

南京航空航天大学金城学院

# 毕 业 设 计

题 目 基于蚁群算法的 TSP 问题研究

学生姓名

李 奇

学 号

2005021237

系 部

信息工程系

专 业

信息工程

班 级

20050212

指导教师

谭静 讲师

二〇〇九年六月

**南京航空航天大学金城学院**  
**本科毕业设计（论文）诚信承诺书**

本人郑重声明：所呈交的毕业设计（论文）（题目：基于蚁群算法的TSP问题研究）是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业设计（论文）中特别加以标注引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名：

2009 年 6 月 6 日

（学号）：2005021237



# 基于蚁群算法的 TSP 问题研究

## 摘 要

本文研究了基于蚁群算法解决 TSP 问题的原理，算法流程以及用 MATLAB 程序的仿真。论文首先简单回顾了蚁群算法的历史、发展以及应用，然后详细介绍了基本蚁群算法的原理，包括基本蚁群算法的行为描述和机制原理。其次从基本蚁群算法的系统学特征出发，讨论它具有分布式，自组织，正反馈等特征。接着引出了基本蚁群算法解决的 TSP 问题，先讨论了组合优化问题，然后从 TSP 问题的定义，实用价值，理论意义的角度对 TSP 问题进行阐述。并且重点运用 MATLAB 的仿真方法，实现了基于蚁群算法的仿真，给出了求解 TSP 问题的数学模型，实现步骤，描述了蚁群算法的优缺点。论文最后以 MATLAB 仿真实验为基础，对蚁群算法的主要参数进行了详细的讨论，并且给出了优化的参数选择，解决了算法中存在的不足。论文实现了基于蚁群算法对 TSP 问题的求解和仿真。

**关键字：** 蚁群算法，组合优化，信息素，TSP 问题



# TSP research based on ant colony algorithm

## Abstract

This paper researched the principle based on ant colony algorithm to solve TSP problem, the algorithm processes and procedures using MATLAB simulation. Paper first briefly reviewed the history of ant colony algorithm, development and application, and then described in detail the basic principle of ant colony algorithm, including the conduct of the basic ant colony algorithm and the mechanism described in principle. Second, the basic ant colony algorithm from the characteristics of the system starting to discuss it with the distributed, self-organization, characteristics of positive feedback. Then leads to the basic ant colony algorithm to solve the TSP problem, first discuss the combinatorial optimization problems, and then from the definition of TSP problem, practical value, the theoretical significance of the point of view on the issue of the TSP. Focus on the use of MATLAB and the simulation method, ant colony algorithm based on the realization of the simulation, are given for solving the mathematical model of the TSP problem, the realization of these steps, describing the advantages and disadvantages of the ant colony algorithm. Finally simulation results in MATLAB based on the main parameters of ant colony algorithm are discussed in detail, and optimized parameters are given options to solve the shortcomings of existing algorithms. Paper achieved on optimization and simulation based on Ant colony algorithm to solve the problem.

**Key Words:** ant colony algorithm; combinatorial optimization; pheromone; TSP



## 目 录

摘 要 .....	i
Abstract .....	ii
第一章 绪 论 .....	- 1 -
1.1 蚁群算法概况.....	- 1 -
1.2 论文的主要内容.....	- 2 -
第二章 基本蚁群算法简介 .....	- 3 -
2.1 基本蚁群算法的原理.....	- 3 -
2.1.1 蚁群行为描述 .....	- 3 -
2.1.2 基本蚁群算法的机制原理 .....	- 5 -
2.2 基本蚁群算法的系统学特征.....	- 6 -
2.2.1 分布式 .....	- 6 -
2.2.2 自组织 .....	- 7 -
2.2.3 正反馈 .....	- 7 -
第三章 组合优化以及TSP问题简介.....	- 9 -
3.1 组合优化简介.....	- 9 -
3.1.1 引言 .....	- 9 -
3.1.2 组合优化问题 .....	- 9 -
3.1.3 NP完全问题 .....	- 10 -
3.2 TSP问题简介 .....	- 10 -
3.2.1 TSP问题的定义.....	- 10 -
3.2.2 TSP的实用价值.....	- 11 -
3.2.3 TSP问题的理论意义.....	- 11 -
3.3 基本蚁群算法的数学模型.....	- 11 -
第四章 基本蚁群算法求解TSP .....	- 14 -
4.1 参数设置平台 .....	- 14 -



---

4.2 基本蚁群算法求解TSP的实现流程 .....	15 -
4.2.1 基本蚁群算法的实现步骤 .....	15 -
4.2.2 基本蚁群算法的结构流程 .....	15 -
4.3 基本蚁群算法的仿真结果 .....	17 -
第五章 蚁群算法的主要参数设置 .....	19 -
5.1 关键参数介绍 .....	19 -
5.2 不同参数设置的试验 .....	21 -
5.2.1 信息素挥发度的设置 .....	21 -
5.2.2 蚁群数量的设置 .....	21 -
5.2.3 启发式因子的设置 .....	22 -
5.3 不同参数设置的实验结果和结论 .....	22 -
5.3.1 信息素挥发度的选择设置 .....	22 -
5.3.2 蚁群数量的选择设置 .....	24 -
5.3.3 启发式因子的选择设置 .....	25 -
第六章 总结和展望 .....	27 -
6.1 工作总结 .....	27 -
6.2 技术展望 .....	27 -
参考文献 .....	28 -
致 谢 .....	29 -
附 录 .....	30 -
附录 1: 基于基本蚁群算法的TSP问题源码 .....	30 -



## 第一章 绪 论

蚂蚁是地球上最常见、数量最多的昆虫种类之一，常常成群结队地出现在人类的日常生活中。这些昆虫的群体生物智能特征，引起了一些学者的注意。意大利学者 M.Dorigo, V.Maniezzo 等人在观察蚂蚁的觅食习性时发现，蚂蚁总能找到巢穴与食物源之间的最短路径。经研究发现，蚂蚁的这种群体协作功能是通过一种遗留在其来往路径上的叫做信息素（Pheromone）的挥发性化学物质来进行通信和协调的。化学通信是蚂蚁采取的基本信息交流方式之一，在蚂蚁的生活习性中起着重要的作用。通过对蚂蚁觅食行为的研究，他们发现，整个蚁群就是通过这种信息素进行相互协作，形成正反馈，从而使多个路径上的蚂蚁都逐渐聚集到最短的那条路径上。

### 1.1 蚁群算法概况

M.Dorigo 等人于 1991 年首先提出了蚁群算法。其主要特点就是：通过正反馈、分布式协作来寻找最优路径。这是一种基于种群寻优的启发式搜索算法。它充分利用了生物蚁群能通过个体间简单的信息传递，搜索从蚁巢至食物间最短路径的集体寻优特征<sup>[1]</sup>，以及该过程与旅行商问题求解之间的相似性。得到了具有 NP 难度的旅行商问题的最优解答。同时，该算法还被用于求解 Job-Shop 调度问题、二次指派问题以及多维背包问题等，显示了其适用于组合优化类问题求解的优越特征。

蚁群算法之所以能引起相关领域研究者的注意，是因为这种求解模式能将问题求解的快速性、全局优化特征以及在有限时间内答案的合理性结合起来。其中，寻优的快速性是通过正反馈式的信息传递和积累来保证的。而算法的早熟性收敛又可以通过其分布式计算特征加以避免，同时，具有贪婪启发式搜索特征的蚁群系统又能在搜索过程的早期找到可以接受的问题解答。这种优越的问题分布式求解模式经过相关领域研究者的关注和努力，已经在最初的算法模型基础上得到了很大的改进和拓展。

以蚁群算法为代表的群智能已成为当今分布式人工智能研究的一个热点，许多源于蜂群和蚁群模型设计的算法已越来越多地被应用于企业的运转模式的研究<sup>[2]</sup>。美国五角大楼正在资助关于群智能系统的研究工作—群体战略（Swarm Strategy），它的一个实战用途是通过运用成群的空中无人驾驶飞行器和地面车辆来转移敌人的注意力，让自己的军队在敌人后方不被



察觉地安全进行。英国电信公司和美国世界通信公司以电子蚂蚁为基础，对新的电信网络管理方法进行了试验。国内，国家自然科学基金“十五”期间学科交叉类优先资助领域中的认知科学及其信息处理的研究内容中也明确列出了群智能领域的进化、自适应与现场认知主题。

多年来世界各地研究工作者对蚁群算法进行了精心研究和应用开发，该算法现已被大量应用于数据分析、机器人协作问题求解、电力、通信、水利、采矿、化工、建筑、交通<sup>[5]</sup>等领域。蚁群算法最初用于解决TSP问题，经过多年的发展，已经陆续渗透到其他领域中，如图着色问题、大规模集成电路设计、通讯网络中的路由问题以及负载平衡问题、车辆调度问题等。蚁群算法在若干领域已经获得成功的应用，其中最成功的是在组合优化问题中的应用。

## 1.2 论文的主要内容

因为蚁群算法有很多种，并且TSP问题是蚁群算法众多解决问题中的经典问题，所以具有很大的随意性。虽然蚁群算法的选择性很多，但是所有的算法都是基于最基本的蚁群算法，并且基本蚁群算法也可以很好的解决TSP问题，以及说明参数选择问题。所以本文以基本蚁群算法为基础，通过MATLAB仿真这个平台，重点讨论了基本蚁群算法如何解决TSP问题。

第一章主要介绍了蚁群算法的历史、发展、应用，对蚁群算法原理的核心——信息素作了简单的描述。

第二章对基本蚁群算法的原理进行了介绍，包括基本蚁群算法的行为描述，机制原理。同时探讨了基本蚁群算法的系统学特征。

第三章对组合优化以及TSP问题的定义，实用价值，理论意义进行了描述。并且介绍了基本蚁群算法求解TSP问题的数学模型。

第四章首先介绍了利用基本蚁群算法的参数仿真平台，及实现步骤和结构流程。并给出了用MATLAB实现的仿真结果。

第五章主要讨论了蚁群算法中主要参数对于蚁群算法的全局搜索能力以及收敛性的影响，并做了相应仿真实验加以验证。

第六章对蚁群算法的发展进行了展望，同时对本文进行了总结。





## 第二章 基本蚁群算法简介

### 2.1 基本蚁群算法的原理

#### 2.1.1 蚁群行为描述

根据仿生学家的长期研究发现：蚂蚁虽然没有视觉，但运动时会通过在路径上释放出一种特殊的分泌物——信息素来寻找路径。当它们碰到一个还没有走过的路口的时，就随机的挑选一条路径前行，同时释放出与路径长度有关的信息素。蚂蚁走的路径越长，则释放的信息量越小。当后来的蚂蚁再次碰到这个路口的时候，选择信息量较大路径的概率相对较大，这样便形成了一个正反馈机制。最优路径上的信息量越来越大，而其他路径上的信息量却会随着时间的流逝而逐渐消减，最终整个蚁群会找出最优路径。同时蚁群还能够适应环境的变化，当蚁群的运动路径上突然出现障碍物时，蚂蚁也能很快的重新找到最优路径。可见，在整个寻径过程中，虽然单只蚂蚁的选择能力有限，但是通过信息素的作用使整个蚁群行为找出最优路径。这里用人工蚂蚁觅食图来描述蚁群搜索原理。

