算是随机选择一个节点，从它出发由最近邻距离方法获得。

在寻找全局最优路径的迭代过程中，关于终止条件的设置目前有两种方案，第一种是设置最大循环次数  $NcMax$ ，第二种就是在连续  $N$  次全局最优解没有发生变化则终止迭代，其中  $N$  是参数。该文从本节到以后的实验中，均选择第一种设置方式。理由是在最大循环次数  $NcMax$  内，连续  $N$  次没有变化，不认为是找到了全局最优，而是假定它陷入了局部最优，可以进一步采用自适应改变参数的方法(见下一章)来解决找到最优解。假定的理由是，蚂蚁群算法的目标就是不断逼近最优解，通常无法找出最优解都是因为陷入局部最优而造成的，而且全局最优解就在局部最优解的附近。所以假定它陷入了局部最优是为了继续增大找到最优解的概率。

该实验的终止条件为循环次数  $NcMax$  为 500 次。 $Q$  的范围内选择均匀 10 个点做初始值，每组做 10 次实验，数据结果如表 3.1。

实验 10 次的理由是蚁群算法作为一类随机搜索算法本质上具有概率搜索的特征。在参数相同的情况下每次实验获得全局最优路径长度可能不一样，为了相互之间的比较，所以需要实验多次取平均值，尽量减少随机的因素。

对表格中的数据名词解释如下：

- (1) 平均全局最优路径，指的是在多次实验中，每次实验的全局最优路径的和除以实验次数得到的。
- (2) 最优的全局最优路径，指的是在多次实验中，全局最优路径最短的那次实验结果。
- (3) 最差的全局最优路径，指的是在多次实验中，全局最优路径最长的那次实验结果。
- (4) 全局最优路径的差值，指的是在多次实验中，最优的全局最优路径与最差的全局最优路径之间的差值。
- (5) 平均收敛次数，指的是在多次实验中，获得最优的全局最优路径时平均收敛次数。

全局最优路径的差值是用来说明算法的稳定性，性能良好的算法的计算结果在多次实验中差距过大。平均收敛次数则说明算法的收敛程度，它说明了该算法在寻找全局最优路径时所表现出来的性能。

以上的数据名词解释对后续章节的表格的名称也适用。

下面是根据上面设置的参数，进行的信息素强度的初始值实验。

表 3.1 信息素强度初始值实验

编号	Q 取值	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	$(3L_{nn})^{-1}$	428.125	423.741	438.382	14.641	238
2	$(6L_{nn})^{-1}$	427.768	423.741	436.149	12.408	180
3	$(9L_{nn})^{-1}$	427.299	423.741	438.051	14.310	154
4	$(12L_{nn})^{-1}$	426.823	423.741	437.696	13.955	148
5	$(15L_{nn})^{-1}$	426.964	423.741	439.468	15.727	150
6	$(18L_{nn})^{-1}$	425.919	423.741	434.104	10.363	144
7	$(21L_{nn})^{-1}$	425.629	423.741	429.589	5.848	119
8	$(24L_{nn})^{-1}$	425.489	423.741	429.99	6.249	118
9	$(27L_{nn})^{-1}$	425.31	423.741	429.567	5.826	106
10	$(30L_{nn})^{-1}$	425.228	423.741	428.464	4.723	96

从表 3.1 分析得出以下几点：

- (1) 从上表分析可知，平均收敛次数均小于循环次数  $NcMax$ ，且从全局最优路径来看最优解为实际的最优解，说明计算的全局最优路径结果已经收敛。
- (2) 从上表分析可知，从全局最优路径上来看，多次实验中，至少有一次都能找到最优解，由于最优结果为实际的最优解，说明各类参数设置是合理的。 $n$  值越大，全局最优路径的差值越小，平均收敛次数越少，在取  $n = 27$  和  $n = 30$  时，平均值较好则证明算法很稳定，且平均收敛次数很少。
- (3) 综合结果可得出各条路径初始值  $Q \in [(nL_{nn})^{-1}, (0.9nL_{nn})^{-1}]$ ，即  $Q \in [\tau_0, 1.1\tau_0]$ ，算法具有较好的性能。这与文献[1]的结论  $Q = \tau_0$  是基本是一致的，这也为后面的其他参数的研究和自适应优化提供了正确的方向。

各条路径上信息素强度值  $Q \in [(nL_{nn})^{-1}, (0.9nL_{nn})^{-1}]$ ，即  $Q \in [\tau_0, 1.1\tau_0]$ ，这样当蚂蚁经过一条在历次迭代中非全局最优结果中的路径时，换句话说，该路径上的信息素强度并未在全局更新规则中得到更新，那么该路径的信息素强度是不会变化的。这样局部更新规则在此时经过该路径并不起作用。只有当全局规则更新过的



路径，才能在后来的局部更新中变化。

本小节首先从全局更新规则和局部更新规则中分析了各条路径上信息素强度在更新过程中会限定在一定范围内，然后从理论关系上得出了  $Q$  的取值范围，并在实验中进一步得出具体取值。

## 3.2 蚂蚁数目与节点的关系的分析

### 3.2.1 蚂蚁数目

在蚁群算法中参数初始化的过程中，需要对蚂蚁数目和蚂蚁所处节点的位置进行确定。蚁群算法作为一种随机搜索算法，是通过多个候选解组成的群体的进化过程中来搜索最优解。蚂蚁群体在搜索的过程中，蚂蚁个体之间通过信息素的交流，可以相互协作，表现出复杂而有序的行为来完成任务。蚂蚁的数目以及蚂蚁个体在初始化时所处的位置，这两者对算法的性能都起着至关重要的作用。

对于蚁群算法中，一只蚂蚁遍历完所有节点一次，获得一个解； $m$  只蚂蚁各遍历完所有节点一次结束后，可以获得本次最优解。在理论上说，蚂蚁数目过少或是过多都会影响蚁群算法的性能，蚂蚁数目过多或是过少是相对节点的数目来衡量的。蚂蚁数目过少，会使得某些路径从未被搜索到，全局搜索的能力降低，虽然提高收敛的速度，却陷入了局部最优解的困境，导致算法的稳定性较差；蚂蚁数目过多，可扩大全局搜索的范围，但是降低了收敛速度，而且状态转移规则中包含使用了先验知识，即完全倾向于信息素强度大的规则，同样会导致产生局部最优解的尴尬局面。

蚂蚁数目与收敛速度的关系可以通过对蚁群算法的时间复杂度分析可知： $T(n) = O(N_c * m * n^2)$ ，换句话说，如果蚂蚁数目  $m$  过多，与节点  $n$  成这种线性关系  $m = k * n (k \geq 1)$ ，则蚁群算法的时间复杂度可变化为  $O(N_c * n^3)$ ，所以蚂蚁数目与城市节点数目的关系是一个值得研究的问题。

实验采用 Oliver 30 城市 TSP 问题作为仿真对象，城市节点数目  $n = 30$ ，信息素强度值为： $Q = (nL_{mm})^{-1}$ ，其中  $L_{mm}$  是随机选择的节点出发采用邻距离方法获得，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 3$ ，全局信息素挥发参数  $\rho = 0.1$ ，信息素挥发参数  $\alpha_1 = 0.1$ ，状态转移规则中的  $q_0 = 0.1$ 。

该实验的终止条件为循环次数  $NcMax$  为 500 次，蚂蚁数目  $m$  从 5 开始，每次增加 5 个，直至 50，共 10 个点。

上述操作，每组做 10 次实验，数据结果如表 3.2:

表 3.2 蚂蚁的数目初始化实验

编号	蚂蚁数目	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	5	449.706	432.93	464.34	31.41	111
2	10	431.325	425.104	438.382	13.278	109
3	15	427.951	423.741	441.065	17.324	98
4	20	426.425	423.741	438.382	14.641	91
5	25	425.49	423.741	438.382	14.641	84
6	30	424.003	423.741	424.692	0.951	75
7	35	424.125	423.741	425.267	1.526	88
8	40	424.295	423.741	425.267	1.526	93
9	45	425.759	423.741	438.382	14.641	99
10	50	426.805	423.741	443.207	19.466	106

从表 3.2 分析得出以下几点:

- (1) 蚂蚁数目和节点数目之比过于小的时候，比如小于 0.5 时，平均全局最优路径明显变差；最优全局最优路径与实际的最优解也存在较大差距；全局最优路径的差值过大，说明此时算法的稳定性也比较差。
- (2) 蚂蚁数目和节点数目之比在 1:1 左右的时候，算法的平均全局最优路径与实际的最优解较为接近；多次实验内最优的全局最优路径为实际的最优解说明算法是有效的；全局最优路径的差值明显减小，说明算法是稳定的；平均收敛次数也是较小的，说明此时算法性能较好。

蚂蚁数目和节点数目之比超过 1:1 之后，全局最优路径的平均值、最优值、最差值稍微变差了一点，说明蚂蚁数目过多，虽然能加强全局的搜索范围，但是在使用全局信息素规则和局部信息素更新规则上对信息素影响较大，对算法的性能造成一定影响；但是它比蚂蚁数目明显过少时效果要好，这就是因为蚂蚁越多全局搜索能力越强。

### 3.2.2 蚂蚁初始化位置的影响

在初始时刻，蚂蚁的初始化放在城市的节点上有多种方式，蚂蚁在节点位置

上对路径信息素的更新过程会产生影响，从而使蚁群算法的性能受到影响。

蚂蚁的初始化位置方式目前常见的有三种：

- (1) 所有蚂蚁都从某一个节点出发；
- (2) 所有蚂蚁均匀分布在各个节点上；
- (3) 所有蚂蚁随机分配在节点上；

本节针对这三种方式分别做出仿真实验，实验采用 Oliver 30 城市 TSP 问题作为仿真对象，城市节点数目  $n=30$ ，信息素强度值采用结论： $Q=(nL_{mn})^{-1}$ ，其中  $L_{mn}$  是从随机选择的节点出发采用邻距离方法获得，信息启发因子  $\alpha=2$ ，期望启发式因子  $\beta=3$ ，全局信息素挥发参数  $\rho=0.1$ ，局部信息素挥发参数  $\alpha_1=0.1$ ，蚂蚁数目为采用 3.2.1 的结论  $m \approx n$ ，即  $m=30$ ，状态转移规则中的  $q_0=0.1$ 。

该实验的终止条件为循环次数为 500 次。

上述操作，三种方法每组均做 10 次实验，三种蚂蚁的初始化位置与全局最优路径长度  $L_{gb}$  的数据关系如下：

表 3.3 蚂蚁初始化位置实验

编号	$L_{gb}$ 平均值	$L_{gb}$ 最小值	$L_{gb}$ 最大值	差值	平均收敛次数
(1)	430.493	423.949	443.207	19.125	98
(2)	424.296	423.741	425.51	1.769	82
(3)	426.985	423.949	433.322	9.373	89

由上表分析可得出结论，蚂蚁的位置初始化采用均匀分布效果最佳，这是因为采用均匀分布的方式蚂蚁在寻找路径过程，更能扩大全局的搜索范围，采用随机分布，可能会导致某些蚂蚁会处于相同节点，在搜索过程中更容易陷入局部最优；采用指定点出发也是上述原因。

### 3.3 蚁群算法的启发因子与信息素挥发因子选择的研究

#### 3.3.1 启发因子的影响

信息素是表示过去的信息的载体，启发函数则表示未来信息的载体，它们直接影响到蚁群算法的寻找全局最优路径的值和收敛性。 $\alpha$  表示信息启发因子，表示轨迹的相对重要性，反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起

的作用，其值越大，则该蚂蚁越倾向于选择其他蚂蚁经过的路径，蚂蚁之间的协作性越强，这样就会减少搜索的随机性。 $\beta$ 为期望启发式因子，表示能见度的相对重要性，反应了启发信息在蚂蚁选择路径中受重视程度，其值越大，加强了搜索的随机性，该状态转移概率越接近于贪心规则。

根据状态转移规则可知，信息启发因子 $\alpha$ 和期望启发式因子 $\beta$ 是在同时在该规则里共同来影响蚂蚁对路径的选择。如果它们的选择不恰当，对算法的性能上会有较大影响。在实际的应用中，信息启发因子 $\alpha$ 和期望启发式因子 $\beta$ 之间的联系是非常紧密的。为了寻找到最优路径，蚁群必须将随机搜索的特点和快速收敛的特点结合起来寻找一个平衡点。

在启发因子实验中，首先分别对这两个启发因子，这样可以确定各自参数在另外一个参数不变的条件的变化规律，得出相应的范围，总结其变化规律，目的是为后面的参数组合实验提供了参数范围。

实验采用 Oliver 30 城市 TSP 问题作为仿真对象，城市节点数目  $n = 30$ ，蚂蚁数目采用 3.2.1 节的结论  $m = 30$ ，初始化方式采用均匀分布，信息素强度值 3.1.2 的结论： $Q = (nL_m)^{-1}$ ，其中  $L_m$  是随机选择的节点出发采用邻距离方法获得，全局信息素挥发参数  $\rho = 0.1$ ，信息素挥发参数  $\alpha_1 = 0.1$ ，状态转移规则中的  $q_0 = 0.1$ 。该实验的终止条件为循环次数  $NcMax$  为 500 次。

实验一：信息启发因子 $\alpha$ 从 0.5 到 5，每次增加 0.5，其他参数按照上面的设置。针对上述操作，期望启发式因子 $\beta = 3$ ，上述操作，每组做 10 次实验，数据结果如表 3.4。

