# 第八章 基于蚁群算法的工序路径建模与优化方法

## 一. 蚁群算法概述

### 1.算法起源

20世纪90年代意大利学者M．Dorigo，V．Maniezzo，A．Colorni等从生物进化的机制中受到启发，通过模拟自然界蚂蚁搜索路径的行为，提出来一种新型的模拟进化算法—— 蚁群算法，这是群智能理论研究领域的主要算法。用该方法求解TSP问题、分配问题、job-shop调度问题，取得了较好的试验结果。初步的研究显示蚁群算法在求解复杂优化问题（特别是离散优化问题）方面有一定优势，是一种拥有较好发展前景的算法。

**2.提出背景**

20世纪50年代中期人们从生物进化的机理中受到启发，创立了仿生学，提出了许多用以解决复杂优化问题的新方法，如进化策略、遗传算法等，这些算法成功地解决了一些实际问题。

与大多数基于梯度的应用优化算法不同，群智能依靠的是概率搜索算法。虽然概率搜索算法通常要采用较多的评价函数，但是与梯度方法及传统的演化算法相比，其优点还是显著的 ，主要表现在以下几个方面：

1 无集中控制约束，不会因个别个体的故障影响整个问题的求解，确保了系统具备更强的鲁棒性；

2 以非直接的信息交流方式确保了系统的扩展性；

3 并行分布式算法模型，可充分利用多处理器；

4 对问题定义的连续性无特殊要求；

5 算法实现简单。

群智能理论研究领域有两种主要的算法：蚁群算法(Ant Colony Optimization, ACO)和粒子群算法（Particle Swarm Optimization, PSO）。前者是对蚂蚁群落食物采集过程的模拟，已成功应用于许多离散优化问题。粒子群算法也是起源于对简单社会系统的模拟，最初是模拟鸟群觅食的过程，但后来发现它是一种很好的优化工具。

### 蚁群算法的基本原理

**1.觅食行为**

蚂蚁是一种社会性昆虫，相互之间有简单的通讯和信息传递。蚁群的觅食行为是它们重要的行为之一，引起了研究者的注意，因为蚂蚁有能力在没有任何可见提示的情况下，总能找到从蚁穴到食物源的最短路径，甚至能根据环境的变化(如遇到障碍物等)而动态地选择新的最短路径。蚁群在很多情况下还能完成远远超出蚂蚁个体能力的复杂任务，体现出较高的智慧水平。

蚂蚁能够在没有任何可见提示下找出从蚁穴到食物源的最短路径，并且能随环境的变化而动态地搜索新的路径，产生新的选择，找到新的最短路径。其中的“奥妙”在于每只蚂蚁在走过的路径上分泌一种化学物质--信息素，其它蚂蚁在运动过程中能感知这种物质的存在和强度，并以此指导自己的运动方向，沿着信息素浓度高的方向移动，最终形成几乎所有蚂蚁都会选择的最短路径。在选择过程中，蚂蚁也有一定的小概率“失误”，选择信息素浓度不高的方向前进，这样则可以避免因大部分蚂蚁选择相对较短的路径而不再选择最短路径的问题。可见蚂蚁个体之间并不进行直接的信息传递和交流，它们只同周围的环境进行相互作用，通过周围的环境间接地影响同伴，进行信息的传递和更新。

关于在受约束条件下的蚂蚁觅食行为，Deneubourg等人进行了著名的双支桥实验，并根据实验观察的结果提出信息素模型。此模型中，在某一时刻蚂蚁选择分支的概率依赖于已经走过该分支的蚂蚁数量，走的蚂蚁数量越多，选择的概率越高。双支桥实验及其结果如图8-1所示：



图8-1 等长双桥实验

实验中，蚂蚁从蚁穴出发，通过两个长度相等的分支到达食物源，两个分支上最初没有信息素。实验将30分钟内蚂蚁通过两个分支的百分比进行了记录，结果如图8-1(b)所示，起初蚂蚁选择两个分支的概率相等，经过最初的一个短暂的震荡阶段，蚂蚁开始倾向于沿着一条分支移动，最后大部分(百分之九十以上)的蚂蚁都选择了同一条路径。