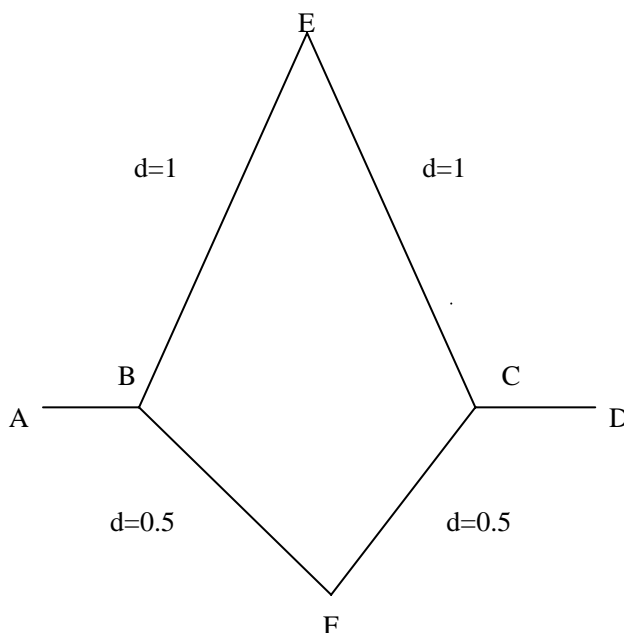


图 2.1 人工蚂蚁觅食模拟

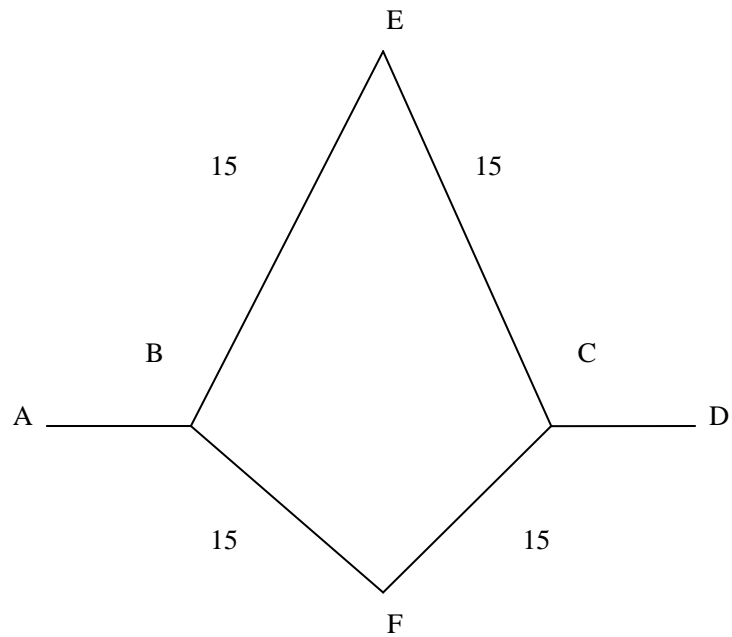


图 2.2 人工蚂蚁觅食模拟

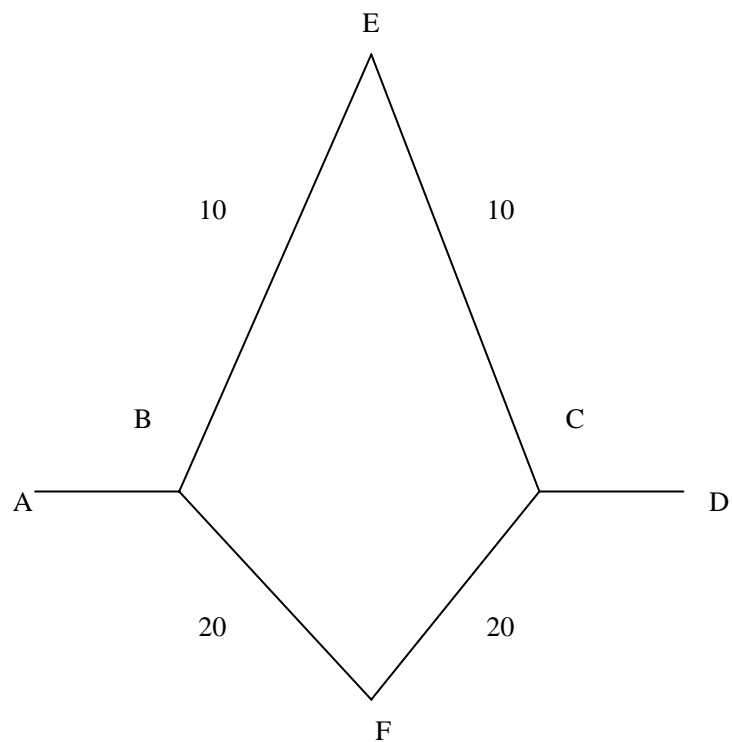


图 2.3 人工蚂蚁觅食模拟

图 2.1 中，设 A 点是蚁巢，D 点是食物源，EF 之间区域是障碍物。由于障碍物的存在，



蚂蚁只能经由 A 经 E 或 F 到达 D，或由 D 到达 A，各点之间的距离如图 2.1 所示。假设每个时间单位有 30 只蚂蚁由 A 到达 D 点，有 30 只蚂蚁由 D 到达 A 点，蚂蚁过后留下的信息量为 1。为了方便起见，设该物质停留时间为 1。在初始时刻，由于路径 BF、FC、BE、EC 上均无信息存在，位于 A 和 D 的蚂蚁可以随机选择路径，从统计学的角度可以认为蚂蚁以相同的概率选择 BE、FC、BE、EC，如图 2.2 所示。经过一个时间单位后，在路径 BFC 上的信息量是路径 BEC 上的信息量的 2 倍。又经过一段时间，将有 20 只蚂蚁由 B、F 和 C 点到达 D，如图 2.3 所示。随着时间的推移，蚂蚁将会以越来越大的概率选择路径 BFC，最终将会完全选择 BFC，从而找到由蚁巢到食物源的最短路径。

### 2.1.2 基本蚁群算法的机制原理

模拟蚂蚁群体觅食行为的蚁群算法是作为一种新的计算智能模式引入，该算法基于以下基本假设：

1. 蚂蚁之间通过信息素和环境进行通信。每只蚂蚁仅根据其周围的局部环境作出反应，也只对其周围的局部环境产生影响。
2. 蚂蚁对环境的反应由其内部模式决定。因为蚂蚁是基因生物，蚂蚁的行为实际上是其基因的适应性表现，即蚂蚁是反应型适应性主体。
3. 在个体水平上，每只蚂蚁仅根据环境做出独立选择；在群体水平上，单只蚂蚁的行为是随机的，但蚁群可通过自组织过程形成高度有序的群体行为。

由上述假设和分析可见，基本蚁群算法的寻优机制包含两个基本阶段：适应阶段和协作阶段。在适应阶段，各候选解根据积累的信息不断调整自身结构，路径上经过的蚂蚁越多，信息量越大，则该路径越容易被选择；时间越长，信息量会越小；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。

蚁群算法实际上是一类智能多主体系统，其自组织机制使得蚁群算法不需要对所求的问题的每一方面都有详尽的认识。自组织本质上体现了从无序到有序的动态演化，其逻辑结构如下图所示。

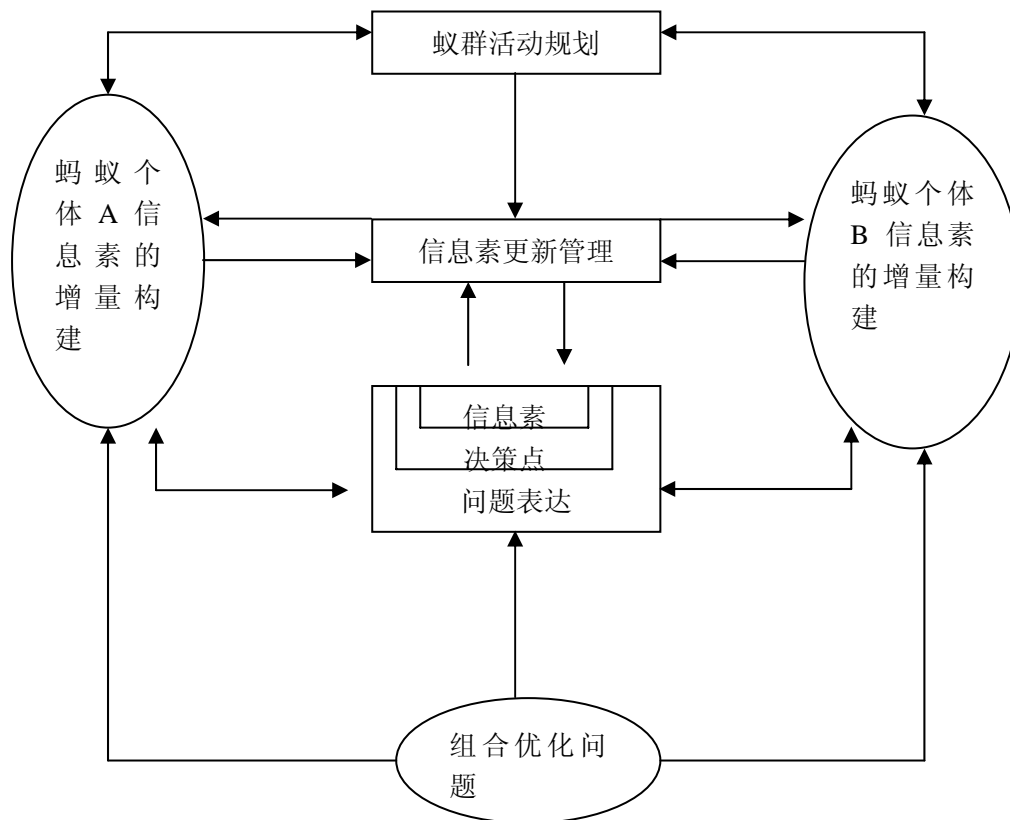


图 2.4 基本蚁群算法的逻辑结构

由图 2.4 可见，先将具体的组合优化问题表述成规范的格式，然后利用蚁群算法在“探索”和“利用”之间根据信息素这一反馈载体确定决策点，同时按照相应的信息素更新规则对每只蚂蚁个体的信息素进行增量构建，随后从整体角度规划出蚁群活动的行为方向，周而复始，即可求出组合优化的最优解。