表 3.4 信息启发因子变化实验

编号	$\alpha$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	0.5	478.716	457.036	494.539	37.503	204
2	1	452.262	439.392	468.383	28.991	148
3	1.5	434.128	424.692	450.219	25.527	112
4	2	425.094	423.949	429.415	5.466	87
5	2.5	424.972	423.741	429.589	5.848	80
6	3	425.32	423.741	429.747	6.006	91
7	3.5	425.138	423.741	429.415	5.674	97
8	4	426.026	423.741	431.761	8.02	110

编号	$\alpha$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
9	4.5	426.583	423.949	430.897	6.948	118
10	5	428.177	423.741	434.104	10.363	130

从表 3.4 分析得出以下几点：

- (1) 在期望启发式因子  $\beta$  不变的情况下， $\alpha$  的范围在  $\alpha \in [2, 2.5]$  时全局最优路径的平均值较好，通过差值可以得出此时算法的稳定性也较强，通过平均收敛次数可以得出算法的性能较强。
- (2)  $\alpha$  是应用在状态转移公式中的  $[\tau(r, u)]^\alpha$  上，根据我们在 3.1 小节给出信息素强度的范围  $(nL_{mn})^{-1} \leq \tau(r, s) \leq (L_{gb})^{-1}$  可知， $\tau(r, u)$  的范围在  $0 < \tau(r, u) < 1$  时， $\alpha \geq 1$  时， $\alpha$  取值越大，可以知道  $[\tau(r, u)]^\alpha$  就越小，但是该值大小在独立的情况下无法判断其对算法性能的影响。
- (3)  $\alpha$  在 3.5 左右也能取得不错的结果，但是由于  $\alpha$  的值越大，计算的次数越多，消耗的资源也会变多，所以在结果相差无几的情况下  $\alpha$  选择值小的。

实验二：期望启发因子  $\beta$  从 1 到 6，每次增加 0.5，其他参数按照上面的设置。针对上述操作，采用信息启发式因子  $\alpha = 2$ ，每组做 10 次实验，数据结果如表 3.5 所示。

表 3.5 期望启发式因子变化实验

编号	$\beta$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	1	472.41	436.577	493.685	57.108	124
2	1.5	446.294	433.149	464.685	31.536	112
3	2	430.633	424.673	439.711	15.038	103
4	2.5	429.737	424.464	443.744	19.28	98
5	3	424.415	423.741	425.524	1.783	90
6	3.5	424.805	423.741	425.38	1.639	86
7	4	424.088	423.741	424.9	1.159	82
8	4.5	424.104	423.741	425.267	1.526	87
9	5	425.08	423.741	429.38	5.639	95
10	5.5	425.651	423.741	429.38	5.639	97
11	6	425.737	423.741	432.662	8.921	102

从表 3.5 分析得出以下几点：

- (1) 在信息启发因子  $\alpha$  不变的情况下， $\beta$  的范围在  $\beta \in [3, 4.5]$  时全局最优路径的平均值较好，算法的稳定性也较强。
- (2)  $\beta$  是应用在状态转移公式中的  $[\eta(r, u)]^\beta$  上， $\eta_{ru}$  表示由节点  $r$  转移到节点  $u$  的启发程度，它的取值为  $\eta_{ru} = 1/d_{ru}$ ，其中  $d_{ru}$  表示节点  $r$  和节点  $u$  之间的距离， $\eta_{ru}$  的范围在  $0 < \eta_{ru} < 1$ 。在  $\beta \geq 1$  的情况下， $\beta$  取值越大， $[\eta(r, u)]^\beta$  也越小。

综合分析表 3.4 和 3.5 的数据结合状态转移公式来分析结果。由上述分析可知  $\tau(r, u)$  的范围在  $0 < \tau(r, u) < 1$ ， $\eta_{ru}$  的范围在  $0 < \eta_{ru} < 1$ 。

假设  $\alpha$ ， $\beta$  的取值在  $(0, 1)$  之间，各条路径上的  $[\tau(r, u)]^\alpha$  和  $[\eta(r, u)]^\beta$  两者各自差别不大， $\alpha$ ， $\beta$  在上述范围内越大， $[\tau(r, u)]^\alpha$  和  $[\eta(r, u)]^\beta$  两者越接近 1，就削弱了路径上的信息素强度和能见度的作用，大大降低了算法的性能，这已经在表 3.4 中  $\alpha = 0.5$  时得到验证。

如果  $\alpha$ ， $\beta$  的取值均大于 1 时，各条路径上的  $[\tau(r, u)]^\alpha$  和  $[\eta(r, u)]^\beta$  两者各自之间的差别就明显多了，从表 3.4 和 3.5 可以看出算法的性能得到明显的改善。对于  $\alpha$ ， $\beta$  两者的取值，只能给出上述范围，具体取值则需要根据实际问题来组合后通过实验结果来说明。

### 3.3.2 全局信息素挥发因子的影响

在蚁群算法中，蚂蚁在一次循环过程完成之后，需要使用全局信息素更新规则对全局最优路径上的信息素强度进行更新，全局信息挥发因子  $\rho$  对算法的全局搜索能力和收敛速度是有很影响的。

根据全局信息素更新规则的公式(2—8)：

$$\tau(r, s) \leftarrow (1 - \rho) \times \tau(r, s) + \rho \times \Delta \tau(r, s)$$

交换一下顺序则有：

$$\tau(r, s) \leftarrow \tau(r, s) + \rho \times (\Delta \tau(r, s) - \tau(r, s))$$

由全局信息素更新规则的目的可知，信息素强度在更新之后是变大的，所有  $\Delta \tau(r, s) \geq \tau(r, s)$ 。由此可知  $\rho$  越大，那么全局最优路径解上的各条路径上的信息素强度越大，则这些路径被再次选择的可能性也越大，虽然加快了收敛速度，会降低全局的搜索能力；反之  $\rho$  越小，就降低了收敛速度而提高了全局的搜索能力。从上述分析可知， $\rho$  的取值在一定范围内才能保证在加快收敛速度的同时尽可能

保证不陷入局部最优中。

实验采用 Oliver 30 城市 TSP 问题作为仿真对象,城市节点数目  $n=30$ , 蚂蚁数目采用 3.2.1 节的结论  $m=30$ , 初始化方式采用均匀分布, 信息素强度值采用 3.1.2 的结论:  $Q=(nL_{mn})^{-1}$ , 其中  $L_{mn}$  是从随机选择的节点出发采用邻距离方法获得, 信息素挥发参数  $\alpha_1=0.1$ , 状态转移规则中的  $q_0=0.1$ , 信息启发因子  $\alpha=2$ , 期望启发式因子  $\beta=4.5$ , 终止条件为循环次数 NcMax 为 500 次。

全局信息素挥发因子  $\rho \in [0.01, 0.9]$ , 分为两个不同的增长阶段, 第一个增长阶段从 0.01 开始, 每次递增 0.01, 直至 0.1, 第二个增长阶段从 0.1 开始, 每次递增 0.1, 直至 0.9。采用两个不同阶段的生长方式, 是通过实验发现必须包含第一阶段的数据才说明问题。每组做 10 次实验, 数据结果如表 3.6。

表 3.6 全局信息素挥发因子变化实验

编号	$\rho$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	0.01	429.351	424.692	434.509	9.817	163
2	0.02	425.611	423.741	430.36	6.619	134
3	0.03	425.074	423.741	432.565	8.824	115
4	0.04	424.593	423.741	425.82	2.079	110
5	0.05	424.506	423.741	425.267	1.526	101
6	0.06	424.441	423.741	425.989	2.248	94
7	0.07	424.427	423.741	424.692	0.951	82
8	0.08	424.426	423.741	425.267	1.526	88
9	0.09	424.328	423.741	424.9	1.159	84
10	0.1	424.295	423.741	425.267	1.526	78
11	0.2	424.3	423.741	424.9	1.159	80
12	0.3	424.41	423.741	425.267	1.526	96
13	0.4	424.638	423.949	424.9	0.951	99
14	0.5	424.867	423.949	428.673	4.724	112
15	0.6	425.239	423.741	429.99	6.249	127
16	0.7	426.416	423.741	429.99	6.249	130
17	0.8	426.819	423.741	438.382	14.641	157
18	0.9	429.241	423.741	443.207	19.466	177

从表 3.6 分析得出以下两点：

- (1) 全局信息素挥发参数  $\rho \in [0.07, 0.2]$  时，全局最优路径的平均值较好，说明算法的有效性，全局最优路径的差值较小说明算法稳定性较强，而平均收敛次数较小则说明算法性能较好。
- (2) 从表 3.6 的数据可以得出实验的数据对理论的支持：全局信息素挥发参数的值过大或是过小，都影响了算法的性能。

### 3.3.3 启发因子和信息素挥发因子组合的影响

在蚁群算法中，各类参数的作用是相互影响的， $\alpha$ 、 $\beta$ 、 $\rho$  的值通过多次实验组合获得一组较合理的结果。下面的实验是以上述两小节的结论  $\alpha \in [2, 2.5]$ ， $\beta \in [3, 4.5]$ ， $\rho \in [0.07, 0.2]$  来选择参数来组合进行实验。

实验采用 Oliver 30 城市 TSP 问题作为仿真对象，城市节点数目  $n=30$ ，蚂蚁数目采用 3.2.1 的结论  $m=30$ ，初始化方式采用均匀分布，信息素强度值采用 3.1.2 的结论： $Q=(nL_{mn})^{-1}$ ，其中  $L_{mn}$  是从随机选择的节点出发采用邻距离方法获得，状态转移规则中的  $q_0=0.1$ 。该实验终止条件循环次数 NcMax 为 500 次。

每组做 10 次实验，实验数据如表 3.7。

表 3.7 启发因子和信息挥发因子组合实验

编号	$\alpha$	$\beta$	$\rho$	全局最优路径				平均收敛次数
				平均	最优	最差	差值	
1	2	3	0.1	424.546	423.741	425.199	1.458	79
2			0.2	424.039	423.741	425.624	1.883	94
3		3.5	0.1	424.717	423.741	425.179	1.438	80
4			0.2	424.519	423.741	425.464	1.723	92
5		4	0.1	424.377	423.741	425.103	1.362	78
6			0.2	424.744	423.741	425.464	1.723	89
7		4.5	0.1	424.295	423.741	425.267	1.526	82
8			0.2	424.336	423.741	425.267	1.526	80
9	2.5	3	0.1	423.983	423.741	424.9	1.159	82
10			0.2	424.01	423.741	424.9	1.159	78
11		3.5	0.1	424.71	423.741	425.464	1.723	88



编号	$\alpha$	$\beta$	$\rho$	全局最优路径				平均收敛次数
				平均	最优	最差	差值	
12		3.5	0.2	423.973	423.741	424.9	1.159	74
13		4	0.1	424.374	423.741	424.9	1.159	72
14			0.2	424.696	423.949	425.267	1.318	82
15		4.5	0.1	424.578	423.741	425.475	1.734	90
16			0.2	424.563	423.741	425.267	1.526	84

从表 3.7 分析得出以下两点：

- (1)  $\alpha$ 、 $\beta$ 、 $\rho$  组合的取值在上述小节给出的范围内进行的，从表中的结果可以看出，所有组合在 10 次实验中，都可能找出实际的最优解，组合的平均全局最优路径在 424 左右，说明算法在上述取值范围是有效且稳定的，而平均收敛次数也说明了算法的性能相对较好，通过实验来确定最终的组合结果也是合理有效的。
- (2) 从表中的结果来看， $\alpha$ 、 $\beta$  在组合所分布的位置来看，说明  $\alpha$ 、 $\beta$  在合理取值范围内，偏大的取值获得平均全局最优路径结果更合理一些，这与 3.3.1 小节的综合分析结论是一致的。

### 3.4 蚁群算法的蚁群系统状态转移参数的研究

在蚂蚁系统中，蚂蚁完全依赖概率来进行路径选择，有倾向性的对新路径进行搜索，而在蚁群系统中则增加了一种决策规则，利用了关于问题的先验知识，就是利用已知的启发因子和各路径上的信息素强度。

$$s_k = \begin{cases} \arg \max_{u \in allowed_k} \{ [\tau(r, u)]^\alpha [\eta(r, u)]^\beta \}, & q \leq q_0 \\ S, & q > q_0 \end{cases}$$

实验采用 Oliver 30 城市 TSP 问题作为仿真对象，城市节点数目  $n = 30$ ，信息素强度值采用 3.1.2 的结论： $Q = (nL_m)^{-1}$ ，其中  $L_m$  是随机选择的节点出发采用邻距离方法获得的，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 3$ ，全局信息素挥发参数  $\rho = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目为  $m \approx n$ ，即  $m = 30$ ，循环次数为 500 次。

状态转移规则中的  $q_0 \in [0, 0.9]$ ，分为两个不同的增长阶段，第一个增长阶段从

0 开始，每次递增 0.01，直至 0.1，第二个增长阶段从 0.1 开始，每次递增 0.1，直至 0.5，每组做实验 10 次，数据结果如表 3.8。在此次实验中，实验分为两个不同的增长阶段是由于在第二阶段只能大致说明其变化规律，增加第一阶段的实验数据来详细分析在该范围内参数变化对蚁群算法性能影响。

表 3.8 蚁群算法的蚁群系统状态转移实验

编号	$q_0$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	0	427.421	423.741	442.49	18.749	145
2	0.01	425.51	423.741	438.382	14.641	117
3	0.02	425.959	423.741	438.382	14.641	120
4	0.03	425.611	423.741	434.49	10.749	110
5	0.04	424.815	423.741	429.38	5.639	98
6	0.05	424.853	423.741	429.38	5.639	101
7	0.06	424.308	423.741	428.464	4.723	94
8	0.07	424.295	423.741	425.267	1.526	88
9	0.08	424.121	423.741	424.692	0.951	78
10	0.09	424.692	424.237	425.998	1.761	84
11	0.1	424.695	423.741	424.415	0.674	91
12	0.2	437.162	432.146	444.744	12.598	127
13	0.3	466.99	445.974	494.542	48.568	228
14	0.4	479.131	460.267	506.722	46.455	286
15	0.5	522.266	487.085	562.936	75.851	347