等长度双桥实验还可以扩展到分支长度不相等的情况，实验的结果也是经过初始波动后，蚂蚁逐渐趋向于选择短的分支，并且最终选择短分支的几率随着两个分支的比(长分支：短分支)的增大而增长。

蚂蚁觅食过程如图8-2所示。



图8-2 蚂蚁觅食行为

图中先有5只蚂蚁(1、2、3、4、5)外出觅食，沿途撒下信息素。有如图的五种路线，显然蚂蚁2所走的路线最短，最先返回巢穴，因此蚂蚁2再次出发时，路径2上信息素最多(因为此时其它蚂蚁还未返回，此路径上经过的蚂蚁数量为2)，它依旧按照原路(路径2)行进，而其他蚂蚁返回再去搬运食物时，路径2上面的信息素最多，因此它们也选择路径2方向移动，从而路径2成为当前的最短路径。但图中的最短路径并不是路径2，而是路径6，这条路径的发现是靠蚂蚁选择路径时发生的小概率“错误”完成的，如蚂蚁1在选择路径时“误”选了路径6，于是就发现了最短路径，随着越来越多的“错误”和信息素积累更新及选择路径的规则，路径6便确定成为最终的最优路径

1. **人工蚁的提出**

蚁群算法中，提出人工蚁的概念。人工蚁一方面是真实蚂蚁行为特征的一种抽象，保留着蚁群觅食行为中最关键的部分；另一外面，它还具有一些真实蚂蚁不具有的特征，如人工蚂蚁生活在离散的状态，具有一定的记忆能力，更新信息素依赖特定的问题等。关于人工蚁以下几个概念：

(1)信息方格：信息方格是指蚂蚁观察到和移动的距离范围。蚂蚁有一个参数为速度半径的方格世界--即蚂蚁观察到的范围和移动的距离范围，这个速度半径一般是3cm，那么它观察和移动的范围就是3\*3cm2的信息方格。

(2)环境：蚂蚁所在的环境包含障碍物、别的蚂蚁、信息素等。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让信息素消失(避免算法因局部信息素强度过大而陷入局部最优和停滞)。

(3)信息素：信息素是蚂蚁在寻找食物和巢穴时沿途洒下的一种化学物质，能被其它经过此路径的蚂蚁感知，并指导它们的移动方向。路径上的信息素越强，蚂蚁超此路径移动的概率越大。信息素以一定的速率消失。

(4)觅食规则：蚂蚁在感知范围内寻找食物，若有，便直接朝食物方向移动；若无，朝信息素最多的方向移动，并以小概率犯错误(并不一定朝信息素最多的方向移动)。蚂蚁的小概率错误也能有效的避免局部最优和进行“创新”，找到新的最优解。

(5)移动规则：移动时朝信息素最多的方向前进，若没有信息素的指引，按惯性或者随机选择移动方向，并伴有随机的小扰动。蚂蚁移动过程中会记住走过的点，尽量避开重复经过这些已经走过的点。

(6)避障规则：遇到障碍时随机选择移动方向，但若有信息素指引时，按照觅食规则选择路径。蚂蚁按照移动的规则能动态地找到新的最短路径。

(7)播撒信息素：蚂蚁在经过的路径上播撒信息素，并且信息素以一定的概率消失。

根据这几条规则，蚂蚁之间并没有直接的关系，但是每只蚂蚁都和环境发生交互，而通过信息素这个纽带，实际上把各个蚂蚁之间关联起来了。比如，当一只蚂蚁找到了食物，它并没有直接告诉其它蚂蚁哪里有食物，而是向环境播撒信息素，当其它的蚂蚁经过它附近的时候，就会感觉到信息素的存在，进而根据信息素的指引找到了食物。