## 2.2 基本蚁群算法的系统学特征

基本蚁群算法在系统学上表现以下三个特征：分布式，自组织，正反馈。三者表现出一个整体，相辅相成。

### 2.2.1 分布式

自然界中的真实蚁群行为也出现了分布式。当蚁群需要完成某项工作时，其中的许多蚂蚁都为共同目的进行着同样的工作，而最终任务的完成不会由于某些个体的缺陷而受到影响。

蚁群算法作为对蚁群觅食行为的抽象，也体现了群体行为的分布式特征。每只人工蚂蚁



在问题空间的多个点同时开始相互独立地构造问题解，而整个问题的求解不会因为某只人工蚂蚁无法成功获得解而受到影响。

具体到不同的优化问题而言，蚁群算法体现出的分布式特征就具有了更加现实的意义。因为所有的仿生优化算法均可看做按照一定规则的问题解空间搜索最优解的过程，所以搜索点的初始选取就直接关系到算法求解结果的优劣和算法寻优效率。当求解许多复杂问题时，从一点出发的搜索受到局部特征的限制，可得不到所求问题的满意解；而基本蚁群算法则可以看做是一个分布式的多智能系统，它在问题空间的多点同时独立地进行解搜索，不仅使得算法具有较强的全局搜索能力，也增加了算法的可靠性。

### 2.2.2 自组织

基本蚁群算法的另一个重要特征是自组织。自组织的概念是随着系统科学的发展而建立起来的。通常把系统论中的组织行为分为自组织和他组织两大类<sup>[4]</sup>。其根本区别在于组织力或组织指令是来自于系统内部还是来自于系统外部，前者称为自组织，而后者即为他组织。如果系统在获得时间的、空间的或者功能的结果过程中，没有外界的特定干预，则称为系统是自组织的。

从抽象意义上讲，自组织就是在没有外界作用下使得系统从无序到有序的进化过程。基本蚁群算法体现了这一过程，以蚁群优化为例，当算法开始的初期，单只人工蚂蚁无序地寻找解，经过一段时间的算法演化，人工蚂蚁越来越趋向于寻找到接近最优解的一些解，这恰恰体现了从无序到有序的自组织进化。

自组织大大增强了算法的鲁棒性<sup>[5]</sup>（在这里可以理解为算法抗干扰性，就是指不是针对具体问题算法也同样可以求解，极大的体现了蚁群算法的抗干扰能力），传统的算法都是针对某一具体问题而设计的，这往往建立在对该问题有了全面清晰认识的基础上，通常很难适应其他问题。而自组织的蚁群算法不需要对待求解问题的所有问题的所有方面都有所认识，因而比较容易应用到一类问题中。

### 2.2.3 正反馈

反馈是控制论中的重要概念，它代表信息输入对输出的反作用。在系统学上认为，反馈就是把系统现在的行为作为影响系统未来行为的原因。反馈分正反馈和负反馈两种，前者指的是以现在的行为去加强未来的行为，而后者则是以现在的行为去削弱未来的行为。

由自然界中真实蚂蚁的觅食行为可见，蚂蚁能够最终找到最短路径，直接依赖于最短路



径上信息素的堆积，而信息素的堆积却是一个正反馈的过程，对于基本蚁群算法而言，初始时在环境中存在完全相同的信息量，给系统一个微小的扰动，使得各个边上的信息量大小不同，蚂蚁构造的解就存在了优劣。算法采用的反馈方式是在较优解经过的路径留下更多的信息素，更多的信息素又吸引了更多的蚂蚁，这个正反馈的过程使得初始值不断地扩大，同时又引导整个系统向着最优解的方向进化。

单一的正反馈或者负反馈存在于线形系统之中，是无法实现系统的自我组织的。自组织系统通过正反馈和负反馈机制，它体现于蚁群算法在构造问题解的过程中用到了概率搜索技术，通过该技术增加了生成解的随机性。随机性的影响就在于接受了解在一定程度上的退化，另一方面又使得搜索范围得以在一段时间内保持足够大。这样正反馈缩小搜索范围，保证算法朝着最优解的方向进行；而负反馈保持搜索范围，避免算法过早收敛于不好的结果。恰恰是在正反馈和负反馈共同作用影响下，基本蚁群算法得以自组织地进化，从而得到问题在一定程度上的满意解。



### 第三章 组合优化以及 TSP 问题简介

#### 3.1 组合优化简介

##### 3.1.1 引言

伴随计算机技术的飞速发展，计算机正日益成为人们解决问题的不可或缺的重要工具。但在实践中人们发现，存在这样一类问题，运用精确的算法在多项式时间内无法找到问题的最优解，这类问题称为组合优化问题。这类问题通常随着问题规模的扩大，问题空间呈现组合爆炸特征，求解问题的空间、时间复杂度将呈指数级增长，使用常规的方法求解，在现有条件下是无法实现的，属于NP完全问题<sup>[6]</sup>。TSP问题就是其中最经典问题之一。因此利用仿生进化算法，在较短的时间内，求得问题的满意解，就成为研究的重点。

##### 3.1.2 组合优化问题

组合优化问题的目标是从组合问题的可行解中求出最优解。在现实世界里存在大量组合优化问题，其中许多问题如旅行商问题、图着色问题、分配问题、调度问题、布线问题及路由选择问题等<sup>[7]</sup>，至今没有找到有效地多项式算法，这些问题已被证明是NP完全问题。

优化问题有三个基本要素：变量、约束和目标函数。在求解过程中选定的基本参数称为变量，对变量取值的限制称为约束，表示可行方案衡量标准的函数称为目标函数。组合优化问题就是在给定的约束条件下，求目标函数最优值的问题。组合优化问题的一个实例可以表示为一个对偶  $(S, f)$ ，其中解空间  $S$  为可行解目标函数  $f$  是一个映射，定义为：

$$f : S \rightarrow R$$

求目标函数最小值问题称为最小化问题，记为：

$$\min f(i), i \in S$$

求目标函数最大值问题称为最大化问题，记为：

$$\max f(i), i \in S$$

显然，只要改变目标函数的符号，最小化问题与最大化问题就可以等价转换。使目标函数最优值的解称为最优解（整体最优解）。设  $(S, f)$  是组合优化问题的一个实例， $i_{opt} \in S$ ，





若:  $f(i_{apt}) \leq f(i)$  (对所有的  $i \in S$  成立), 称  $i_{apt}$  为最小化问题  $\min f(i), i \in S$  的最优解。

优化问题在工程中有着广泛的应用, 其涉及的问题是在一组限制条件下求解问题的最好(或最优)解的问题。按照人类认知问题的特点, 最优化问题很自然的被划分为两类: 一类是连续变量的问题, 另一类是离散变量的问题。对于离散变量问题来说, 一般是寻找离散事件的最优编排, 分组, 次序或筛选等。不难看出这些问题的解的个数是有限的。通常把这类问题称为组合最优化问题, 而研究用数学方法来求解这些问题就被称为组合最优化。

### 3.1.3 NP 完全问题

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<sup>[8]</sup>。

定义一: 对于判定问题 D, 存在一个多项式  $P(t)$ , 使得对其每一输入规模为  $n$  的实例, 可以在  $P(n)$  时间内求得最优解, 此类问题称为 P(Polynomial)问题。其它的则是 NP(Non — Polynomial)问题。

假如一个问题是 P 问题, 我们通常认为它是“简单的”, 而 NP 问题, 通常认为它是“复杂的”。NP 问题是指可在多项式时间里检验的问题, 至今尚未找到多项式时间算法。

定义二: 若问题  $L \in NP$ , 所有 NP 问题都可以经过多项式计算时间转化为 L, 则 L 称为 NP 完全问题。

如果 NP 完全类问题中有一个问题能用多项式确定性算法解决, 则其他所有的 NP 完全类问题都能用多项式确定性算法来解决, 很多问题被证明为 NP 完全问题, 如 TSP 问题, NP 完全问题也是检验仿生算法有效性和可靠性的有效平台。

## 3.2 TSP 问题简介

### 3.2.1 TSP 问题的定义

TSP问题的英文名(traveling salesman problem), 中文译为旅行商问题<sup>[9]</sup>。问题的定义很简单, 即为一个旅行商要走访N个城市, 每个城市必须经过一次且只能经过一次, 最后回到出发的城市就算是完成了一次旅行, 问如何能找到一条最短的路径。其相应的数学描述如下: 设有一个城市集合  $C = \{c_1, c_2, c_3, c_4, \dots, c_n\}$ , 其每对城市  $c_i, c_j \in C$  间的距离为  $d(c_i, c_j) \in Z^+$ 。

求一条经过C中每个城市正好一次的路径  $(c_{\pi(1)}, c_{\pi(2)}, c_{\pi(3)}, c_{\pi(4)}, \dots, c_{\pi(n)})$ , 使得

$$\sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(n)}, c_{\pi(1)}) \text{ 最小。}$$





### 3.2.2 TSP 的实用价值

TSP问题广泛的存在于许多领域，而且问题规模（城市的数目）都很大。比如说，电路板钻孔问题，X射线结晶学问题，VLSL（超大规模集成电路）制造问题等等<sup>[10]</sup>。总之，凡是抽象成为遍历全部城市一次且仅一次，求代价最小的路径的问题，都可以当作TSP问题来解决。

### 3.2.3 TSP 问题的理论意义

同实用价值相比，TSP问题的理论意义更加重大。事实上，该问题是作为所有组合优化问题的范例而存在的<sup>[11]</sup>。它已经成为并将继续成为测试新算法的标准问题。这是因为，TSP问题展示了组合优化的所有方面。它从概念上来讲非常简单，但是其求解的难度是很大的。如果针对TSP问题提出的某种算法能够取得比较好的实算效果，那么对其进行修改，就可以应用于其它类型的组合优化问题并取得良好的效果。基于上述原因，该问题吸引了许多不同领域的研究者。