从表 3.8 分析得出以下几点：

- (1) 蚁群系统状态转移参数  $q_0$  的范围在  $0.07 \leq q_0 \leq 0.1$  时，全局最优路径的平均值较好，算法的稳定性也较强。
- (2) 利用了关于问题的先验知识，容易陷入局部最优解。目的是为了加快收敛速度，但是过于依赖先验知识，明显会陷入局部最优解。但是如果不采用先验知识，数据结果稍微要差一些。
- (3) 参数  $q_0$  可以调节通过概率来探索新路径的程度，决定算法是应该使蚂蚁的搜索活动集中于最优解的邻域空间内还是探索其他区域。

### 3.5 蚁群算法的参数最优组合原则

在对蚁群算法各个参数进行研究和实验分析的基础上，本节总结出了蚁群算法的参数最优组合原则，它为蚁群算法在解决实际问题提高效率提供了理论支持，具体原则如下：

- (1) 确定初始化信息素强度： $Q = (nL_{mn})^{-1}$ 。
- (2) 根据节点数目来确定蚂蚁数目： $m \approx n$ 。
- (3) 确定 $\alpha$ 、 $\beta$ 、 $\rho$ 等的取值范围 $\alpha \in [2, 2.5]$ ， $\beta \in [3, 4.5]$ ， $\rho \in [0.07, 0.2]$ 。
- (4) 确定 $q_0$ 值，其中 $q_0 \in [0.07, 0.1]$ 。

通过上述步骤给出的部分参数的具体值和部分具体范围反复实验来获得一组效率较高的参数。

### 3.6 本章小结

本章主要对蚁群算法的参数选择进行了分析，并通过实验给出研究结果。首先是信息素强度在规则中的应用，然后分析了蚂蚁数目与节点之间的关系，接着讨论了状态转移规则中的三个参数以及它们的组合关系，并提出了蚁群算法的参数选择的原则，有利于指导用蚁群算法来解决实际的工程问题。

## 第四章 蚁群算法的自适应优化研究

蚁群算法是一种随机搜索算法，与其他模拟进化算法一样，通过候选解组成群体来寻找最优解的进化过程，它包含两个基本阶段：适应阶段和协作阶段。在适应阶段，各候选解根据累积信息不断调整自身结构；在协作阶段，候选解之间通过信息交流，以期望产生性能更好的解。

蚁群算法作为一种新型的进化算法，与其他进化算法同样存在收敛时间长、容易陷入局部收敛而产生局部最优解的问题。如何加快收敛速度而又尽可能避免产生局部最优解是值得探讨的问题。目前已经有文献<sup>[27~29]</sup>对蚁群算法自适应改变参数做了大量的研究，各自提出了自适应改变参数的方式，还有通过与其他算法思想融合来改变参数的方式，比如与遗传算法<sup>[30~31]</sup>、模拟退火算法<sup>[32~33]</sup>等，都取得了许多有价值的成果。

### 4.1 全局最优系数

自适应改变蚁群算法的各类参数是研究蚁群优化算法的途径之一。通过在上一章对蚁群算法的各类参数的研究，发现参数 $\rho$ 和 $q_0$ 有着共同的特征，就是它们的取值过大或是过小，都对蚁群算法的性能产生较大的影响。 $\rho$ 越大，加快了收敛速度，会降低全局的搜索能力； $\rho$ 越小，提高了全局的搜索能力却降低了收敛速度。 $q_0$ 同样具有这类特征，他们之间的不同之处在于他们的取值范围不同。如果能够应用这两个参数的长处，避免它们的短处，则可以提高蚁群算法的性能。如何发挥它们的长处和避免它们的短处，就是在算法开始的过程中它们的值取的稍微大一点，然后根据迭代过程中对全局最优路径的分析来动态将它们减少，需要指出的是它们的取值仍然是在上一章节中结论给出的范围中。前者做法的目的是为了加快算法的收敛速度，后者的目的是为了扩大全局搜索范围而避免陷入局部最优。

自适应改变蚁群算法各类参数的意思是指在蚁群算法的迭代过程中根据某些规则动态改变各类参数。上一章的各类实验已经说明了蚁群算法是一种稳定的算法，在参数设置合理的情况下，多次实验中最好的全局最优路径和最差的全局最优路径差别都不大，而且它们都接近或是等于实际的最优解。这说明那些接近实际最优解，也就是那些陷入了局部最优解的路径，就是在实际全局最优解的路径

附近；换句话说，陷入了局部最优解的路径集合和实际全局最优解的路径的集合绝大多数相同，所以在陷入局部最优解后，只要在这些路径上向下一步搜索的过程中稍微扩大一下范围，就很可能找出全局最优解。

根据上述的分析，本文对算法的优化首先引进全局最优系数： $\delta=n/N$ ， $n$ 表示最优解在连续  $n$  次循环过程后均没有发生变化， $N$  表示初始化时设置的循环总次数。然后根据  $\delta$  作为条件，来自适应改变  $\rho$  的值和  $q_0$  的值。如何改变  $\rho$  的值和  $q_0$  的值是下面小节研究的重点。

## 4.2 蚁群算法参数的自适应优化

### 4.2.1 信息素挥发系数优化

根据  $\delta$  作为条件来自适应改变  $\rho$  的值。前面已经提到过，如果  $\rho$  的值过大，由于全局更新规则只对全局最优路径更新，算法容易陷入局部最优解的状态。如果  $\rho$  的值过小，虽然提高了算法的全局搜索能力，但是降低了收敛速度。所以，可以自适应的改变  $\rho$  的值，使得提高收敛速度的同时不陷入局部最优解。

在连续  $n$  次循环过程最优解都没有变化，即当  $\delta(t_n) > \delta$  时，说明搜索过程可能陷入了局部最优解，此时，根据每条路径上的信息素强度来实时调整该路径上的信息素挥发参数  $\rho(t)$  的值。 $\rho(t)$  的值与蚁群算法的不同之处是在陷入局部最优解后自适应的改变  $\rho$  的值；与文献[27]中的不同之处是在  $t$  时刻路径的全局信息素挥发参数公式，不仅与路径长度相关，还与期望启发式因子  $\beta$  相关。期望启发式因子  $\beta$  作为状态转移规则中的参数，与  $\eta_{ij}$  的关系是  $[\eta(r,u)]^\beta$ ， $\eta_{ij}$  表示由节点  $i$  转移到节点  $j$  的启发程度，它的取值为  $\eta_{ru}=1/d_{ru}$ ， $\beta$  是作为一个与距离长度相关的参数。 $\rho$  的调整公式如下：

$$\rho(t) = \begin{cases} \rho(t-1) \times (\frac{L_{gb}}{L_{mn}})^{\frac{1}{2^\beta}}, & \text{当 } \delta(t) \geq \delta \text{ 时, 且 } if(i, j) \in \text{最优路径} \\ \rho(t-1), & \text{否则} \end{cases} \quad (4.1)$$

其中  $\rho(t)$  表示在  $t$  时刻，全局的信息素挥发参数，取值范围在区间  $[\rho_{\min}, 1)$  内， $\rho_{\min}$  为  $\rho$  的最小值， $L_{mn}$  是随机选择的节点出发采用邻距离方法获得， $L_{gb}$  为当前迭代结束后的全局最优路径， $\beta$  为蚁群算法中的期望启发式因子。

提出  $\rho$  限制在最小值  $\rho_{\min}$  的目的是在于防止  $\rho$  值过小，收敛速度过于慢而导致算法性能降低。由  $L_{gb} < L_{mn}$  可知  $L_{gb}/L_{mn} < 1$ ，在  $\beta$  在取值范围大于 1 的情况下， $\beta$

越大则  $(L_{gb}/L_{nn})^{1/2^\beta}$  越接近 1。

#### 4.2.2 自适应的蚁群系统状态转移参数优化

根据  $\delta$  来自适应改变  $q_0$  的值。 $q_0$  作为蚁群算法中状态转移规则中的参数，是用来调节状态转移规则中究竟该使用概率规则还是先验知识的规则，它的大小决定了利用先验知识与探索新路径之间的相对重要性。如果  $q_0$  的值过小，则更多使用先验知识的规则，这样以前蚂蚁走过的路径对后来的蚂蚁更加具有吸引力，算法容易陷入局部最优解的状态。如果  $q_0$  的值过大，则更多使用概率规则。调整  $q_0$  的值可以调节探索新路径的程度和决定是否使蚂蚁的搜索活动集中于最优解的邻域内。

$q_0$  的调整公式如下：

$$q_0(t) = \begin{cases} q_0(t-1) \times \left(\frac{L_{gb}}{L_{nn}}\right)^{\frac{1}{2^\alpha}} & \text{当 } \delta(t_n) \geq \delta \text{ 时} \\ q_0(t-1), & \text{否则} \end{cases} \quad (4.2)$$

#### 4.2.3 全局最优系数的影响

该实验是基于优化的自适应蚁群算法求解 Oliver 30 TSP 问题。参数的初始化情况如下：由于  $\tau_0$  通过给出的问题即可计算得出， $\tau_0 = (nL_{nn})^{-1}$ ， $n = 30$ ； $L_{nn}$  是随机选择一个节点，由最近邻距离方法获得的路径长度。 $\Delta\tau(r,s)$  可以近似取  $(L_{nn})^{-1}$ ，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 4.5$ ，全局信息素挥发初始参数  $\rho = 0.1$ ， $\rho_{\min}$  为 0.05，状态转移参数初始值  $q_0 = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目为采用上一章的结论  $m \approx n$ ，即  $m = 30$ ，循环次数为 500 次。

全局最优系数  $\delta \in [0.01, 0.1]$ ，从 0.01 开始，每次递增 0.01，直至 0.1，每组做 10 次实验，数据结果如表 4.1。选择这个范围的理由是因为在这个范围内，实验后发现在 0 到 0.1 之间具有较好的解。在数据列表中，是根据  $\delta$  的由小变大的范围来确定阶段的。

表 4.1 全局最优系数取值范围实验

编号	$\delta$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
1	0.01	427.069	423.741	438.382	14.641	89

编号	$\delta$	全局最优路径				平均收敛次数
		平均	最优	最差	差值	
2	0.02	425.338	423.741	432.662	8.921	75
3	0.03	424.767	423.741	429.38	5.639	70
4	0.04	423.948	423.741	424.692	0.951	59
5	0.05	423.988	423.741	424.9	1.159	65
6	0.06	424.179	423.741	425.267	1.526	73
7	0.07	424.047	423.741	428.464	4.732	79
8	0.08	424.72	423.741	434.9	11.159	87
9	0.09	425.548	423.741	438.382	14.641	92
10	0.1	425.684	423.741	441.403	17.662	95

从表 4.1 分析可得：

- (1)  $\delta \in [0.04, 0.05]$  之间，全局最优路径的平均值较好，算法的稳定性也较强。
- (2)  $\delta$  过大或是过小，实验效果变差。 $\delta$  过小导致全局信息素挥发系数  $\rho$  和状态转移参数  $q_0$  变化过快，未发挥作用；过大就使得这两个系数变化较慢，新的规则所起的作用较少。

### 4.3 基于自适应的蚁群算法描述

以 TSP 为例，优化后的自适应蚁群算法的具体步骤实现如下：

- (1) 初始化各类参数，城市节点数目  $n$ ，信息素强度值  $Q = (nL_m)^{-1}$ ，信息启发因子  $\alpha$ ，期望启发式因子  $\beta$ ，全局信息素挥发参数的初始值  $\rho$  以及限制条件  $\rho_{\min}$ ，局部信息素挥发参数  $\alpha_1$ ，蚂蚁数目采用  $m \approx n$ ，状态转移规则中的  $q_0$ ，最大循环次数  $NcMax$ 。其它的包括： $C$  表示节点集， $s$  表示  $tabu_k$  的列数，初始化  $s = 1$ ；循环次数  $N_c = 0$ ，设置，蚂蚁个体  $k$ ；
- (2) 循环次数  $N_c$  开始自增： $N_c \leftarrow N_c + 1$ ；
- (3) 将源节点置于禁忌表  $tabu_k$  表中的每行中的第一列， $m$  为行数；
- (4) 每个蚂蚁开始遍历整个图中的节点，蚂蚁编号  $k \leftarrow k + 1$ ；
- (5) 根据  $q_0$  的值来使用伪随机比例规则，计算概率选择下一个节点  $j$ ，其中  $j \in allowed_k$ 。禁忌表  $tabu_k$  是用来保存第  $k$  只蚂蚁的行走路径的表，即禁忌表每行都是保存某个蚂蚁的行走路径；

- (6) 将蚂蚁  $k$  移动到新的节点  $j$ ，应用局部更新规则计算该路径上的信息素强度，然后修改禁忌表  $tabu_k$ ；
- (7) 如果  $\{C - tabu_k\}$  不为空，即为遍历完整个节点集合  $C$ ，则跳转到第(5)步，否则继续执行下面的步骤；
- (8) 蚂蚁  $k$  一次循环结束，应用全局更新规则，更新当前全局最优路径上的信息素；
- (9) 根据每次迭代获得的当前全局最优路径重复的次数来计算全局最优系数  $\delta$ ，由已经给出的计算公式此来更新全局信息素挥发系数  $\rho$  和状态转移参数  $q_0$ ；
- (10) 如果  $k \leq m$ ，则跳转到第(4)步，否则继续执行下面的步骤
- (11) 如果循环次数  $N_c < NcMax$ ，则清空禁忌表  $tabu_k$ ，跳转到第(2)步，否则结束循环输出计算结果。