### 3.蚁群算法的流程与特点

蚂蚁在循环过程中在经过的路径上释放信息素，概率地选择下一个移动的方向或目标，这个概率是信息素和启发式因子的函数。蚂蚁在一次循环中不允许访问已经访问过的目标。开始时对所有蚂蚁进行初始化(如设定蚂蚁数量、蚂蚁最大信息素量、启发式因子系数和信息素挥发系数等)，之后蚂蚁开始搜索过程，根据目标函数对每只蚂蚁的适应度作出评价，并判断是否满足终止条件，若满足，程序结束，若不满足，计时器增加一个单位时间，更新信息素。

简单蚁群算法的流程如图8-3所示：



图8-3 简单蚁群算法流程图

根据具体的算法不同，以及要解决的问题不同，算法的流程也有一定程度的差异和变化，算法中参数的确定及公式也要根据具体的问题而定。

蚁群算法主要有以下特点：

(1)正反馈算法。蚂蚁在路径上播撒信息素，经过的蚂蚁越多，信息素的浓度越高，后来的蚂蚁选择该路径的概率也越高，从而又增加了该路径的信息素的强度，形成自催化过程，这种利用信息作为正反馈使系统中较强的解自增强，使问题的解向着全局最优的方向不断进化，最终有效地获得相对最优的解。

(2)并发性算法。搜索过程不是从一点出发，而是同时从多个点同时进行，在问题空间中同时构造问题的多个解。

(3)较强的鲁棒性，易与其它方法融合。单个蚂蚁的行为不会对系统找到最优解产生影响，而且当环境改变时蚁群同样可以找到最优解。改进的算法也可以应用于其它不同类型的问题，还可以与遗传算法、免疫算法等结合。

(4)概率型全局搜索方法。这种非确定性使算法能有更多的机会求解全局最优解。

(5)不依赖严格的数学性质(如函数的连续性、可导性及目标函数和约束函数的精确数学描述)。

(6)搜索时间长，易出现停滞现象。当群体规模较大时，很难再较短的时间内收敛于最优解，要得到好解，需要较长一段时间。搜索进行到一定程度时，可能发生局部最优的情况。信息素的蒸发和小概率的“错误”都可以使在一定程度上防止局部最优和停滞现象。

**4.发展及现状**

Dorigo提出的蚁群优化算法---蚂蚁系统（Ant System, AS）很好地解决了计算机算法学中经典的旅行商问题（TSP）。基本的蚁群算法在提出后一直不断地发展和完善，并在TSP以及其他问题求解中进一步得到了验证。这些AS改进版本的一个共同点就是增强了蚂蚁搜索过程中对最优解的探索能力，它们之间的差异仅在于搜索控制策略方面。而且，取得了最佳结果的ACO是通过引入局部搜索算法实现的，这实际上是一些结合了标准局域搜索算法的混合型概率搜索算法，有利于提高蚁群各级系统在优化问题中的求解质量。

最初提出的AS有三种版本：Ant-density、Ant-quantity和Ant-cycle。在Ant-density和Ant-quantity中蚂蚁在两个位置节点间每移动一次后即更新信息素，而在Ant-cycle中当所有的蚂蚁都完成了自己的行程后才对信息素进行更新，而且每个蚂蚁所释放的信息素被表达为反映相应行程质量的函数。通过与其它各种通用的启发式算法相比，在不大于75城市的TSP中，这三种基本算法的求解能力还是比较理想的，但是当问题规模扩展时，AS的解题能力大幅度下降。

因此，其后的ACO研究工作主要都集中于AS性能的改进方面。

### 4.1.标准蚁群算法

最初的蚁群算法称为蚂蚁系统(Ant System，简称AS)，在蚂蚁系统中，状态转移概率和信息素更新公式分别为式(8-1)和式(8-2)：

 (8-1)

其中，为蚂蚁k从城市i转移到城市j的概率；、分别为信息素和启发式因子的相对重要程度；为边(i, j)上的信息素量；是启发式因子；是蚂蚁k下步允许选择的城市。

 (8-2)