## 3.3 基本蚁群算法的数学模型

设  $\tau_{ij}(t)$  为  $t$  时刻路径  $(i,j)$  上的信息量， $n$  表示 TSP 规模， $m$  为蚁群中蚂蚁的总数目。

在初始时刻各条路径上的信息量相等，并设  $\tau_{ij}(0) = \text{const}$ 。

蚂蚁  $k$  ( $k=1, 2, \dots, m$ ) 在运动过程中，根据各条路径上的信息量决定其转移方向，这里用禁忌表<sup>[12]</sup>  $\text{tabu}_k$  ( $k=1, 2, \dots, m$ ) 来记录蚂蚁  $k$  当前所走过的城市，集合随着  $\text{tabu}_k$  进化过程作动态调整。在搜索过程中，蚂蚁根据各条路径上的信息量及路径的启发信息来计算状态转移概率。 $p_{ij}^k(t)$  表示在  $t$  蚂蚁  $k$  由城市  $i$  转移到城市  $j$  的状态转移概率

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ik}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} & (3-1) \\ 0 & \end{cases}$$

式中， $\text{allowed}_k$  表示蚂蚁  $k$  下一步允许选择的城市； $\alpha$  为信息启发式因子，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的所起的作用，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，反映了蚂蚁在运动过程中启发信息在蚂蚁选择路径中的受重视程度，其值越大，则该状态转移概率越接近于贪心规则<sup>[13]</sup>； $\eta_{ij}(t)$ 为启发函数，其表达式如下

$$\eta_{ij}(t) = 1/d_{ij} \quad (3-2)$$

式中， $d_{ij}$ 表示相邻两个城市之间的距离。对蚂蚁  $k$  而言， $d_{ij}$  越小，则  $\eta_{ij}(t)$  越大， $p_{ij}^k(t)$  也就越大。显然，该启发函数表示蚂蚁从城市  $i$  转移到城市  $j$  的期望程度。

为了避免残留信息素过多引起残留信息淹没启发信息，在每只蚂蚁走完一步或者完成对所有  $n$  个城市遍历（也即一个循环结束）后，要对残留信息进行更新处理。这种更新策略模仿了人类大脑记忆的特点，在新信息不断存入大脑的同时，存储在大脑中的旧信息随着时间的推移逐渐淡化，甚至忘记。由此， $t+n$  时刻在路径  $(i,j)$  上的信息量可按如下规则进行调整

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3-3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3-4)$$

式中， $\rho$  表示信息素挥发系数，则  $1-\rho$  表示信息素残留因子，为了防止信息的无限积累， $\rho$  的取值范围为： $\rho \in [0,1]$ ； $\Delta\tau_{ij}(t)$  表示本次循环中路径  $(i,j)$  上的信息素增量，初始时刻  $\Delta\tau_{ij}(0) = 0$ ， $\Delta\tau_{ij}^k(t)$  表示第  $k$  只蚂蚁在本次循环中留在路径  $(i,j)$  上的信息量。

根据信息素更新策略的不同，Dorigo M 提出了三种不同的基本蚁群算法模型，分别称之为 Ant-Cycle 模型、Ant-Quantity 模型及 Ant-Density 模型，其差别在于  $\Delta\tau_{ij}^k(t)$  求法的不同。

在 Ant-Cycle 模型中

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{若第 } k \text{ 只蚂蚁在本次循环中经过 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (3-5)$$

式中， $Q$  表示信息素强度，它在一定程度上影响算法的收敛程度； $L_k$  表示第  $k$  只蚂蚁在本次循环中所走过路径的总长度。

在 Ant-Quantity 模型中



$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{d_{ij}}, & \text{若第}k\text{只蚂蚁在}t\text{和}t+1\text{之间经过}(i, j) \\ 0, & \text{否则} \end{cases} \quad (3-6)$$

在 Ant-Density 模型中

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q, & \text{若第}k\text{只蚂蚁在}t\text{和}t+1\text{之间经过}(i, j) \\ 0, & \text{否则} \end{cases} \quad (3-7)$$

公式（3-6）和（3-7）中利用的是局部信息，即蚂蚁完成一步后更新路径上的信息素；而公式（3-5）中利用的是整体信息，即蚂蚁完成一次循环后更新所有路径上的信息素，在求解 TSP 时性能较好，因此通常采用公式(3-5)作为蚁群算法的基本模型。



## 第四章 基本蚁群算法求解 TSP

本章首先利用了 MATLAB 平台做出一个参数设置平台，是为后一章做参数讨论做铺垫，主要是运用 MATLAB 中 GUIDE 设计工具。接着给出了基本蚁群算法求解 TSP 的实现流程，并且做了相应的仿真，得出了仿真结果。并在第五章中进行了蚁群算法主要参数设置的分析。

### 4.1 参数设置平台

为了后一章利用仿真数据讨论参数选择更加方便，首先在这里设计了一个简易的参数设置平台。直接用 M 文件编写 GUI 程序很烦琐，而使用 GUIDE 设计工具可以大大提高工作效率。GUIDE 相当于一个控制面板，从而可以利用各种设计工具以辅助完成界面设计任务<sup>[14]</sup>。

第一步，首先在命令窗口打入 `guide`，出现 GUIDE 集成界面

第二步，利用工具条中的工具，快速建立如下图的界面布局

第三步，设置控件属性，如文本框属性 `string` 设置成空，用来输入基本蚁群算法主要参数，包括信息启发式因子  $\alpha$ ，期望启发式因子  $\beta$ ，挥发系数  $\rho$ ，蚂蚁信息素强度，蚂蚁数目。静态文本属性 `string` 设置为对应的参数名。

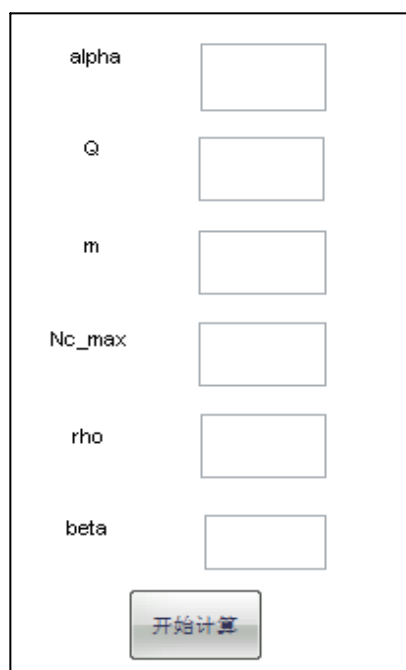


图 4.2 基本蚁群算法参数设置平台



第四步，保存设计。GUIDE 默认把 GUI 程序保存为两个同名文件：一个是 .fig 文件，用来保存窗体布局 and 所有控件的界面信息；一个是 .m 文件，该文件的初始内容是 GUIDE 自动产生的程序框架，其中包括了各个控件回调函数的定义。

第五步，生成的 .m 文件中在开关按钮开关的回调函数中调入基本蚁群算法的程序，利用 get() (取属性值函数) 来传递控件中用户输入的参数值，如语句 `m=str2double(get(handles.exit5,'string'))`。这样就实现了这个简易的参数设置平台。

## 4.2 基本蚁群算法求解 TSP 的实现流程

### 4.2.1 基本蚁群算法的实现步骤

以 TSP 为例，基本蚁群算法的具体实现步骤如下：

(1) 参数初始化。令时间  $t=0$  和循环次数  $N_c=0$ ，设置最大循环次数  $N_{c_{\max}}$ ，将  $m$  只蚂蚁置于  $n$  个城市上，令每条边  $(i,j)$  的初始化信息量  $\tau_{ij}(t) = \text{const}$ ，其中  $\text{const}$  表示常数，且初始时刻  $\Delta\tau_{ij}(0) = 0$ 。