分析优化后的自适应的蚁群算法，对比 2.2.3 蚁群算法本身的实现步骤，重点在加入了第(9)步，完成了自适应参数的变化来优化蚁群算法的性能。

#### 4.4 在 TSP30、TSP50、TSP75 不同节点规模下验证

采用 Oliver 30、Eilon 50、Eilon 75 三个著名的城市 TSP 问题作为仿真对象。通过这三个 TSP 问题来将优化的自适应蚁群算法与其他算法进行比较，主要从平均最优路径、最优路径差值、平均收敛次数等三个方面来比较算法的有效性、稳定性、性能，从实验上来证明本文对算法的优化是合理的。

对于 Oliver 30 问题，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 4.5$ ，全局信息素挥发参数  $\rho = 0.1$ ，状态转移规则中的  $q_0 = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目为  $m \approx n$ ，即  $m = 50$ ，全局最优系数  $\delta = 0.04$ 。该实验的终止条件为循环次数  $NcMax$  为 500 次。优化后的蚁群算法做 10 次实验来分析，获得结果如表 4.2。

表 4.2 本文优化的蚁群算法同其他算法的比较实验(Oliver 30)

算法名称	全局最优路径		平均收敛次数
	平均	最优	
本文优化的蚁群算法	423.931	423.741	69
文献[21]优化的蚁群算法	433.25	425.29	75
文献[22]优化的遗传算法	426.25	423.74	100



对于 Eilon 50 问题，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 4.5$ ，全局信息素挥发参数  $\rho = 0.1$ ，状态转移规则中的  $q_0 = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目为  $m \approx n$ ，即  $m = 50$ ，全局最优系数  $\delta = 0.04$ 。

该实验的终止条件为循环次数 NcMax 为 500 次。优化后的蚁群算法做 10 次实验来分析，获得结果如表 4.3。

表 4.3 本文优化的蚁群算法同其他算法的比较实验(Eilon 50)

算法名称	全局最优路径		平均收敛次数
	平均	最优	
本文优化的蚁群算法	433.427	428.101	92
文献[30]优化的蚁群算法	436.54	428.39	96
文献[36]优化的遗传算法	435.065	428.871	32

对于 Eilon 75 问题，信息启发因子  $\alpha = 2$ ，期望启发式因子  $\beta = 4.5$ ，全局信息素挥发参数  $\rho = 0.1$ ，状态转移规则中的  $q_0 = 0.1$ ，局部信息素挥发参数  $\alpha_1 = 0.1$ ，蚂蚁数目为  $m \approx n$ ，即  $m = 75$ ，全局最优系数  $\delta = 0.04$ 。

该实验的终止条件为循环次数 NcMax 为 1000 次。优化后的蚁群算法做 10 次实验来分析，获得结果如表 4.4。

表 4.4 本文优化的蚁群算法同其他算法的比较实验(Eilon 75)

算法名称	全局最优路径		平均收敛次数
	平均	最优	
本文优化的蚁群算法	549.673	542.309	178
文献[37]优化的蚁群算法	551.763	547.496	326
文献[38]优化的遗传算法	597.818	566.819	425

通过上面表 4.2~表 4.4 可以看出，优化后的自适应蚁群算法平均结果、最优结果说明比文献所列的优化的遗传算法和优化的蚁群算法好，说明本文对蚁群算法的优化是有效的、稳定的。同时，本文对蚁群算法的优化在平均收敛次数上要少于文献中优化的蚁群算法，说明本文优化后的蚁群算法在性能上要高于后者。针对上述不同的节点规模，优化后的自适应蚁群算法在平均结果、最优结果值较好，说明本文对蚁群算法的优化适应节点规模的变化。

## 4.5 本章小结

本章根据蚁群算法的参数  $\rho$  和  $q_0$  在算法运行中的特点，对蚁群算法进行了自适应优化的研究。首先对参数  $\rho$  和  $q_0$  进行了分析，然后对算法进行了优化。引进全局最优系数  $\delta$ ，提出了自适应的优化思路和优化的理由，然后根据参数  $\rho$  和  $q_0$  的特点，提出了自适应变化公式，并通过实验完成了  $\delta$  的范围研究。最后在实验中比较优化的自适应蚁群算法与其他算法的性能，给出研究结论。

## 第五章 QoS 组播路由技术研究

QoS 组播路由是根据实际业务的需求，取两部分相关的技术长处来得到的。第一部分就是 QoS 技术，第二部分就是组播技术。关于 QoS 技术、组播技术的概述，如何在组播中应用 QoS，抽取 QoS 网络模型，目前的相关算法和协议，面临的各种问题如何去解决，都是本章要研究的内容。

### 5.1 QoS 组播路由技术特点

#### 5.1.1 基本概念

QoS(Quality of Services)，即服务质量，定义为在网络传输中满足一系列的要求，例如带宽、延时、延时抖动、丢包率以及代价等度量。这些 QoS 度量用来衡量业务的各种需求特性，同时也是对相应网络状态的描述。

QoS 是网络的一种安全机制，是用来解决网络延时和阻塞等问题的一种技术。在正常情况下，如果网络只用于特定的无时间限制的应用系统，并不需要 QoS，比如 Web 应用，或发送 E-mail 等。对于媒体应用，QoS 就表现出来了它必要性。当网络过载或拥塞时，QoS 能确保重要业务量不受延时或丢弃，同时保证网络的高效运行。QoS 是网络与用户之间以及网络上互相通信的用户之间关于信息传输与共享的质的约定，服务质量包括两个方面：用户的要求和网络服务提供者的行为<sup>[39]</sup>。

路由通信问题一般可以分为三大类：单播路由(Unicast Routing)、多播路由(Multicast Routing)、广播(Broadcast Routing)。单播路由问题是在给定的源节点和目的节点之间搜索满足约束的路径，它在发送者和每一接收者之间都需要独立的路径。多播路由问题是在给定的源节点和目的节点集合之间搜索满足约束的树。广播是指在 IP 子网内广播数据包，所有在子网内部的主机都将收到这些数据包。组播路由技术则是找到从给定的源节点和目的节点集合之间的树。

QoS 组播路由则是为了找到一棵从给定的源节点到目的节点集合同时满足多约束条件而为用户提供 QoS 服务的组播树。QoS 网络提供更高优先服务的一种能力，包括专用带宽、抖动控制和延时、丢包率的优化。它是一种安全机制，是用来解决网络延时和阻塞等问题的一种技术。

QoS 的优点:

- (1) 提供服务保障: 在不增加对带宽的占用情况下提供更好的服务。
- (2) 提供有差别服务: 针对用户的不同要求, 网络服务商可以提供不同的服务。

QoS 的缺点:

- (1) 网络状态的汇聚导致状态信息不精确, 这使得算法必须具有在一定程度上容忍不精确网络状态信息的能力。

组播, 是 IP 网络数据传输三种方式之一。它在发送者和所有接收者之间实现单点对多点的网络连接。如果源节点同时给多个的目的节点传输相同的数据, 只需发送一份数据, 即可到达所有的目的节点。主机之间“一对一组”的通讯模式, 也就是加入了同一个组的主机可以接受到此组内的所有数据, 网络中的交换机和路由器只向有需求者复制并转发其所需数据。

组播的优点:

- (1) 减少对带宽的占用: 如果一个发送者给多个接收者同时发送相同的数据, 单播传输方式需要在发送者和每一个接收者之间实现点对点的数据传输, 并且需要为每一个接收者发送一份数据的拷贝, 所以随着接收者数量的增加, 对带宽的需求也将增加。组播技术则不同, 发送者只需发送一份数据, 数据只在组播树的分叉处进行复制, 向不同的组播树枝传输数据拷贝, 所以带宽需求不会随着接收者数量的变化而改变。可以说从本质上减少对整个网络带宽的需求。
- (2) 减轻了发送者的负载: 组播技术中发送者只需要发送一份数据的拷贝, 不需要向每一个接收者发送数据的拷贝。这样就减轻了发送者的负载, 具备广播所具备的优点。

组播的缺点:

- (1) 与单播协议相比没有纠错机制, 发生丢包错包后难以弥补, 但可以通过一定的容错机制和 QoS 加以弥补。
- (2) 现行网络虽然都支持组播的传输, 但在客户认证、QoS 等方面还需要完善, 这些缺点在理论上都有成熟的解决方案, 只是需要逐步推广来应用到网络中。

### 5.1.2 组播路由树型结构

组播是源节点向目的节点传递数据报, 有三种基本类型: 洪泛树、有源树、

共享树<sup>[25]</sup>。

(1)洪泛树：它不构造组播树，它只向前传送组播路由，算法简单。它的基本原理如下：当组播路由器收到发往某个组播地址的数据包后，首先判断是否是首次收到该数据包，如果是首次收到，则将其转发到所有组播路由器接口上，以确保数据包能够到达所有接收者；如果不是首次收到，则抛弃该数据报。洪泛法需要维护“首次收到”列表的状态，增加了路由器的负担，虽然可以保证不对相同的数据包进行二次转发，但不能保证对相同数据包只接收一次，这样就增加了该路由器的数据收发的负担。

(2)有源树：也称为基于信源的树或是最短路径树(SPT, Shortest Path Tree)，它是构造以组播源端为根并且从根到所有接收端的路径均为最短的分布树。如果组播组中有多个组播源，则必须为每个组播源构造一棵组播树。由于不同组播源发出的数据包被分散到各自分离的组播树上，因此采用 SPT 有利于网络中数据流量的均衡。同时，因为从组播源到每个接收者的路径最短，所以端到端(end-to-end)的延时性能较好，有利于流量大、延时性能要求较高的实时媒体应用。有源树(SPT)的缺点是要为每个组播源构造各自的分布树，当数据流量不大时，构造 SPT 的开销相对较大。

构造有源树(SPT)有多种算法，当前 Internet 上常用的算法包括以下两种：

#### (a)反向路径正向发送(RPF, Reverse Path Forwarding )

它是已经在 IP 组播中广泛应用的有源树(SPT)路由算法，算法如下：当节点接收到一个分组之后，检查接收到分组的端口是否是该节点到分组信源的最短路由上的端口：如果是，则该节点向除该端口之外的所有端口转发此分组；否则，抛弃该分组。RPF 假定信道对称，暗含按照最小延时的原则对组构造一棵生成树。

由于 RPF 属于广播算法，为了使它适应组播，进行了两点改进，根据信源建立路由器与链路的“父子”关系，只有父路由器才允许向子链路发送该信源的分组，从而消除了挂在同一网段上多个路由器同时向该网段上发送该信源分组产生的重复分组问题；从树上移去下行链路不存在接收节点的路由器(RPM, Reverse Path Multicast)，从而降低由于 RPF 的广播特性使特定组业务对与该组无关的路由器和子网产生的影响。

#### (b) Dijkstra 算法

Dijkstra 算法的基本思想是每一次将距离信源最近的节点加入到树上，构造一棵生成树，然后将非成员树叶删除。暗含要求信源具备所有信宿的地址信息。Dijkstra 算法实现简单，时间复杂度为  $O(n^2)$ ，但算法集中执行，需要全网拓扑，一

次性耗时长,大量信源同时发送数据时 CPU 负载重,所得的组播树的费用也较高。Dijkstra 算法所得到的结果是精确的,算法计算时间复杂度的量级属于多项式量级。算法中,每个节点都拥有它所在网络的全部拓扑信息。各个节点采用全部拓扑信息来计算从源节点到目的节点的最短路径不会产生环路。

(3)共享树,也称为 RP 树,是指为每个组播选择一个共用的根(称为汇聚点,RP),以 RP 为根建立的组播树。同一组播的组播源端将所要组播的数据单播到 RP,再由 RP 向其他成员进行转发。目前,共享树主要有 Steiner 树,受约束 Steiner 树和有核树(CBT)。共享树在路由器所需存储的状态信息的数量和路由树的总代价两个方面都具有较好的性能。当组的规模较大,而每个成员的数据发送率较低时,使用共享树比较适合。但当通信量大时,使用共享树将导致流量集中及根(RP)附近的瓶颈。这里介绍一下 Steiner 树,受约束 Steiner 树和有核树(CBT)。

Steiner 树:它的目的在于使组播树的总体代价最小化,是 NP 完全问题。由于使用资源比较少,因此产生拥塞的可能性也较小。但是由于 Steiner 树不稳定,树的形状会随着接收端的改变而发生变化。如果组播组成员包括网络中的所有节点,Steiner 树问题就可归为最小生成树问题。没有约束的 Steiner 树问题可用来解决树的优化问题。

受约束 Steiner 树(CMST, Constrained Minimal Steiner Tree):在对树的代价进行优化的同时,有时还需要考虑各种约束条件,该类问题就是受限 Steiner 树问题,它是 QoS 组播路由中应用最广研究最多的问题。约束条件包括链路约束、路径约束和树约束。链路约束定义了从源端到目的地的每一条链路的约束,如带宽约束;路径约束定义了从源到目的地的一条路径上端到端 QoS 需求,如延时;树约束定义了对组播中整个组播树的约束,例如对组播树延时的约束是对树中从源到所有目的地的路径中最大延时的约束。