其中，为信息素蒸发系数，0<<1；为第k只蚂蚁在本次迭代中留在边(i, j)上的信息素量；为一正常数；为第k只蚂蚁在本次周游中的路径长度。

M.Dorigo在AS算法中提出了3种模型：ant-cycle，ant-quantity和ant-density。三者的区别在于计算第k只蚂蚁在迭代中留在边(i，j)上的信息素量：

ant-cycle：  (8-3)

ant-quantity： (8-4)

ant-density： (8-5)

AS算法实际上是正反馈和启发式算法相结合的一种算法，因为它不仅利用了路径上的信息素，还用到了城市间距的倒数作为启发式因子。试验结果表明，ant-cycle模型比其它两个模型有更好的性能，它利用全局信息素更新，而ant-quantity和ant-density利用局部信息素更新。M.Dorigo在求解30个城市的TSP时发现当={0.5,1},={1,2,3,4,5}时，AS算法总能收敛到最优解，并且当蚂蚁数m接近城市数n时，算法有较好的性能。

### 4.2.改进的蚁群算法

针对蚁群算法的不足(如复杂度高，容易出现停滞现象等)，大批学者围绕如何改进蚁群算法，提高算法的性能做了大量工作。其中应用广泛且具有代表性的改进蚁群算法主要有带精英策略的蚂蚁系统、基于排序的蚂蚁系统、蚁群系统及最大--最小蚂蚁系统。改进的蚁群算法主要改进信息素更新的策略和状态转移规则。

(1)带精英策略的蚂蚁系统(Ant System with Elitist, ASelite)，又称最优解保留策略蚂蚁系统，是最早的改进蚂蚁系统。因为在某些方面它类似于遗传算法中采用的精英策略，因此把它称作带精英策略的蚂蚁系统。

通过使用最优蚂蚁可以提高蚂蚁系统中解的质量。每次迭代完成之后，全局最优解得到更进一步的利用，即在对信息素的轨迹进行更新时，假定有许多蚂蚁选择了该路径。与AS算法相比，该算法在信息素更新时加强了对全局最优解的利用，它的信息素更新策略为：

 (8-6)

其中为蚂蚁在边(I, j)上增加的信息素量；为最优蚂蚁数；为全局最优解。

实验结果显示最优蚂蚁数有一定的范围，当最优蚂蚁数小于该范围时，随着最优蚂蚁数的增加，算法发现较好解的能力增加，且缩短了发现较好解的时间，但蚂蚁数超过一定范围时，算法的性能会随着蚂蚁数的增加而降低。

(2)基于排序的蚂蚁系统(rank-based version of ant system, Asrank)。它是将遗传算法中排序的概念扩展应用到蚂蚁系统中得到的。基本思想是：先是根据适应度对种群进行分类，然后被选择的概率取决于个体的排序。适应度越高，个体在种群中的排名越靠前，被选择概率越大。或在每次迭代完成后，将蚂蚁所经过路径按从小到大的顺序排列，并根据路径长度赋予不同的权重，路径长度越短权重越大。全局最优解的权重为w，第r个最优解的权重为max{0，w-r}。其各路径上信息素的更新策略如下：

 (8-7)

(3)蚁群系统(Ant Colony System，ACS)。它是AS算法的改进型版本，它与AS算法的主要区别为：①在选择下一座城市时，ACS算法更多地利用当前的较好解；②只在全局最优解所属的边上增加信息素；③当蚂蚁从城市m爬行到城市n时，边(m，n)上的信息素将会适当的减少。

在ACS算法中，蚂蚁使用伪随机比率选择规则选择下一座城市，即对于位于城市i的蚂蚁k，以概率q0移动到城市l，其中l为使达到最大的城市。该选择方式使得蚂蚁将以概率q0将最大可能的城市选入蚂蚁所构造的解，除此之外，蚂蚁以(1- q0)的概率选择下一座城市j，状态转移公式为：

 (8-8)