(2) 循环次数  $N_c \leftarrow N_c + 1$ 。

(3) 蚂蚁的禁忌表索引号  $k=1$

(4) 蚂蚁数目  $k \leftarrow k + 1$ 。

(5) 蚂蚁个体根据状态转移概率公式 (3-1) 计算的概率选择城市  $j$ 。

(6) 修改禁忌表指针，即选择好之后将蚂蚁移动到新的城市，并把该城市移动到该蚂蚁个体的禁忌表中。

(7) 若集合  $C$  中的城市未遍历完，即  $k < m$ ，则跳转到到第 (4) 步，否则继续往下一步执行。

(8) 根据公式 (3-3) 和 (3-4) 更新每条路径上的信息量。

(9) 若满足结束条件，即循环次数  $N_c \geq N_{c_{\max}}$ ，则循环结束并输出程序计算结果，否则清空禁忌表并跳转到第 (2) 步。

### 4.2.2 基本蚁群算法的结构流程

以 TSP 为例，基本蚁群算法的结构流程如图 4.1 所示。

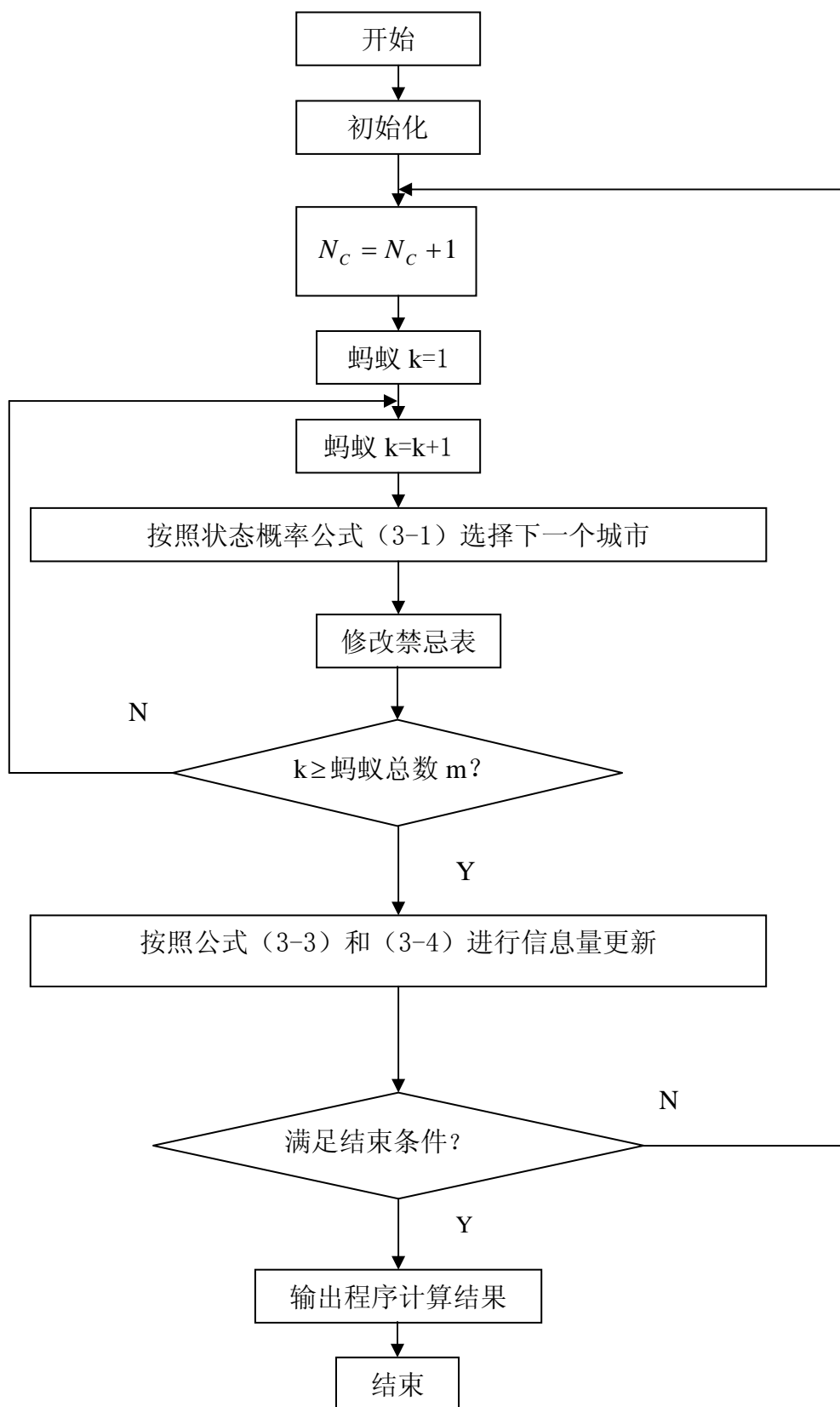


图 4.1 基本蚁群算法的程序结构流程



### 4.3 基本蚁群算法的仿真结果

以基本蚁群算法为例，并且对信息启发式因子 $\alpha$ ，期望启发式因子 $\beta$ ，蚂蚁数目 $m$ ，挥发系数 $\rho$ ，蚂蚁信息素强度 $Q$ 进行设置，这里设置缺省值为 $\alpha=1$ 、 $\beta=5$ 、 $\rho=0.5$ 、 $m=51$ 、 $Q=100$ ，算法能在全局搜索能力和收敛速度方面显示出更好的性能，并且算法也得到了较好的最优路径值。

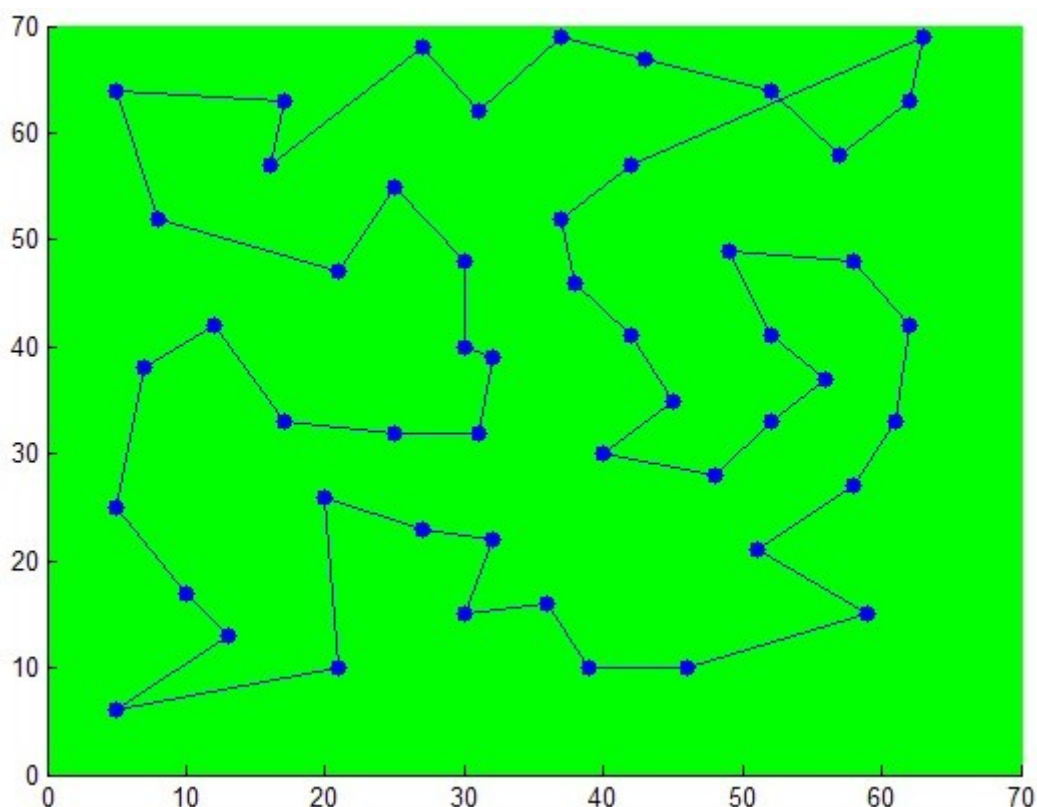


图 4.3  $N_{C_{MAX}}$  中迭代的最佳路线路径图

图 4.3 表示的是 51 个城市的连接图，说明基本蚁群算法很好的解决了 TSP（NP 完全问题），体现了蚁群算法的具有正反馈，自组织等优点。

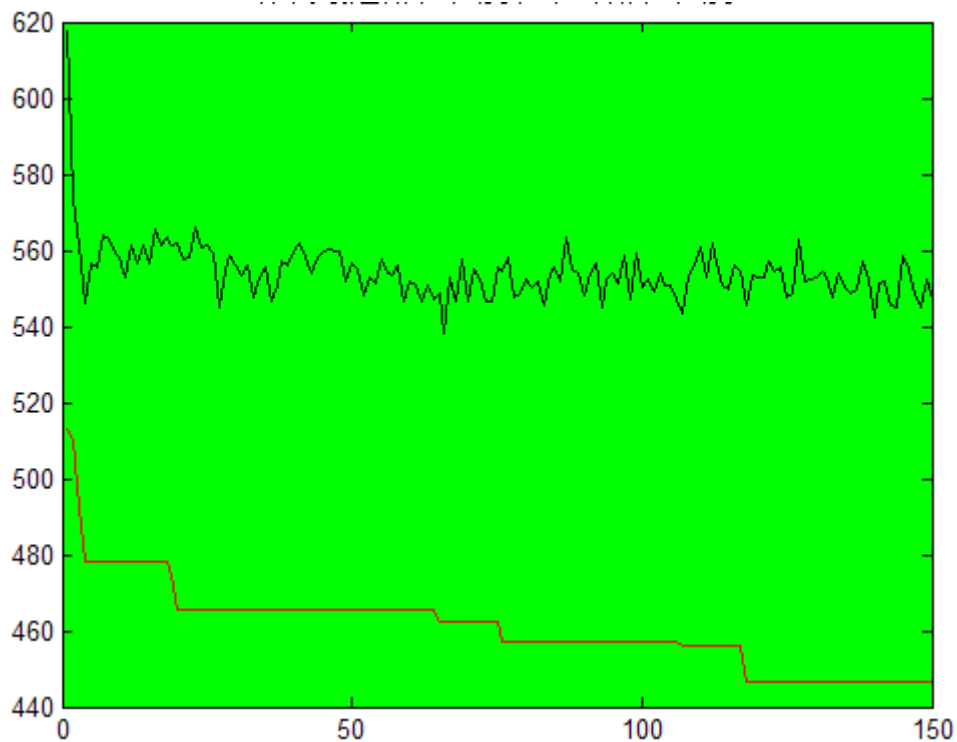


图 4.4 各迭代平均路径长度和最短路线路径图

图 4.4 黑线（上线）是表示各迭代平均路径长度，红线（下线）表示各迭代最短路径（即最优路径）。黑线平均路径比最短路径图高出很多，因为有的蚂蚁走的路径好，比较短，有的蚂蚁走的路径差，非常长，说明蚁群算法也存在过早局部收敛的缺点。从红线可以看出到迭代超过 120 算法才收敛结束，说明蚁群算法也同样存在着收敛速度慢，所以参数的设置就显得比较重要。





## 第五章 蚁群算法的主要参数设置

### 5.1 关键参数介绍

蚁群算法中  $\alpha$ 、 $\beta$ 、 $\rho$  等参数对算法性能有很大的影响。 $\alpha$  值的大小表明留在每个结点上的信息量受重视的程度， $\alpha$  值越大，蚂蚁选择以前经过的路线的可能性越大，但过大会使搜索过早陷于局部最小解； $\beta$  的大小表明启发式信息受重视的程度， $\beta$  值越大，蚂蚁选择离它近的城市的可能性越大； $\rho$  表示信息素的保留率，如果它的值取的不恰当，得到的结果会很差。如下几个图的比较：