有核树(CBT):CBT 以选定的点为根,由根到所有组成员的最短路径合并而成的树,即以选定的汇合点作为组播树的根,其他组成员按照最短路径的原则与汇合点建立连接,构造一棵由所有发送节点共享的树。有核树中网络费用性能和延时特性都不是最优的,此外,有核树不具备广播特性,即数据只发向明确发出加入组请求的节点,这样就避免了大量无效分组的扩散,适合于接收者分布稀疏的情况。

Steiner 树是以组播源点为根形成的最小费用树,是一个 NP 完全问题,下面仅介绍一些典型的 Steiner 树的算法。

#### (a) KMB 算法<sup>[41]</sup>

它是由 Kou, Markowski 和 Berman 为解决 Steiner 树问题提出的。KMB 首先从原来的网络拓扑图中构造一个完全图，将组播成员间的最短费用路径抽象为完全图的边，然后在完全图的基础上计算最小生成树，最后将树中的各边还原为最短路径，并删除还原操作中形成的环路，即得到 Steiner 树问题的近似解。KMB 算法简述如下：

第一步：对于给定连通图  $G=(V,E)$ ，给定点集  $D_M \subseteq V$ ；构造完全图  $G'=(D_M, E')$ ，其中  $\forall (u,v) \in E'$  是图  $G$  中节点  $u$  到节点  $v$  的最短路径，即边  $(u,v) \in E'$  的代价等于图  $G$  中节点  $u$  到节点  $v$  的最短路径的代价。

第二步：构造图  $G'$  的最小生成树  $T'$ ；

第三步：将生成的树  $T'$  的边扩展成图  $G$  中最短路径，形成图  $G$  的子图  $G''$ ；

第四步：求  $G''$  的最小生成树  $T''$ ；

第五步：从  $T''$  中删除度为 1 的 Steiner 节点，得到准 Steiner 树  $T_{KMB}$ 。

#### (b) BSMA 算法<sup>[42]</sup>

由 Mehrda Parsa 等人提出，解决延时约束的树的优化问题。它的思路是首先用 Dijkstra 的最短路径算法求出从源节点到各目的节点的延时最短路径，构成一棵组播树，在不违反延时约束的情况下，用费用更低的路径取代原路径，在替换边时用到第  $k$  条最短路径算法。这个算法求到的组播树的费用较小。BSMA 有以下优点：(1) 算法的度量和费用度都为实数；(2) 终点的延时约束可以互不相同。缺点是复杂度太高，不适用于求解大型网络的组播路由问题。BSMA 算法可以通过调节参数  $k$  在组播树费用和计算时间之间进行权衡。

### 5.1.3 组播路由算法分析

针对 QoS 路由问题，可以从不同的角度对这些 QoS 路由算法进行分类。根据不同的网络状态信息维护方式和可行路径搜索方式，QoS 路由策略主要包括源路由、分布式路由和层次结构路由。

#### (1) 源路由组播算法

源路由假设每个网络节点都持有整个网络状态信息，包括网络的拓扑结构，每个链路的状态。基于这些信息，路径的选择在本地的源节点进行，根据网络的状态信息计算出源节点到目的节点集合的组播树。

它的优点是通过对分布式问题转化成集中式问题使算法本身更加简洁。源节点维护全局网络状态，本地计算整个路由可以避免分布式算法带来的很多不便，

还可以确保无环路由。另外，源路由算法概念简单，易于实现、评估、调试及升级。另外，集中式启发式算法要比分布式启发式算法更容易设计。

它的缺点也是源路由算法面临的最大问题，即算法的可扩展性较差。对于源路由算法，每个网络节点必须维护全局网络状态信息，而这些信息为了能够更精确地反应网络的动态特性，需要相应链路状态协议来维持和更新网络中的每个节点的状态。对于一个较大的网络，频繁的信息更新会造成很大的网络负载。另外，QoS 路由协议提供的网络状态信息的不精确性对源路由算法的影响相对较大，导致算法得到的路径最终不能满足业务需求。源路由算法往往具有很高的计算复杂性，尤其在网络较大时更是如此。

## (2) 分布式路由组播算法

采用分布式处理的方式，通过各个相关节点进行路由计算来选择路径，需要通过使用控制信息来获得网络节点的状态信息。每个节点持有的网络状态信息共同用于整个路径的搜索。

优点：在分布式算法中，对于路由的计算是在源节点和目的节点之间的节点上进行的。因此，具有较好的可扩展性和较快的响应时间，并且可同时并行计算多条路径，增大了算法成功的概率。

缺点：基于全局状态的分布式路由算法或多或少面临与源路由相同的问题，而不需要任何全局状态信息的分布式路由算法甚至需要发送更多的消息。相当一部分 QoS 路由问题是 NP 完全问题，对这些问题还没有什么有效的分布式 QoS 路由算法。另外，由于各节点维护的网络状态信息不一致，可能产生环路。

## (3) 层次结构组播路由算法

在层次结构路由算法中，采用了一种等级体系结构，路由节点按区域聚集成组，这些组又进一步聚集成更高层次的组，最终形成一个多层次结构。每个物理节点都要保存一份汇聚后的全局状态。

优点：层次结构路由算法的明显优点是可扩展性好，网络节点只需要维护部分全局状态信息即可。层次结构路由算法继承了源路由算法的简洁性，又避免了源路由算法扩展性差的问题。另外，同一层次路由计算的共享，使它继承了分布式算法的一些优点。

缺点：网络状态的汇聚导致状态信息不精确，这使得算法必须具有在一定程度上容忍不精确网络状态信息的能力。

### 5.1.4 组播路由协议



要确定组播路由，首先要收集网络状态信息，路由算法再根据这些信息来确定路由，这些信息的收集由路由协议来完成。下面介绍其中最常见的组播路由协议。

### (1) 距离向量组播路由协议(DVMRP, Distance Vector Multicast Routing Protocol) [43]

DVMRP 是在单播路由选择信息协议(Routing Information Protocol, RIP)基础上扩展而来，两者都使用距离向量算法得到网络的拓扑信息，不同之处在于路由选择信息协议(RIP)根据路由表向前转发数据，而 DVMRP 则是基于反向路径转发(Remote-path Forwarding, RPF)。这种基于距离向量算法的组播路由协议，目前已基本上被 PIM 和 MOSPF 所取代，为了使新加入的组播成员能及时收到组播数据，DVMRP 采用定时进行洪泛，发送数据包给将组播信息发送到网络上的每一点，这种方法导致大量路由控制数据包的扩散，这部分开销限制了网络规模的扩大，另外在没有接收者的情况下定时进行洪泛会大大地浪费了网络的带宽资源，在路由器中存储大量的组播转发信息，占用了宝贵的路由器资源。目前提出了分层 DVMRP，即对组播网络划分区域，在区域内的组播可以按照任何协议进行，而对于跨区域的组播则由边界路由器在 DVMRP 协议下进行，这样可大大减少路由开销。

### (2) 开放式组播最短路径优先协议(MOSPF, Multicast Open Shortest Path First)<sup>[44]</sup>

MOSPF 是一种基于链路状态的路由协议，是对单播 OSPF 协议的扩展。MOSPF 是利用点到点的链路状态数据库，以 OSPF V2 为基础的组播路由协议。由于 OSPF 应用 Dijkstra 算法进行路由选择，因此每个节点都要保存全网的拓扑信息。所有节点对网络的看法是一致的，它不需发送任何分组，节点就可根据全网链路状态表计算组中每个信源的组播树，而且各节点对该树的看法一致。因此，链路利用率比 DVMRP 高。为了减少计算量，MOSPF 可以按需执行算法，即只有当一个节点收到一个信源关于某个组播组的第一个分组时，才执行算法。这种做法的缺点是对第一个分组带来较大延时。MOSPF 的最大优点是享有 OSPF 对网络拓扑的变动快速的反应能力。然而这个快速反应的能力是以路由器 CPU 资源的巨大消耗为代价的。随着网络中组数量的增加，这种消耗也在迅速增加。MOSPF 继承了 OSPF 对网络拓扑的变化响应速度快的优点，但拓扑变动使所有路由器的缓存失效重新计算 SPT，因而消耗大量路由器 CPU 资源。这就决定了 MOSPF 不适合高动态性网络(组成员关系变化大、链路不稳定)，而适用于网络连接状态比较稳定的环境。

### (3) 无关组播协议(PIM, Protocol Independent Multicast)

PIM 定义两种模式：密集模式(Dense-Mode)<sup>[45]</sup>和稀疏模式(Sparse-Mode)<sup>[46]</sup>，PIM 不依赖于某一特定单播路由协议，它可利用各种单播路由协议建立的单播路由表完成 RPF 检查功能，而不是维护一个分离的组播路由表实现组播转发。由于 PIM 无需收发组播路由更新，所以与其它组播协议相比，PIM 开销降低了许多。PIM 设计的出发点在于在广域网范围内同时支持共享树与有源树，并能完成两者之间的灵活转换，因而集中了它们的优点，提高了组播效率。在组员密集时以广播形式传输数据，然后从树上删除不存在接收节点的分支；而组员分布稀疏时，构造共享树传送，避免分组的广播开销。

#### (4)有核树组播路由协议(CBT, Core-Based Trees)<sup>[46]</sup>

CBT 的基本目标是减少网络中路由器组播状态，以提供组播的可扩展性。为此，CBT 被设计成稀疏模式(与 PIM-SM 相似)。CBT 使用双向共享树，双向共享树以某个核心路由器为根，允许组播信息在两个方向流动。这一点与 PIM-SM 不同(PIM-SM 中共享树是单向的，在 RP 与组播源之间使用 SPT 将组播数据转发到 RP)，所以 CBT 不能使用 RPF 检查，而使用 IP 包头的目标组地址作检查转发缓存。从路由器创建的组播状态的数量来看，CBT 比支持 SPT 的协议效率高，在具有大量组播源和组的网络中，CBT 能把组播状态优化到组的数量级。CBT 将组播扩张限制在接收者范围内，即使第一个数据包也无需在全网扩散，但 CBT 导致核周围的流量集中，网络性能下降。所以某些版本的 CBT 支持多个核心以平衡负载。

## 5.2 QoS 路由数学模型分析

### 5.2.1 QoS 路由参数的选择

网络应用产生的业务流通常对网络传输提出多样的 QoS 质量保证要求。QoS 的目标是通过提高网络总体吞吐量，有效地利用网络资源。

典型 QoS 度量参数主要包括以下几方面：

- (1) 延时(Delay): 两个节点之间发送和接收数据包的时间间隔，可以分为固定部分和可变部分。固定部分主要是指数据包转发和在传输链路上的传播的延时；可变部分主要是数据包在网络节点的队列中等待引起的延时。
- (2) 延时抖动(Delay jitter): 指延时的变化，即相邻两个数据包到达接收者的时间间隔与发送者发送这两个数据包的时间间隔之间的差异。
- (3) 带宽 (Bandwidth): 是网络发送数据能力的一种度量，指单位时间内传送

的数据包数量。

- (4) 丢包率(Packet-Loss): 是指在网络中传输数据包时丢弃数据包占全部传输包的比率。造成数据包丢失的原因很多,最主要的原因是网络本身不可靠,数据包可能因为传输错误(如校验错误)而丢失。此外,节点在网络拥塞时也可能引起数据包丢失。

- (5) 代价(Cost): 可以是链路使用价格或链路上某些网络资源使用状态的测度。

选择合理的度量参数是进行 QoS 路由设计的第一步,而选择度量参数时通常不同的 QoS 度量具有不同的特性, QoS 度量可以分为可加性度量、可乘性度量和最大最小性度量三类<sup>[6]</sup>。

对于图  $G$  中的路径  $P=(j,k,l,...,u,v)$ ,  $w_i(u,v)$  表示链路  $(u,v)$  的第  $i$  个度量,整个路径  $P$  的第  $i$  个属性记为  $w_i(P)$ , 则以上三类度量的定义如下:

定义 1: 可加性度量, 若  $w_i(P)=w_i(j,k)+w_i(k,l)+...+w_i(u,v)$ , 则称路径  $P$  的第  $i$  个属性为可加性度量。

定义 2: 可乘性度量, 若  $w_i(P)=w_i(j,k) \times w_i(k,l) \times ... \times w_i(u,v)$ , 则称路径  $P$  的第  $i$  个属性为可乘性度量。

定义 3: 最大最小性度量, 若  $w_i(P)=\min\{w_i(j,k), w_i(k,l), ..., w_i(u,v)\}$  或者  $w_i(P)=\max\{w_i(j,k), w_i(k,l), ..., w_i(u,v)\}$ , 则称路径  $P$  的第  $i$  个属性为最大最小性度量。

可加性度量包括延时、延时抖动、花费, 分组丢失率为可乘性度量, 带宽为最大最小性度量, 即路径带宽为路径上瓶颈链路的带宽。

上面的度量参数能够有效地反映网络链路的基本特征。选择度量参数时, 应该尽可能相互独立以避免多个参数之间存在冗余信息。由于在网络应用中, 在多个参数相互关联的情况下, 难以对每个度量参数的性质进行独立的讨论, 此时选择的度量参数之间的关系是值得研究的对象。

虽然复杂的路由算法可以更合理地满足各种 QoS 需求, 但是在计算路由时资源开销过大使得网络的扩展性变差, 无法用于大规模网络中。因此, QoS 路由度量参数选择一方面需要考虑满足 QoS 请求业务, 同时由于路由算法的可扩展性要求, 参数选择时如何降低算法的复杂度成为需要考虑的另一重要因素。

### 5.2.2 QoS 路由的网络模型 $\tau$

QoS 网络的拓扑结构可以抽象为一个加权图  $N(V, E)$ , 其中  $V$  表示为网络中节点的集合,  $E$  表示网络中双向链路的集合。对于任意的网络节点  $u$ 、 $v$ , 如果存在  $u$