在选择下一座城市之前随机生成q，如果q的值小于等于常数q0，则从城市i到所有可行的城市中找出最大的城市，即为下一个要选择的城市；如果随机数q大于q0，选择下一座城市的公式为：

 (8-9)

其局部信息素更新公式为：

 (8-10)

全局信息素更新公式为：

 (8-11)

其中，为常数；(0, 1)为可调参数。

ACS算法在大多数情况下要优于或者性能相当于AS算法、模拟退火、进化规划、遗传算法和模拟-遗传算法，在解决非对称TSP时，ACS算法更具优势。

(4)最大-最小蚂蚁系统(max-min ant system, MMAS)，它是目前求解TSP和二次分配问题（quadratic assignment problem, QAP）等问题最好的蚁群算法模型。与其他寻优算法相比，仍然属于最好的方法之一。MMAS直接来源于蚁群算法，主要改进了三个方面：①每次迭代后，只有最优解所属路径上的信息被更新；②将各路径的信息素浓度限制于，超出范围的被强行设定为或者，可以有效的避免算法过早收敛于局部最优解；③初始时刻，各路径上的信息素浓度设为，取较小的值。所有蚂蚁完成一次迭代后，更新所有路径上的信息素：

 (8-12)

更新的路径可以是全局最优解，也可以是本次迭代的最优解。实践证明逐渐增加全局最优解的使用频率，会使该算法获得较好的性能。

## 蚁群算法实例求解步骤

**1.车间调度问题**

调度问题（scheduling），又称排序问题，指将若干工件（job）在一些机器上进行加工，合理安排机器和工件，以使目标函数最优的过程。

**算例描述：**

已知：有n个有两道工序的工件{J1,J2,......,Jn}，需要在两台机器M1，M2上加工；不同工件之间无顺序约束，工件在两台机器上加工顺序相同；

工件约束：每个工件上有两道工序，均需且只可被两台机器加工一次，且只有在工件到达机器后才可以被加工，工序开始则不能间断；

机器约束：每台机器某一时刻只能执行一个工件的一道工序，而且执行过程是非抢占的；机器不发生故障。

目标：给出调度方案，使调度完工时间最小。

表8-1各工件工序加工时间如下表所示

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 工件 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 工序1 | 10 | 13 | 15 | 12 | 14 | 9 | 16 | 17 | 15 |
| 工序2 | 14 | 17 | 13 | 12 | 10 | 13 | 18 | 9 | 6 |

 (8-13)

其中，dij是第i个工件的第j道工序。

蚁群算法求解车间调度过程：

1. 工件的工序1时间矩阵视为一般意义上的距离矩阵（维数为9\*1矩阵）。
2. 初始化参数(α,β,ρ)。在进行迭代寻优前设置初始参数，合理的参数有利于更快的找到较优的迭代结果。
3. 迭代寻优。这个过程是整个算法的核心，用通俗语言可理解为每只蚂蚁各自从某个城市按照一定的几率转移到下一个城市，一般距离比较近的城市转移的概率更大，当所有城市均被搜索并且返回原点即完成了一只蚂蚁的寻优过程，所有蚂蚁完成自己搜索的城市路线后即完成了一次迭代，并更新信息素。
4. 在达到最大迭代次数后，对结果进行分析判断是否达到了要求精度，与其他方法进行比较，对蚁群算法性能进行评价，或改变初始参数观察对结果的影响，尝试改进寻优机制，找到更好的结果或者加快迭代速度。

蚁群算法程序实现的步骤如图8-6。通过程序实现可以得出结果，在最大迭代次数N=100，初始蚂蚁数量为5的情况下，最短加工时间:135；最短路径:1 9 6 4 3 2 5 7 8。下图8-7和8-8为程序运行的结果图，从图中可以看出，在解决本例中的车间调度问题时，蚁群算法能很快收敛到最短等待时间。



图8-4 基于蚁群算法的车间调度求解步骤

实验结果如下：



图8-5 蚁群算法求解车间调度甘特图(100次迭代)



图8-6 蚁群算法求解车间调度时间收敛图(100次迭代)