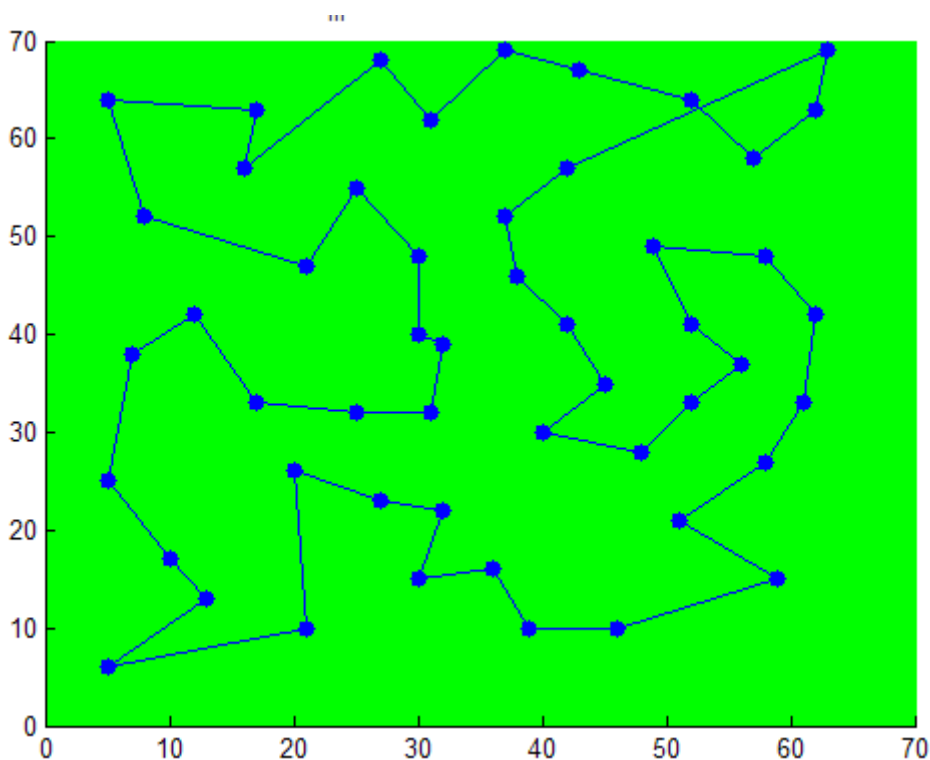


图 5.1 参数设置合理的  $N_{C_{MAX}}$  迭代最佳路线路径图

图 5.1 可以看出路线比较清晰，并且没有出现某一区域出现局部回路问题，也就是说没有出现局部最优，体现了蚁群算法具有全局搜索能力。

图 5.2 可以看出路线很混乱，说明图中很多地方出现局部最优的情况，最终得到的最优路径值就会很差。

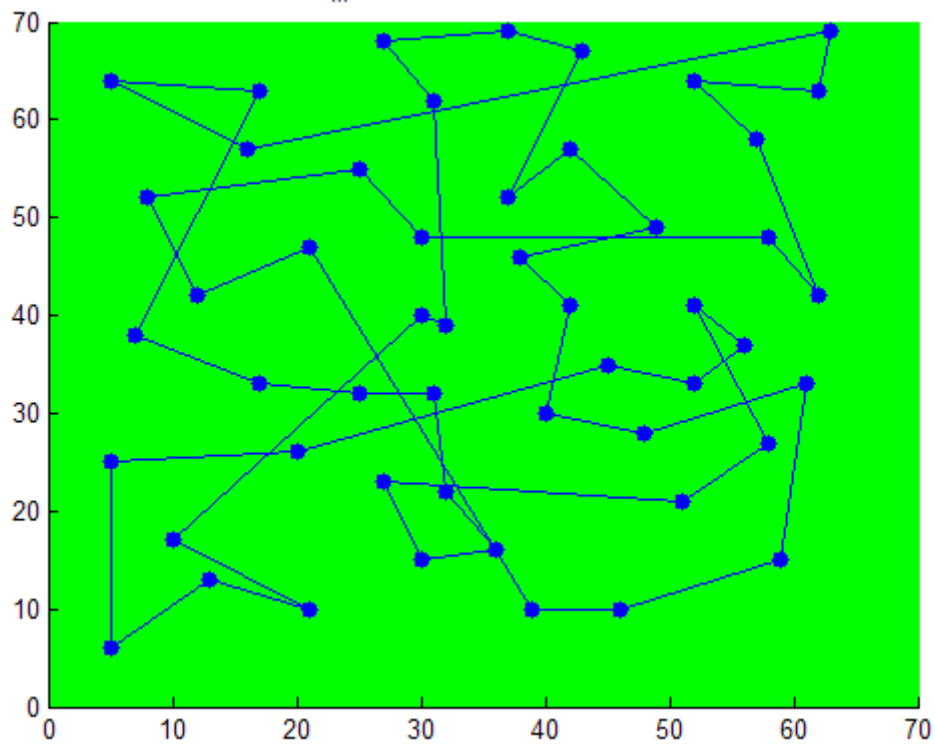


图 5.2 参数设置不合理时  $N_{C_{MAX}}$  迭代最佳路线路径图

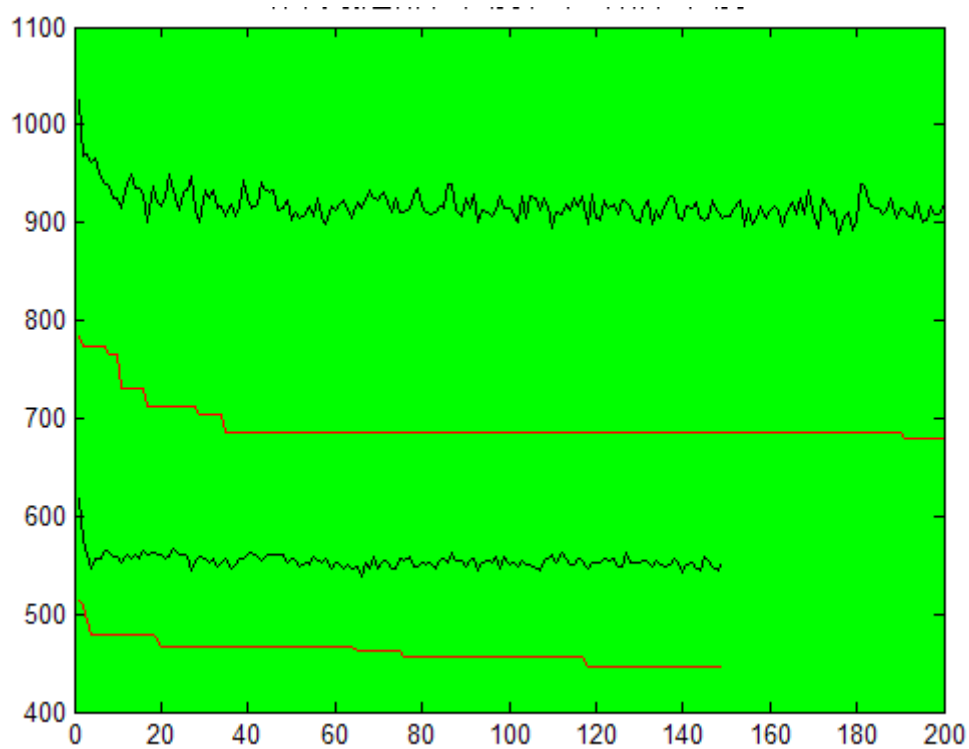


图 5.3 前后不同参数设置的最优路线路径比较



图 5.3 中是图 5.1 和图 5.2 各迭代平均路径长度（黑线表示）和最优路线路径长度（红线表示）的比较。参数设置较好的各迭代平均路径长度起伏变化更加稳定，同时最优路线路径的长度结果更加好（即位于图下方的两条线）。而参数设置较差的却较早的陷入局部优化，得到的结果很差，并且各迭代的平均路径长度起伏变化非常不稳定（即位于图上方的两条线）。所以选择合理的参数能使蚁群算法的性能更好。

## 5.2 不同参数设置的试验

蚁群算法中信息启发式因子  $\alpha$ 、期望启发式因子  $\beta$ 、挥发系数  $\rho$ 、蚂蚁数目  $m$  的设置值的不同会对蚁群算法的全局搜索能力和收敛速度产生很大的影响，接下来会对不同参数的设置对蚁群算法性能的影响进行描述，然后给出实验结果和结论。

### 5.2.1 信息素挥发度的设置

在蚁群算法中，人工蚂蚁是具有人类记忆功能的，随着时间的推移，以前留下的信息将要逐渐消逝。在算法模型中用参数  $\rho$  表示信息消逝程度（或称信息素挥发度），而  $1-\rho$  就是信息素保留系数。蚁群算法存在着收敛速度慢、易于陷入局部最优等缺陷。而信息素挥发度  $\rho$  的大小直接关系到蚁群算法的全局搜索能力及其收敛速度：由于信息素  $\rho$  的存在，当要处理的问题规模比较大时，会使那些从来未被搜索到的路径（可行解）上的信息量减小到接近于 0，因而降低了算法的全局搜索能力，而且当  $\rho$  过大时以前搜索过的路径被再次选择的可能性过大，也会影响到算法的随机性能和全局搜索能力；反之，通过减小信息素挥发度  $\rho$  虽然可以提高算法的随机性能和全局搜索能力，但又会使算法的收敛速度降低。关于蚁群算法中信息素挥发度  $\rho$  对算法性能的影响及其在实际应用中的选择，可以通过计算机仿真实验来分析和确定。

### 5.2.2 蚁群数量的设置

蚁群算法是一种随机搜索算法，通过多个候选解（可行解的一个子集）组成的群体的进化过程来寻求最优解，在该过程中既需要每个个体的自适应能力，更需要群体的相互协作。蚁群在搜索过程中之所以表现出复杂而有序的行为，个体之间的信息交流与相互协作起着至关重要的作用。对于 TSP 问题，单个蚂蚁在一次循环中所经过的路径，表现为问题可行解集中



的一个解， $m$  只蚂蚁在一次循环中所经过的路径，则表现为问题解集中的一个子集。显然；子集越大(即蚁群数量多)可以提高蚁群算法的全局搜索能力以及算法的稳定性;但蚂蚁数目增大后，会使大量的曾被搜索过的解(路径)上的信息量的变化比较平均，信息正反馈的作用不明显，搜索的随机性虽然得到了加强，但收敛速度减慢;反之，子集较小(即蚁群数量少)，特别是当要处理的问题规模比较大时，会使那些从来未被搜索到的解(路径)上的信息量减小到接近于 0，搜索的随机性减弱，虽然收敛速度加快，但会使算法的全局性能降低，算法的稳定性差，容易出现过早停滞现象。

### 5.2.3 启发式因子的设置

信息启发式因子  $\alpha$  反映蚂蚁在运动过程中所积累的信息量(即残留信息量  $\tau_{ij}(t)$ ) 在指导蚁群搜索中的相对重要程度，期望值启发式因子  $\beta$  反映蚂蚁在运动过程中启发信息(即期望值  $\eta_{ij}$ ) 在指导蚁群搜索中的相对重要程度。期望值启发式因子  $\beta$  的大小反映了蚁群在路径搜索中先验性、确定性。 $\beta$  因素作用的强度，其值越大，蚂蚁在某个局部点上选择局部最短路径的可能性越大，虽然搜索的收敛速度得以加快，但蚁群在最优路径的搜索过程中随机性减弱，易于陷入局部最优;而信息启发式因子  $\alpha$  的大小则反映了蚁群在路径搜索中随机性因素作用的强度，其值越大，蚂蚁选择以前走过的路径的可能性越大，搜索的随机性减弱，当  $\alpha$  值过大也会使蚁群的搜索过早陷于局部最优。蚁群算法的全局寻优性能，首先要求蚁群的搜索过程必须有很强的随机性;而蚁群算法的快速收敛性能，又要求蚁群的搜索过程必须要有较高的确定性。因此，两者对蚁群算法性能的影响和作用相互配合、密切相关的。