到  $v$  的链路  $e = (u, v)$ ，则存在另外的一条链路  $e' = (v, u)$ ，这类链路具有对称性的网络称为对称网络，否则称为非对称网络。本文后面研究的 QoS 网络的组播路由是以对称网络为研究对象。

从网络的抽象模型可知，网络的拓扑结构由节点和节点之间的链路两类要素构成，上一小节对两类要素的属性进行了定义，并分析了它们的特征。由于在实际的应用中，这两类要素是主要因素，所以本文的后续研究是对实际的应用进行了抽象，但是突出本质，研究完全具有其实用的价值。另外值得提到的一点是两类要素的属性在模型中都是已知的，就是说各个节点都保存这个网络的拓扑结构的各类信息。这类做法是可行的，理由是经典的链路状态路由算法便是利用了全局的信息，下面介绍到的有源树也是以全局信息为基础来构造的。当然，仅用邻节点信息来构造组播树也是一种思路。仅用邻节点信息的经典路由算法就是距离矢量路由算法。两者各有优缺点，取决于具体的算法，根据其特点来选择不同的方式。

## 5.3 QoS 组播路由的研究

### 5.3.1 QoS 组播路由技术面临的挑战

由于传统的 IP 网络最初是为数据传输而设计，采用单播即点对点的方式作为通信模式。随着数据传输的对象发生变化，出现了一对多和多对多的数据传输方式，便提出了组播的数据传输方式。然而，随着多媒体业务的广泛应用，人们对网络的服务质量又提出了要求。目前 QoS 组播路由问题是国内外研究的一个热点问题。

组播路由的目标是寻找一系列从源节点出发并最终到达所有目的节点的路径，因此组播路由的算法要比单播路由算法复杂一些。在组播路由中提供 QoS 支持，即满足不同的应用对延时、延时抖动、丢包率和带宽不同的要求，是研究 QoS 组播路由算法需要面对的难题。要实现组播通信，最有效的一种方式就是构造组播树(Multicast Tree)。组播树是一棵覆盖所有组播组成员的生成树。所以，组播路由算法就是研究如何在网络中建立一棵组播树。

构造满足多个约束条件的最小代价的组播树是研究 QoS 组播路由技术的关键之一。目前有许多经典的组播路由算法计算组播树<sup>[48]</sup>。斯坦利树(Steiner Tree)，它的目的在于使组播树的总体代价最小化；延时受限斯坦利树(Delay Constrained

Steiner Tree), 它是寻找满足延时受限要求同时费用为最小的组播树。建立满足多个 QoS 约束的最小代价树, 它是提供满足 QoS 需求的多播服务的关键。关于受约束最小 Steiner 树问题, 是 NP 完全问题。

目前针对解决 QoS 组播路由这类 NP 完全问题, 很多人采用启发式算法等来解决此类问题, 但该类算法求解较为复杂, 不能很好地解决问题。通过对智能优化算法的研究, 发现智能优化算法是解决复杂问题的一个重要途径, 也具有广阔的发展前景。智能算法具有并行性、自适应性和自学习性的优点, 已经在各个研究领域得到了广泛的重视。目前智能优化算法的研究主要集中在蚁群算法、神经网络、遗传算法、模拟退火算法和禁忌搜索算法。蚁群算法作为一种仿真优化智能算法, 在解决 NP 完全问题上显示出了强大的生命力和进一步的发展潜力。

采用蚁群算法来解决 QoS 组播路由问题, 在国内外都处在研究阶段。人们对蚁群算法的研究领域已经由当初解决旅行商问题(TSP)领域渗透到其他多个应用领域。解决 QoS 组播路由问题是蚁群算法对多维组合优化问题的应用之一。

因此本文对应用蚁群算法来解决 QoS 组播路由研究有着积极的意义。蚁群算法的不断改进与成熟, 使其除了在解决 QoS 组播路由外, 还在其它组合优化问题中有着非常广阔的应用前景。

### 5.3.2 QoS 组播路由技术设计的目标

网络路由协议设计的目标是路由算法必须具有选择最佳路由的能力, 简单、低耗而效率高, 健壮且稳定、快速收敛、具有灵活性等特性。

IP 网络中对 QoS 组播路由算法的评价指标主要包括:

- (1) 正确性。正确性主要评价 QoS 组播路由算法正确与否, 判断所获得的组播树是否满足给定的约束条件, 即得到的最终解是否是一个可行解。
- (2) 路由请求平均成功率。路由请求平均成功率用于分析 QoS 组播路由算法的完备性, 即当网络存在满足需求的组播树时, 路由算法能否找到一个可行解。
- (3) 算法运行时间。算法运行时间是从仿真的角度评价 QoS 组播路由算法的计算复杂度。
- (4) QoS 度量值结果。QoS 度量值结果是指求得的可行组播树的 QoS 度量参数的值, 具体包括组播树的延时、代价、延时抖动、带宽等。这些 QoS 度量的具体定义前文已有介绍。QoS 度量值的结果是评价一个 QoS 组播

路由算法的重要性能指标。

- (5) 收敛性。必须在计算路由时迅速收敛，在较短的时间内查找路由，提供准确的路由信息。在一些使用智能优化算法(如蚁群算法、遗传算法、模拟退火算法、禁忌搜索算法)解决 NP 完全的 QoS 组播路由问题时，由于求出全局最优解的时间复杂度很大，一般通过限定迭代次数来求得一个性能较优秀的次优解。因此，算法的收敛性是一个重要评价指标。可通过平均迭代次数和最大迭代次数来反映一个 QoS 组播路由算法的收敛性能。平均迭代次数是指算法收敛到最优解(或次优解)时的平均迭代次数。最大迭代次数是指对一个指定的最大迭代次数 *Iteration*，算法迭代到第 *Iteration* 次时终止，此时求得的可行解的 QoS 度量的结果。
- (6) 避免无穷计算。算法在计算路由过程中不能出现无穷计算。
- (7) 控制信息开销小。由于计算和维护路由的过程中，节点之间要相互传递控制信息，消耗了有限的带宽资源，所以必须尽量减少这些控制信息的传递。
- (8) 可扩展性强。协议能够适应网络规模的增长。

## 5.4 本章小结

本章首先描述 QoS 及组播路由的概念，分析了他们各自的优缺点。接着详细介绍了对 QoS 度量的定义和其网络模型，给出组播树的代价、延时、延时约束的定义。然后介绍了组播分布树类型，并列举了一些经典的 QoS 组播路由算法。最后对 QoS 组播路由技术进行了分析，提出了 QoS 组播路由技术面临的挑战。本章内容对本文后续的 QoS 组播路由算法的研究工作奠定了坚实的理论基础。

## 第六章 基于自适应蚁群算法的 QoS 组播路由算法及其应用

蚁群算法作为解决优化组合的算法，可以应用于解决 QoS 组播路由问题。根据优化后的蚁群算法的基本原理，提出了一种新的 QoS 组播路由算法，称为基于自适应蚁群算法的 QoS 组播路由算法(QoS Multicast Routing Algorithm Based on Adaptive Ant Colony, MRABAAC)。Wang Z 等学者已经证明了包含两个及以上约束条件的 QoS 网络路由是一个 NP 完全问题<sup>[47]</sup>。蚁群算法作为解决 QoS 组播路由问题的算法之一，已经有许多文献给出了各自具体的解决算法<sup>[40][42]</sup>，在如何构造组播树上各有特色。

### 6.1 QoS 组播路由数学模型

#### 6.1.1 QoS 组播路由问题数学描述

在第五章中，对实际的网络拓扑结构进行了抽象，得出 QoS 的网络模型。现在以 QoS 的网络模型为基础，提出了 QoS 组播路由问题数学描述。

对于分组交换网络可用  $G=(V,E)$  表示，其中  $V$  表示为网络中节点的集合， $E$  表示网络中双向链路的集合， $R_+$  表示正实数集， $R^+$  表示非负实数集。对于任一网络节点  $n \in V$ ，定义四种属性，分别为延时函数  $\text{delay}(n):V \rightarrow R_+$ ，延时抖动函数  $\text{delay\_jitter}(n):V \rightarrow R^+$ ，包丢失率函数  $\text{packet\_loss}(n):V \rightarrow R^+$  和费用函数  $\text{cost}(n):V \rightarrow R_+$ 。对于任意的一条链路  $e \in E$ ，定义四种属性：延时函数  $\text{delay}(e):E \rightarrow R_+$ ，延时抖动函数  $\text{delay\_jitter}(e):E \rightarrow R^+$ ，带宽函数  $\text{bandwidth}(e):E \rightarrow R_+$ ，费用函数  $\text{cost}(e):E \rightarrow R^+$ 。

对于给定的源点  $s \in V$ ，终点集合  $M$  和源点  $s$  组成的组播树  $T(s,M)$  存在下列关系：

$$\text{delay}(T(s,M)) = \sum_{e \in T(s,M)} \text{delay}(e) + \sum_{n \in T(s,M)} \text{delay}(n) \quad (6.1)$$

$$\text{delay\_jitter}(T(s,M)) = \sum_{e \in T(s,M)} \text{delay\_jitter}(e) + \sum_{n \in T(s,M)} \text{delay\_jitter}(n) \quad (6.2)$$

$$\text{bandwidth}(T(s,M)) = \min\{\text{bandwidth}(e), e \in T(s,M)\} \quad (6.3)$$

$$\text{cost}(T(s,M)) = \sum_{e \in T(s,M)} \text{cost}(e) + \sum_{n \in T(s,M)} \text{cost}(n) \quad (6.4)$$

$$\text{packet\_loss}(T(s, M)) = 1 - \prod_{e \in T(s, M)} (1 - \text{packet\_loss}(e)) \quad (6.5)$$

QoS 组播路由问题就是在网络  $N(V, E)$  中，给定源节点  $s$ ，终节点集合  $M$ ，寻找一棵组播树  $T(s, M)$  满足以下约束条件：

- (1) 延时约束：  $\text{delay}(T(s, M)) \leq D$ ；
- (2) 带宽约束：  $\text{bandwidth}(T(s, M)) \geq B$
- (3) 延时抖动约束：  $\text{delay\_jitter}(T(s, M)) \leq DJ$
- (4) 包丢失率约束：  $\text{packet\_loss}(T(s, M)) \leq PL$
- (5) 费用约束，在满足前面 4 个约束的条件下，  $\text{cost}(T(s, M))$  最小。

其中， $B$ ， $D$ ， $DJ$  和  $PL$  分别为网络中的带宽、延时、延时抖动和包丢失率等的约束限制。

### 6.1.2 实现过程

组播作为一种通过复制分发数据的手段，能将数据从一个节点发送到多个节点上，而又可以尽可能节约带宽、降低延时和延时抖动，节约费用等。构造组播树是解决组播路由问题的有效方法。前面提到的有源树和 Steiner 树都是组播树。在 QoS 组播中，Steiner 树<sup>[43-44]</sup>的效果比有源树要更好一些，它是考虑全局的代价，而不是源点到各个目标点之间的代价最小。

定义 1：Steiner 树问题

给定带权简单无向图  $G=(V, E)$ ，其中  $V$ ， $E$  分别是网络节点和链路集合。对每条链路  $(u, v) \in E$ ，定义一个正实数的代价函数  $\text{cost}(e): E \rightarrow R^+$ ，表示链路上网络资源的利用情况。给定组播源节点  $s$ ，一个组播目的节点集  $M_c \in V - \{s\}$ ，设集合  $D_M = M \cup \{s\}$  表示参与组播会话的节点集，即由源节点和目的节点组成的集合，则 Steiner 树问题就是从图  $G$  中寻找一棵覆盖  $D_M$  中所有节点的最小生成树  $T$ ，使得树的代价  $\text{cost}(T) = \sum_{e \in E_T} \text{cost}(e)$  最小，该最小生成树也称为 Steiner 树，记为  $T_s(V_T, E_T)$ ，其中  $D_M \subseteq V_T \subseteq V$ ， $E_T \subseteq E$ 。

最小生成树和最短路径问题都是 Steiner 树问题的特例。当  $D_M = V$  时，Steiner 树问题就转化为求图的最小生成树问题；当  $|D_M|=2$  时，Steiner 树问题就又转化为两点间最短路径问题。

前面已提到过，带约束条件的 Steiner 树问题称为带约束 Steiner 树问题。在应用到 QoS 组播路由算法中，这些约束条件包括传输延时、丢失率、带宽要求、延时抖动、吞吐量等。最小生成树和最短路径问题都存在多项式时间内的最优解法。



因此，求解该问题的一个合理目标就是寻找该问题的一个启发式算法，使得能在一个低阶多项式时间内得到一个“接近”最优值的解。蚁群算法作为一种启发式算法，是解决带约束 Steiner 树问题的途径之一。

## 6.2 QoS 组播路由算法

### 6.2.1 算法设计的说明

由于 QoS 网络模型与 TSP 存在一定的差别，如果要蚁群算法应用到求解 QoS 组播问题，必须先对参数的转换做出一些说明，然后再阐述基于自适应蚁群算法的 QoS 组播路由算法。

$\eta_{ij}$  表示由节点  $i$  转移到节点  $j$  的启发程度，在解决 TSP 问题时， $\eta_{ij} = 1/d_{ij}$ ，在 QoS 网络模型中则变化为  $\eta_{ij} = 1/\text{cost}(i, j)$ ， $\text{cost}(i, j)$  表示节点  $i, j$  之间的费用大小，用来代替 TSP 中节点之间的距离。