## 2. 零件配送路径规划

[路径](https://baike.baidu.com/item/%E8%B7%AF%E5%BE%84" \t "https://baike.baidu.com/item/%E8%B7%AF%E5%BE%84%E8%A7%84%E5%88%92/_blank)规划是运动规划的主要研究内容之一。运动规划由路径规划和轨迹规划组成，连接起点位置和终点位置的序列点或曲线称之为路径，构成路径的策略称之为路径规划。目标是是的工作站需求得到满足，并能在一定的约束下，达到注入路程最短、成本最小、耗费时间最少等目的。零件配送路径规划优化问题可以描述如下：



零件中心 路线方向 作业单位

图8-7 . 基本问题示意图

路径问题是网络优化问题中最基本的问题之一，由于其应用的广泛性和经济上的重大价值，自1959年提出以来一直受到国内外学者的广泛关注。VPR问题主要包含以下要素：配送中心、车辆、货物、客户、运输网络、约束条件和目标函数等。

**2.1基本问题与模型建立**

某零件配送中心周边有19个待供货区域，其坐标布置如下表，试规划配送点与配送中心之间的路径使在满足配送要求的情况下总路径最短。

表4-1 配送中心和配送点坐标

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地点 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 横坐标 | 0 | 0 | 0 | -2 | -3 | 3 | -4 | -4 | 1 | 1 |
| 纵坐标 | 0 | -1 | 3 | -2 | -3 | -1 | 0 | -1 | -2 | -1 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地点 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 横坐标 | 1 | 3 | -3 | 2 | 1 | 2 | 2 | 1 | -3 | -1 |
| 纵坐标 | 3 | 4 | 0 | 0 | -3 | -1 | 1 | -4 | 2 | -1 |

目标函数：

 （8-14）

约束条件：

 （8-15）

 （8-16）

式（8-14）为目标函数，由于配送中心运输成本与行驶路径相关，行驶距离越长运输成本越高，因此选取加盟车总运输距离最短为目标函数。

式（8-15）为小车装载能力约束，同一配送车辆配送总体积不能超过小车额定载重W。Ui是配送点i需要的货物量。

式（8-16）为决策变量，当i和j两个配送点有路径规划时，取值1，否则为0。

**2.2算法流程**

****

根据载重约束选择下一个工作站或返回配送中心

在所有收货点均有规划路线经过后，返回配送中心

图8-8 算法设计流程图

根据算法设计，结果如图8-9和8-10所示：

路径顺序：1 2 15 18 9 1 20 4 5 8 13 1 3 11 12 17 14 6 16 10 1 19 7 1

最短路径长度：43.6695



图8-9 最终路线设计图

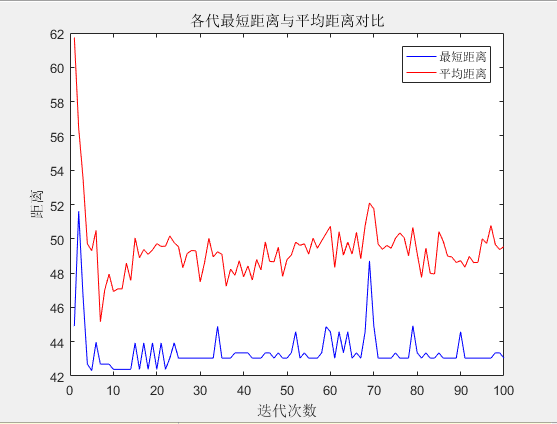


图8-10 距离变化图

## 应用及总结

本章首先阐述了蚁群算法的基本原理，然后，以传统的TSP问题为例，介绍了蚁群算法求解TSP的方法。以TSP为基础，增加适当的约束条件，介绍了车辆调度的问题，通过蚁群算法在物流中心与零售商之间规划最短路径，通过实例介绍了求解方法和步骤。在应用上，目前已经实现的有光网络的智能管理、计算机网络路由、聚类（蚁群蚁卵分类）等等。