## 5.3 不同参数设置的实验结果和结论

本文采用缺省参数设置为  $\alpha = 1$ 、 $\beta = 5$ 、 $\rho = 0.5$ ，蚂蚁的数目总是设置为城市的总数目，即初始时刻每个城市放置一个蚂蚁。实验中所有 TSP 问题数据来源于 Eli51 城市问题（这是蚁群算法解决 TSP 问题的经典城市数据）。

### 5.3.1 信息素挥发度的选择设置

按照以上的实验描述，在其他参数不变的情况下，使信息素挥发系数的变化为  $\rho = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ 。信息素挥发系数对性能影响仿真结果如表 5.1 和图 5.4 所示：



表 3.1 信息素挥发度对算法性能的影响

挥发系数	最优路径长度	搜索循环次数
0.1	452.3	542
0.3	471.5	263
0.5	444.5	142
0.7	442.8	103
0.9	452.3	26

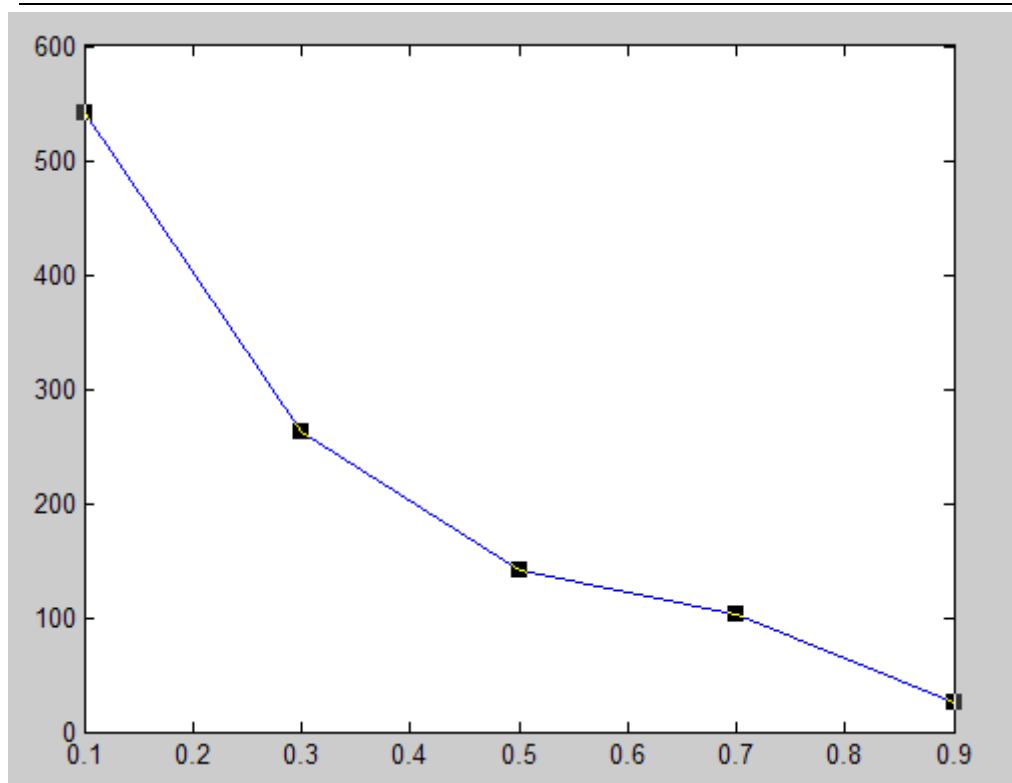


图 5.4 信息挥发系数  $\rho$  与最优解循环次数的关系

由仿真实验不难看出，在其他参数相同的情况下，信息素挥发度  $\rho$  的大小对蚁群算法的收敛性能影响极大，特别是当  $\rho$  很小时，由于路径上的残留信息占主导地位，信息正反馈的作用相对较强，搜索的随机性减弱，因而蚁群算法的收敛速度加快；而在  $\rho$  比较大时，由于信息正反馈的作用占主导地位，搜索的随机性增强，全局搜索能力增强，但收敛速度太慢。因而关于蚁群算法的信息素挥发度  $\rho$  的选择，必须综合全局搜索能力和收敛速度两项性能指



标，由图可知在  $\rho = 0.5 \sim 0.7$  范围内，算法的性能比较稳定和一致，搜索的全局性和收敛速度比较好。

### 5.3.2 蚁群数量的选择设置

在这里还是采用与上面相同的方法，假设其他缺省参数不变，只改变蚂蚁数目  $m$  的数值。 $m \in \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$ 。蚂蚁数量对算法性能的影响的仿真结果如表 5.2 和图 5.5。

表 5.2 蚂蚁数量  $m$  对算法性能的影响

蚂蚁数目 $m$	最优路径长度	搜索循环次数
10	465.5	29
15	453.8	35
20	445.7	44
25	444.8	46
30	445.6	85
35	442.2	136
40	443.5	235
45	437.9	331
50	439.5	456

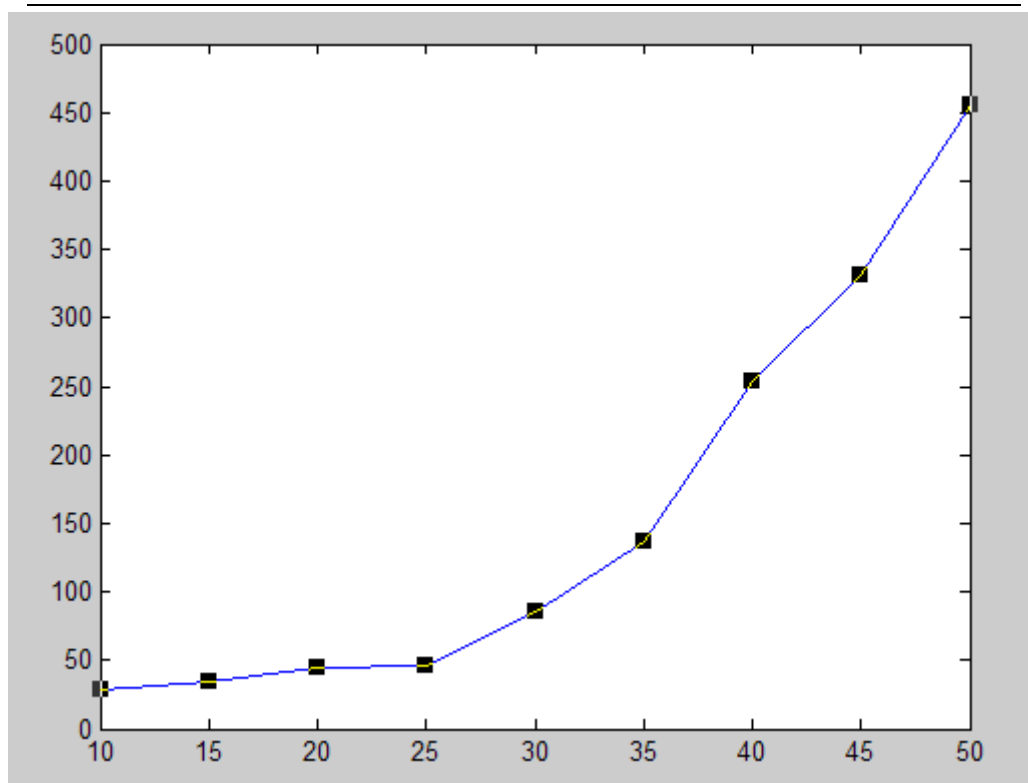


图 5.5 蚂蚁数量  $m$  与最优解搜索循环次数的关系



由仿真实验结果不难看出，蚂蚁数量  $m$  的大小对蚁群算法循环次数(收敛性能)的影响基本呈线性规律变化，当蚂蚁数量过大(如接近问题的规模)时，虽然搜索的稳定性和全局性得到提高，但算法的收敛速度减慢。关于蚁群算法中蚂蚁数量  $m$  的选择，也应该综合考虑算法的全局搜索能力和收敛速度两项指标，针对具体问题的应用条件和实际要求，在全局搜索能力和收敛速度两方面作出合理或折中的选择。上述的实验结果表明在  $m=20\sim 25$  的范围内(相对于本问题的规模为  $n=51$ )，算法的各项性能相对较为平稳。所以本文认为，在蚁群算法中蚂蚁数量的选择宜取为  $m=\sqrt{n} \sim n/2$  之间(其中  $n$  为问题的规模)。

### 5.3.3 启发式因子的选择设置

关于蚁群算法中启发式因子  $\alpha$  及  $\beta$  对算法性能的影响及其在实际应用中的选择，也可以通过计算机仿真实验来分析和确定。在其它参数不变的情况下，取启发式因子  $\alpha$  和  $\beta$  不同数值的组合。启发式因子  $\alpha$  和  $\beta$  对算法性能影响的仿真结果如表 5.3。

表 5.3 启发式因子  $\alpha$  和  $\beta$  的不同组合对算法性能的影响

$\alpha$	$\beta$	最优路径长度	搜索循环次数
0.1	0.5	452.6	158
0.1	1	445.2	94
0.2	2	442.1	81
0.3	3	441.8	65
1.0	4	456.2	51
5.0	8	458.9	23
10.0	10	469.9	15

从仿真实验不难看出，蚁群算法中启发式因子  $\alpha$  及  $\beta$  的不同选取对算法性能有极大的影响：差的搜索结果且出现过早停滞( $\alpha$  及  $\beta$  均过大时)；对于过大的  $\alpha$  值，相当于给予路径上的信息素  $\tau_{ij}$  在蚂蚁的搜索过程中的重要性给予充分的重视，蚂蚁完全依赖信息素  $\tau_{ij}$  的引导进行搜索，如果此时期望信息  $\eta_{ij}$  的启发式因子  $\beta$  相对比较大，则将导致局部最优路径上的信息正反馈作用极强，算法必然会出现过早收敛的现象。当问题规模较大时，在这种情况下所搜索到的通常是局部最优解。差的搜索结果但不出现过早停滞( $\alpha$  和  $\beta$  均过小时)：对于过小的