在 QoS 网络模型的拓扑结构中，在各个节点  $(d, dj, pl, c)$  和每条存在边  $(d, dj, b, c)$  的描述中  $d, dj, pl, b, c$  分别表示延时、延时抖动、丢包率、带宽、代价。由于是计算带多约束条件的 QoS 组播路由，所以在寻找最优路径的过程中，必须判断该路径是否符合多约束条件，这是与 TSP 中每个城市之间都存在路径的思想是不同的。

用蚁群算法来寻找带约束条件的 Steiner 树，蚂蚁形成树型结构的方式与在 TSP 中蚂蚁形成一条路径的方式是截然不同的。求解满足多约束条件由单个源点到多个目标点或由多个源点到多个目标点的组播树就转换为求解包含所有源节点和目标节点的带约束条件的 Steiner 树。在计算的过程中，源节点和目的节点中任何一点都可以作为出发源点，其他点作为目标点，源节点和目的节点的集合称为组播节点集。

### 6.2.2 算法的描述

- (1) 初始化各类参数。网络模型节点数目  $n$ ，信息素强度初始值  $Q = (nL_m)^{-1}$ ，信息启发因子  $\alpha$ ，期望启发式因子  $\beta$ ，全局信息素挥发参数的初始值  $\rho$  以及限制条件  $\rho_{\min}$ ，局部信息素挥发参数  $\alpha_1$ ，蚂蚁数目采用  $m \approx n$ ，状态转移规则中的  $q_0$ 。给出各个节点、边、约束条件等参数，最大的循环次数  $NcMax$ ，组播节点集  $S$ ；

- (2) 精简网络。根据约束条件精简网络拓扑中不符合要求的链路或节点；
- (3)  $N_c = N_c + 1$ ，初始化禁忌表  $\text{tabu}$ ；
- (4) 对于蚂蚁  $k$ ， $\text{taub}_k$  表示蚂蚁已经走过节点的集合， $s$  表示  $\text{taub}_k$  的列数。链路集合  $E_k^s$  表示蚂蚁  $k$  从组播节点集(源节点和目的节点的集合)中  $s$  节点出发后遍历整个网络链路的集合，链路集合  $E_k$  表示各个  $E_k^s$  并集；
- (5) 蚂蚁  $k$  在组播节点集(源节点和目的节点的集合)中选择一个未被搜索的节点出发，称为节点  $i$ ，根据状态转移规则，从  $V_i - \text{tabu}_k$  中选择一个节点下一步所走的相对最优路径  $(i, j)$ ，其中  $V_i$  表示蚂蚁  $k$  在节点  $i$  上下一步允许路径，并判断从组播源点  $s$  到该点所经过的路径和节点是否满足约束条件延时、延时抖动、包丢失率。如果满足以上三个条件，将相对最优路径  $(i, j)$  加入到  $E_k^i$  中，否则继续从  $V_i - \text{tabu}_k$  中选择其他满足条件的节点，直至选择的节点为已经走过的节点。使用局部更新规则公式更新相应路径上的信息素强度，并将节点  $j$  加入到禁忌表  $\text{tabu}_k$ ；
- (6) 如果蚂蚁  $k$  已经从组播节点集的每个节点  $s$  ( $s=1,2,\dots,S$ )出发，完成了路径搜索，则输出链路集合  $E_k$ ，它各个  $E_k^s$  的并集。否则，转到第(5)步；
- (7) 对蚂蚁  $k$  寻找链路集合  $E_k$  所构成的组播树进行组播树枝剪。枝剪规则为：如果组播树的每条路径的叶子节点不是目标节点，则从  $\text{tabu}_k$  中删除该节点，并从  $E_k$  中删除与该节点相连的链路；
- (8) 对组播树集合  $T$  中各个组播树路径集合  $E_k$  和顶点集合  $\text{tabu}_k$ ，计算对应的目标函数值  $L_k$ ，比较后获得当前迭代最优组播树和全局最优组播树；
- (9) 是否所有蚂蚁都完成一次路径搜索，没有则转第(4)步；
- (10) 若  $N_c < N_{c\text{Max}}$  则转到第(3)步；

输出满足约束条件的最优目标函数值及所对应的组播树(即为最优链路)；若整个蚁群的所有路径都不满足约束条件，则所求链路不存在。

### 6.2.3 算法特点分析

算法具有如下特点：

- (1) 该算法由于有约束条件的存在，并不能遍历网络模型中的每一个节点。
- (2) 与蚁群算法不同的是，并非蚂蚁完成一次全局的遍历活动后进行全局更新，而是该蚂蚁要从组播节点集合中各个节点均出发，各自遍历一次后才进行全局更新。

- (3) 形成的多条链路的集合并非就是组播树，因为叶子节点可能并非目的节点，所以对组播树进行裁剪，构成包含组播节点集合的最小生成树。
- (4) 满足约束条件的组播树不一定存在，则降低约束条件标准重新计算组播树。

## 6.3 仿真实验结果及比较分析

### 6.3.1 仿真环境和参数设置

为了方便与本文所列的文献[7]，文献[8]做比较，这里选择的开发的软件环境与它们一致，实验的软件环境：操作系统为 Windows XP，开发工具为 VC++，开发语言 C++。本文实验的硬件环境：CPU 为 AMD 2800++，内存 512M。另外，网络模型与它们一致，如图 6.1 所示。

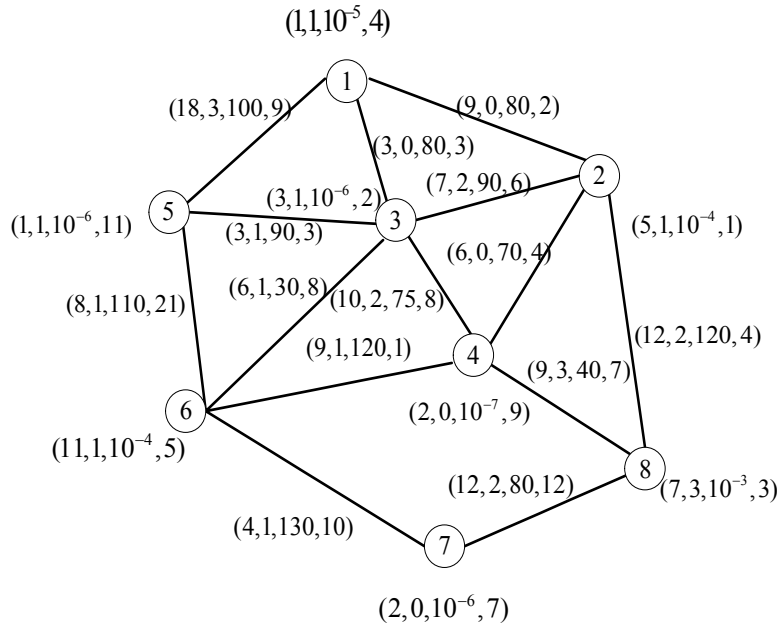


图 6.1 8 个网络节点网络结构模型

该网络节点模型是模拟真实的网络环境，并设置各个节点之间和各边之间的各类参数。在各节点的  $(d, dj, pl, c)$  和每条存在边的  $(d, dj, b, c)$  的描述中， $d$ 、 $dj$ 、 $pl$ 、 $b$ 、 $c$  分别表示延时、延时抖动、丢包率、带宽、代价。利用优化后的自适应蚁群优化算法，选择的参数  $\alpha=2$ ， $\beta=4.5$ ， $\rho=0.1$ ， $q_0=0.05$ ， $\delta=0.1$ ， $\alpha_1=0.1$ ，蚂蚁的数目  $m$  为 8，迭代循环次数  $NcMax=20$ 。其中节点 1 为源节点，节点 2、4、5、7 为目的节点。约束条件为  $B=70$ ， $D=46$ ， $DJ=8$ ， $PL=0.01$ 。根据网络的带宽

条件去掉节点(3, 6)之间的边和(4, 8)之间的边，如图 6.2 所示。

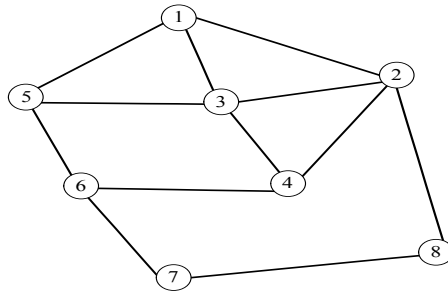


图 6.2 应用了带宽约束条件

### 6.3.2 实验结果及分析

上述仿真场景和参数设置，得出的结果图如图 6.3。为了和其他文献做比较，将他们的结果图也引用进来，在结果分析中做比较分析使用。该文算法生成的结果图如图 6.3，它是满足各类约束条件而代价最小的一棵组播树，从源点 1 出发，到达目点节点 2、4、5、7。由该图可知算法是正确、有效的，能够获得解决 QoS 组播路由的结果。

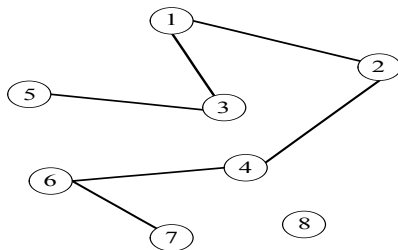


图 6.3 结果图

图 6.4 是文献[8]针对上述网络模型的遗传算法的曲线图，未优化的蚁群算法和优化的自适应蚁群算法如图 6.5 和图 6.7。每个图共三条曲线，分别反映了延时、延时抖动、代价的值及它们的波动情况。

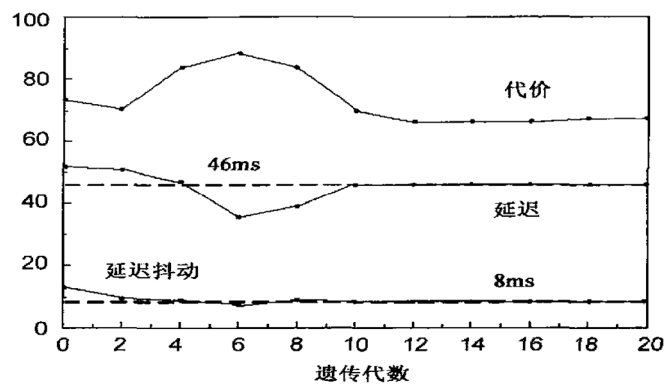


图 6.4 文献[8]的曲线图

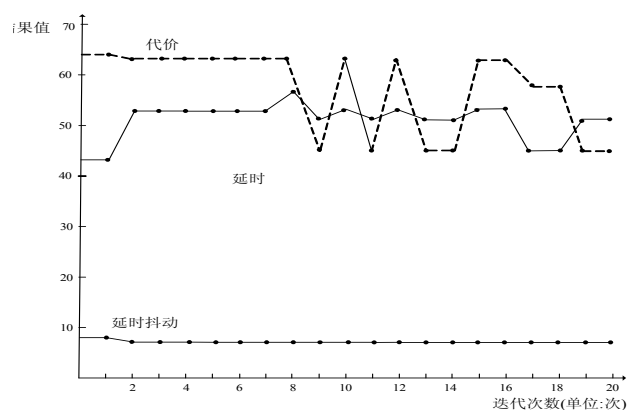


图 6.5 蚁群算法的曲线图

首先对比遗传算法的曲线图 6.4 和蚁群算法的曲线图 6.5，从代价的值上看，蚁群算法要优于遗传算法，但是蚁群算法的曲线波动较大，说明在搜索上范围较广；而和遗传算法相比，蚁群算法的波峰和波谷的值几乎都低于遗传算法生成图的曲线的值。在延时曲线上，遗传算法比蚁群算法平稳一些，延时的值几乎都差不多，蚁群算法的延时曲线波动幅度大是由代价曲线的变动造成的。在延时抖动曲线上，遗传算法和蚁群算法的曲线波动幅度都比较平稳，蚁群算法在延时抖动上的值明显小于遗传算法的值。如果仅从用两种算法来解决上述同一个网络模型的实验结果来看，蚁群算法作为一种新型的启发式算法，在解决组播路由的问题中性能要优于遗传算法。

在延时、延时抖动、带宽、丢包率和最小代价等多约束条件下求解的延时、延时抖动、代价曲线图，图 6.6 是文献[7]对蚁群算法优化时也是针对上述网络模型的遗传算法的曲线图，优化的自适应蚁群算法如图 6.7 所示。

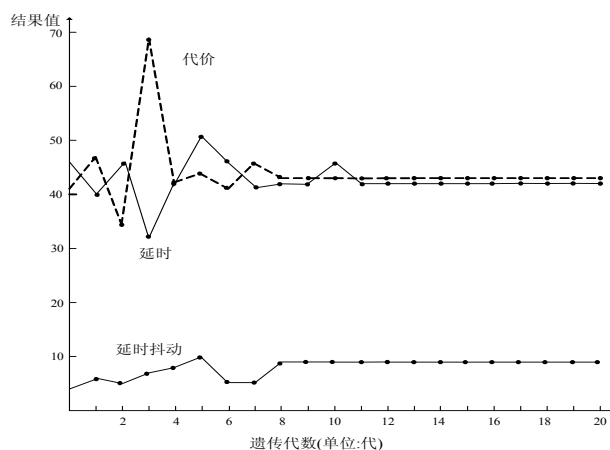


图 6.6 文献[7]