$\alpha$  值，相当于对路径上的信息素  $\tau_{ij}$  在蚂蚁的搜索过程中的重要性未予足够重视，蚂蚁完全依赖期望信息  $\eta_{ij}$  的引导进行搜索，如果此时期望信息  $\eta_{ij}$  的启发式因子  $\beta$  相对也较小，则将导致蚁群陷入了纯粹的、无休止的随机索中(即不停滞)，在这种情况下所进行的搜索一般也很难找到最优解。在本问题中， $\beta$  取 2 左右， $\alpha$  取 0.2 到 0.5 之间的值为宜，蚁群算法均能获得较好的搜索结果，算法的循环次数少(即收敛速度快)，并且性能非常接近。





## 第六章 总结和展望

### 6.1 工作总结

蚁群算法解决 TSP 问题只是解决了众多组合优化问题的一种，但就是围绕着 TSP 问题的解决才有了蚁群算法的发展和创新。它涉及了仿生学，程序设计学，计算机科学仿真技术多个学科，是信息处理领域的一项前沿技术。本文的前期工作是介绍了基本蚁群算法的原理和蚁群算法的机制原则，其中也详细介绍了组合优化问题。

本文的重点是利用蚁群算法解决 TSP 问题，并利用 MATLAB 成功进行了仿真。其中重点讨论了如何通过选择合理的参数配置使蚁群算法性能得到提高，并用大量的仿真实验结果予以证明。

总的来说，本文有两个难点，基本蚁群算法原理的理解，与利用 MATLAB 进行仿真实验解决 TSP 问题。

### 6.2 技术展望

蚁群算法发展至今，人们已经针对不同的具体问题提出了许多不同类型的改进蚁群算法模型，但这些改进的蚁群算法模型往往普适性不强，同时基本蚁群算法模型也不能直接应用于解决所有的优化问题。另外，自然界中的真实蚁群还有许多其他的智能行为，用逆向思维和发散思维开发不同的蚁群算法模型是一条新的发展思路。基于此，今后可以从以下几个方面对其模型进行改进。

- 1、设计一种通用的蚁群算法普适性模型。
- 2、进一步研究自然蚁群行为，提出更加智能化的蚁群混合行为模型。
- 3、摆脱传统模型框架，开发新的蚁群算法模型。

因此，关于蚁群算法理论及其应用的研究必将是一个长期的研究课题。相信随着人们对仿生智能系统理论及应用研究的不断深入，蚁群算法这一新兴的仿生优化算法必将展现出更加广阔、更加引人注目的发展前景。



## 参考文献

- [1] 段海滨.蚁群算法原理及其应用[M]. 科学出版社.2005
- [2] 杨海.王洪国.徐卫志.蚁群算法的应用研究与发展[J].科学和技术信息学报 2007 年 第 28 期
- [3] 刘志硕.申金升.柴跃廷 一种求解车辆路径问题的混合多蚁群算法[J].系统仿真学报 2007 年第 19 卷第 15 期
- [4] 劳眷.TSP 问题中的蚁群优化算法研究[D].长沙:湖南大学,2007 年
- [5] 张宗永.孙静.谭家华.蚁群算法的改进及其应用[J].上海交通大学学报 2002 年第 36 卷第 11 期
- [6] 孙骏.基于蚁群优化算法的 TSP 问题研究[D].武汉:武汉理工大学,2005 年
- [7] 刘乃文.刘方爱.蚁群算法用于 TSP 的并行策略及模型[J].2007 年第 24 卷第 12 期
- [8] 尹晓峰.基于 MATLAB 的混合型蚁群算法求解旅行商问题[J].铁路计算机应用学报 2005 年第 14 卷第 9 期
- [9] 胡小兵.袁锐.黄席.易继军.蚁群算法原理的仿真与研究[J].2004 年第 21 卷第 8 期
- [10] 张宏达.郑全第.基于蚁群算法的 TSP 的仿真和研究[J].2005 年第 35 卷第 4 期
- [11] Dorigo M.Ant algorithms and stigmergy.Future Generation Computer Systems.2000,16(8),873~871
- [12] Dorigo M.Gambardella L M.Ant colony system:cooperative learning approach to the traveling salesman problem[J].IEEE Transactions on Evolutionary Computation.1997,1(1),53~66
- [13] Dorigo M.Luca M.The ant colony algorithm applied to the nuclear reload problem.Annals of Nuclear Energy.2002,29(12),1455~1470
- [14] Dorigo M.Ant colony system: optimization by a colony of cooperating agents.IEEE Transactions on System,Man and Cybernetics-Part B.1996,26(1),29~41



## 致 谢

本文是在谭静老师的耐心指导下完成的。在课题的研究过程中，她每周都对我的毕业设计进行悉心指导和帮助。在我遇到困难时，她也多次给我鼓励和鞭策。谭老师在科学上的执着追求、严谨的治学态度、学者的先锋精神，严以律己、宽厚待人和亦师亦友的高尚品德，给我树立了榜样，使我在今后的工作和生活中受益匪浅。

经过这段时间的毕业设计，我感觉到掌握扎实的基础知识和学会使用必要工具的重要性，深刻体会到网络资源的巨大作用，在遇到难以解决的问题时，可以在因特网这个无穷的空间中寻找所需的资料，到相关的论坛上去求得帮助；学会灵活运用Internet这个现代工具是我们必备的素质。同时毕业设计对我的英语水平也提出了较高的要求，通过阅读英文资料、翻译英文材料切实提高了我使用英语的水平，使我在今后的学习中不至落后现代技术发展的潮流。

在进行毕业设计的过程中，对于我的家人和朋友，我的感激之情无以言表，仅以此文献给他们，感谢他们一直对我的关爱，陪我一路走过艰难的历程。回首大学四年，往事历历在目，心绪难以平复，如此多的关心和帮助让我感到莫大的幸运，感觉充满力量，这些将支持我走向新的岗位，为社会为他人贡献我的绵薄之力。



## 附 录

### 附录 1：基于基本蚁群算法的 TSP 问题源码

```
%%clear all

%%close all

clc

m = str2double(get(handles.edit5,'String'));
alpha = str2double(get(handles.edit1,'String'));
beta = str2double(get(handles.edit8,'String'));
Q = str2double(get(handles.edit4,'String'));
rho = str2double(get(handles.edit7,'String'));
Nc_max = str2double(get(handles.edit6,'String'));

C=[37 52;49 49;52 64;20 26;40 30;21 47;17 63;31 62;
    52 33;51 21;42 41;31 32;5 25;12 42;36 16;52 41;
    27 23;17 33;13 13;57 58;62 42;42 57;16 57;8 52;
    7 38;27 68;30 48;43 67;58 48;58 27;37 69;38 46;
    46 10;61 33;62 63;63 69;32 22;45 35;59 15;5 6;
    10 17;21 10;5 64;30 15;39 10;32 39;25 32;25 55;48 28;56 37;30 40];
n=size(C,1);
D=ones(n,n);
for i=1:n
    for j=1:n
        if i<j
            D(i,j)=sqrt((C(i,1)-C(j,1))^2+(C(i,2)-C(j,2))^2);
        end
        D(j,i)=D(i,j);
    end
end
```



```
end

end

eta=1./D;

pheromone=ones(n,n);

tabu_list=zeros(m,n);

Nc=0;

routh_best=zeros(Nc_max,n);

length_best=ones(Nc_max,1);

length_average=ones(Nc_max,1);

while Nc<Nc_max

    rand_position=[];

    for i=1:ceil(m/n)

        rand_position=[rand_position,randperm(n)];

    end

    tabu_list(:,1)=(rand_position(1:m))';

    for j=2:n

        for i=1:m

            city_visited=tabu_list(i,1:(j-1));

            city_remained=zeros(1,(n-j+1));

            probability=city_remained;

            cr=1;

            for k=1:n

                if length(find(city_visited==k))==0

                    city_remained(cr)=k;

                    cr=cr+1;

                end

            end

        end

    end

end
```



---

```
for k=1:length(city_remained)

    probability(k)=(pheromone(city_visited(end),city_remained(k)))^alpha*(eta
a(city_visited(end),city_remained(k)))^beta;

    end

    probability=probability/sum(probability);

    pcum=cumsum(probability);

    select=find(pcum>=rand);

    to_visit=city_remained(select(1));

    tabu_list(i,j)=to_visit;

end

end

if Nc>0

    tabu_list(1,:)=routh_best(Nc,:);

end

total_length=zeros(m,1);

for i=1:m

    r=tabu_list(i,:);

    for j=1:(n-1)

        total_length(i)=total_length(i)+D(r(j),r(j+1));

    end

    total_length(i)=total_length(i)+D(r(1),r(n));

end

length_best(Nc+1)=min(total_length);    position=find(total_length==length_best(Nc+1));

routh_best(Nc+1,:)=tabu_list(position(1),:); length_average(Nc+1)=mean(total_length);

Nc=Nc+1

delta_pheromone=zeros(n,n);

for i=1:m

    for j=1:(n-1)
```

---



```
delta_pheromone(tabu_list(i,j),tabu_list(i,j+1))=delta_pheromone(tabu_list(i,j),tabu_list(i,j+1))+Q/total_length(i);
end
delta_pheromone(tabu_list(i,n),tabu_list(i,1))=delta_pheromone(tabu_list(i,n),tabu_list(i,1))+Q/total_length(i);
end
pheromone=(1-rho).*pheromone+delta_pheromone;
tabu_list=zeros(m,n);
end
position=find(length_best==min(length_best));
shortest_path=routh_best(position(1),:)
shortest_length=length_best(position(1))
figure(1)
set(gcf,'Name','Ant Colony OptimizationFigure of shortest_path','Color','w')
N=length(shortest_path);
scatter(C(:,1),C(:,2),50,'filled');
hold on
plot([C(shortest_path(1),1),C(shortest_path(N),1)],[C(shortest_path(1),2),C(shortest_path(N),2)])
title('Nc_max代中最佳路线路径图');
set(gca,'Color','g')
hold on
for i=2:N

plot([C(shortest_path(i-1),1),C(shortest_path(i),1)],[C(shortest_path(i-1),2),C(shortest_path(i),2)])
    hold on
end
figure(2)
set(gcf,'Name','Ant Colony Optimization--Figure of length_best and length_average','Color','w')
```



```
plot(length_best,'r')  
title('各代平均路线长度和最短路径长度')  
set(gca,'Color','g')  
hold on  
plot(length_average,'k')
```