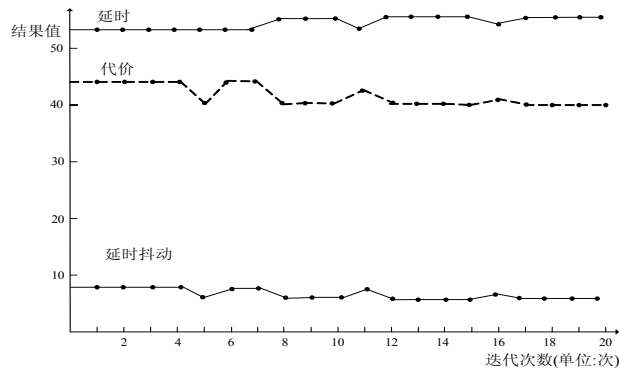


图 6.7 优化后的自适应蚁群算法的曲线图

对比图 6.5、图 6.7。图 6.7 在使用优化的蚁群算法后，代价的曲线波动比图 6.5 的曲线波动平稳一些，仅在靠近开始的地方有一个陡的上升的值。在延时曲线上，图 6.7 的曲线波动比图 6.5 稍微要大一些，而在延时抖动曲线上效果却要好一些。

重点分析图 6.6 和图 6.7 两个图的结果。从代价曲线上看，本文算法的图 6.7 的曲线波动幅度比图 6.6 的较小，而且从值上来说，也要小一些，说明在满足多个约束条件的情况下组播树的效果要好于前者。在代价曲线上，图 6.6 较早的时候就平稳了，而本文算法的曲线有小的波动，这说明本文算法引进全局最优系数  $\delta$  后扩大全局的搜索能力，防止陷入局部最优。在延时曲线上，本文算法的曲线的值稍微大一些，值稍微差一点。在延时抖动曲线上，本文算法的曲线的值趋于平稳，比图 6.6 的延时抖动曲线效果要好一些。

作为解决 QoS 组播路由的算法，能找到满足多个约束的条件下而代价最小的解的结果是最有效的。在上述图中，本文算法的代价曲线图是相对平稳而值是最小的，本文算法的在解决上述问题的效果是优于其他两位的。

综上所述，本文对蚁群算法的自适应优化在应用解决 QoS 组播路由问题中是正确、有效的，有其应用价值的。

## 6.4 本章小结

本章主要提出了基于蚁群算法的 QoS 组播路由算法，并进行了仿真实验，对仿真实验结果进行了分析，并通过实验给出研究结果。首先是描述了 QoS 组播树的构成，然后提出了 QoS 组播路由算法的具体步骤，最后给出了 QoS 组播模型并进行了仿真实验。

## 第七章 结束语

蚁群算法是一种基于蚂蚁种群的新颖的自激励启发式算法，能够解决一系列复杂的组合优化问题。论文主要研究对蚁群算法的优化，然后应用基于优化的自适应蚁群算法来解决具有 NP 难度的 QoS 组播路由问题。

论文的主要内容包括以下几个方面：

- (1) 探讨了研究 QoS 组播路由的意义，分析了国内及国外蚁群算法的研究现状。
- (2) 对于蚁群算法的原理，着重介绍了其特点和三个规则，这是蚁群算法的重点，也是它的基础。这些基础知识为后面在对蚁群算法进行参数的分析时给出了有力的支持。除此之外，也分析了蚁群算法的优缺点，为后面的优化提供理论上的依据。
- (3) 在对蚁群算法参数的选择的各项实验中，都采取多次实验取平均值的方法，尽量减少因为概率性而带来的不稳定因素。分析的参数包括信息素强度范围，蚂蚁数目与节点的关系，启发因子和信息素挥发因子，状态转移参数的范围等。不仅完成了各项实验，而且在实验后都进行了合理性分析，用数据来论证。最后，提出了蚁群算法的参数最优组合原则。
- (4) 对蚁群算法进行了优化，设计了新的蚂蚁转移规则和信息素更新机制。在信息素强度更新的策略中新引入了全局最优系数，通过判断全局最优系数来动态地更新全局信息素挥发系数和状态转移参数。实验结果证明优化后的算法能够确保自适应地提高全局搜索能力，同时也避免陷入局部最优解。在节点规模增长时，优化后的算法仍然能保持良好的性能。
- (5) 分析 QoS 组播路由技术面临的挑战，介绍了对目前常见的一些算法，分析它们的优缺点。详细地研究了 QoS 网络路由模型来进行数学模型的抽取，对如何应用蚁群算法解决 QoS 组播路由问题提出了详细的分析和设计。
- (6) 在解决 QoS 组播路由问题时，介绍了 Steiner 树，它是研究 QoS 组播路由算法的重点之一。如何应用优化的自适应蚁群优化算法构造解决 QoS 组播路由的带约束的 Steiner 树，是论文在实际应用与研究的一个综合体。理论分析和仿真实验结果证明，论文研究工作是成功的。优化后的蚁群算法

是有效的，从参数选择原则和自适应优化到应用解决 QoS 组播路由问题，各项实验结果表明达到了预期的效果。

由于水平有限，论文的研究工作仍然存在一些需要进一步完善的地方，下一步需要从以下几个方面来完成：

- (1) 优化的蚁群算法在收敛性上需要通过数学公式来进行验证。实验能说明一部分问题，但是从数学角度来验证可以体现出算法的严谨性。
- (2) 对蚁群算法的参数选择原则的研究尽量选取更贴近实际问题，对不同类的问题提出相应的选择原则。
- (3) 解决 QoS 组播路由问题，扩大节点的数目，进一步对论文提出的 QoS 组播路由算法进行研究 with 优化。

蚁群算法作为一种根据仿生学抽象出来的具有高度创新性的启发式算法，已经被证明在实际工程中的应用价值，它的发展与前景都是很好的<sup>[51]</sup>。在组合优化领域内，它是非常具有潜力的算法之一。



## 参考文献

- [1] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications[J]. IEEE J. Select. Areas Commun., vol.14, pp.1288–1234, Sept., 1996
- [2] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A Framework for QoS-based Routing in the Internet[S]. RFC no. 2386, Internet RFC, Aug. 1998
- [3] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: 1996, 26(1): 29~41
- [4] Marco Dorigo, Mauro Birattari, and Thomas Stiitzle. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique[J]. IEEE Computational Intelligence Magazine, 2006,11: 28~39
- [5] Dorigo M, Luca Maria Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53~66
- [6] Bin Wang and Jennifer C. Hou. Multicast Routing and Its QoS Extension: Problems, Algorithms, and Protocols[J]. IEEE Network, 2000,1: 22~36
- [7] 孙力娟, 王汝传. 基于蚁群算法和遗传算法融合的 QoS 组播路由问题求解[J]. 电子学报, 2006, 8(34): 1391~1395
- [8] 王征应, 石冰心. 基于启发式遗传算法 QoS 组播路由问题求解[J]. 计算机学报, 2001, 24 (1) : 55~61
- [9] Bonabeau E, Dorigo M, Theraulaz G. Inspiration of optimization from social insect behavior[J]. Nature, 2000, 406(6): 39~42
- [10] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005
- [11] 李士勇. 蚁群算法及其应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2004
- [12] 王晓东. 算法设计与分析[M]. 北京: 清华大学出版社, 2003
- [13] Merkle D, Middendorf M. Modeling the dynamics of ant colony optimization[J]. Evolutionary Computation, 2002, 10(3): 235~262
- [14] Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem[J]. Informs journal on Computing, 1999, 11(4): 358~369

- [15] Colloha A., Dorigo M., Maniezzo V. Ant colony system for job-shop scheduling[J]. JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science, 1994, 34(1): 39~53
- [16] Bullnheimer B., Hartl R.F., Strauss C. An improved ant system algorithm for the vehicle routing problem[J]. Annals of operation sresearch. 1999, 89(13): 319~328
- [17] Gambardella., L.M., Dorigo M. HAS-SOP: An hybrid Ant system for the sequential ordering problem[J]. Technique Rport, No.IDSIA-11-97, IDSIA-A, Lugano, Switzerland, 1997
- [18] Costa D., Hertz A. Ants can colour graphs[J]. Jounal of the Operational Research Society, 1997, 48(3): 295~305
- [19] Maniezzo V., Carbonaro A. An ANTS heuristic for the frequency assignment problem[J]. Future Generation Computer Systems, 2000, 16(8): 927~935
- [20] 王真真. 基于 QoS 的组播路由算法的研究: [硕士学位论文], 南京: 南京理工大学, 2008
- [21] 杨亚楠. 蚁群算法参数优化及其应用: [硕士学位论文], 南京: 南京理工大学, 2008
- [22] 刘彦鹏. 蚁群优化算法的理论研究及其应用: [博士学位论文], 杭州: 浙江大学, 2008
- [23] 王晰. 基于 QOS 约束的组播路由算法研究: [博士学位论文], 南京: 浙南京理工大学, 2004
- [24] 王志广. 基于 QoS 约束组播路由算法及协议研究: [硕士学位论文], 石家庄: 河北工业大学, 2006
- [25] 刘莹, 徐恪. Internet 组播体系结构[M]. 北京: 科学出版社, 2008
- [26] 姜圳. 基于 QoS 的组播路由关键技术研究: [博士学位论文]. 哈尔滨: 哈尔滨理工大学, 2004
- [27] 刘利强, 戴运桃, 王丽华. 蚁群算法参数优化[J]. 计算机工程, 2008, 34(11): 208~210
- [28] 顾军华, 侯向丹, 宋洁, 李琳. 基于蚂蚁算法的组播路由问题求解[J]. 河北工业大学学报, 2002, 31(4): 19~23
- [29] 杨文国, 郭田德. 求解最小 Steiner 树的蚁群优化算法及其收敛性[J]. 应用数学学报, 2006, 29(2): 352~361
- [30] 陈佑建. 蚁群算法的研究及在网络路由优化上的应用: [硕士学位论文], 南京: 河海大学, 2005

- [31] Kou L, Markowsky Q., Berman. A Fast Algorithm for Steiner Trees[J]. Acta Information, 1981, 15(2): 141~145
- [32] Qing Zhu, Mehrdad Parsa, J. J. Garcia-Luna-Aceves. A Source-Based Algorithm for Delay-Constrained Minimum-Cost Multicasting[J]. In Proceedings of IEEE INFOCOM'95[C]. Boston, MA, 1995: 377-385
- [33] D.Waitzman, C.Partridge, S.E. Deering. Distance vector multicast routing protocol[S]. RFC1075, Nov.1998
- [34] Moy J. Multicast Extensions to OSPF[S]. RFC 1584, Mar.1994
- [35] D. Estrin. Protocol Independent Multicast(PIM): Dense Mode Protocol Specification [S]. RFC 2362 June 1998
- [36] D. Estrin, Farinacci, et al. Protocol Independent Multicast-Sparse Mode(PIM-SM): Protocol Specification[S]. RFC 2117 June 1997
- [37] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing[S]. RFC 2189 September 1997
- [38] V. P.Kompella , J. C. Pasquale, Multicast Routing for Multimedia Communication[J]. IEE/ACM Transactions on Networking, 1993, 1(3): 286-292
- [39] Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications[J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228~1234
- [40] 王颖,谢剑英. 一种自适应蚁群算法及其仿真研究[J]. 系统仿真学报, 2002, 14(1): 31-33
- [41] 覃刚力,杨家本. 自适应调整信息素的蚁群算法[J]. 信息与控制, 2002, 31(3): 31~33
- [42] 李开荣, 陈宏建, 陈绫. 一种动态自适应蚁群算法[J]. 计算机工程与应用, 2004, 29: 149~152
- [43] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240~1245
- [44] 朱庆保, 杨志军. 基于变异和动态信息素更新的蚁群优化算法[J]. 软件学报, 2004, 14(2): 185~192
- [45] 高尚, 汤可宗, 杨静宇. 一种新的基于混合蚁群算法的聚类方法[J]. 微电子学与计算机, 2006, 23(12): 38~40
- [46] (意)多里戈(Marco Dorigo), (德)Thomas Stutzle 著, 张军 等译. 蚁群优化[M]. 北

京: 清华大学出版社, 2005

- [47] 吕勇. 蚁群优化算法及在网络路由中的应用研究: [博士学位论文], 杭州: 浙江大学, 2005
- [48] 黄河. 蚂蚁算法在 IP 路由中的应用: [硕士学位论文], 武汉: 武汉理工大学, 2007
- [49] 孙丽敏. 基于遗传算法的 QoS 多组播路由问题研究:[硕士学位论文], 西安理工大学, 2008
- [50] 刘仁宏. 一种改进的蚁群算法求解旅行商问题: [硕士学位论文], 济南: 山东大学, 2008
- [51] 段海滨, 王道波, 于秀芬. 蚁群算法的研究现状及其展望[J]. 中国工程科学, 2007, 9(2): 98~101

## 攻读硕士学位期间发表的论文

王庆美，邱会中. 基于 iSCSI 协议的 IP 存储原理以及在 LINUX 下的实现. 电子科技大学研究生学报,2008,12（59）:54-58

## 致 谢

首先我要衷心地感谢我的导师邱会中副教授。在我的研究生学习阶段，邱老师从生活及工作等方面给予了悉心的指导和无微不至的关怀，教给我研究问题和解决问题的方法和做人的准则。邱老师严谨的治学态度和诲人不倦的师长风范，是我终生的学习典范。在此，我向邱老师表示最诚挚的敬意和谢意。

我还要感谢 1001 教研室里的同学们，特别是周孙玉，欧黎源等，我们在一起进行了很多学术上的探讨，他们给予了我极大的帮助。感谢刘荣，熊爱金和赵怀玉等同学，在我们一起奋斗的日子里她们给了我很大的鼓励，以及论文撰写的指导。

其次，我要向我的父母家人表示由衷的感谢，正是由于他们的无私奉献和鼎力支持，我才能顺利的完成我的硕士学位，我对他们致以最衷心的感谢。

最后，衷心地感谢这次评阅论文的专家和学者们！



# 硕士学位论文

MASTER DISSERTATION