

进化优化算法

基于仿生和种群的计算机智能方法



第七课：蚁群优化算法
(Ant Colony Optimization)

第三次作业

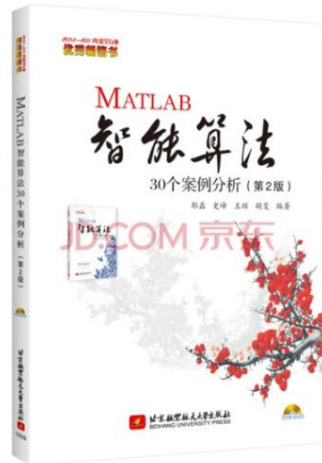
- 作业内容：TSP旅行商问题
- 交作业时间：2024年12月2日之前
- 作业要求：
 1. 采用蚁群优化算法
 2. 编程语言不限
 3. 电子版作业报告一份：包含问题描述、编程思路及方法、源代码（添加适当的注释）
 4. 源代码一份，50次运行测试程序效率及准确性
 5. 程序自己编写，不要拷贝或抄袭现有的程序
 6. 每位同学提交一份作业
- 作业内容：本学期大作业的标题和优化问题简单描述（采用多目标优化算法额外加分、单目标优化问题采用与三次作业不同的算法额外加分）

14个城市旅行商问题

本案例以 14 个城市为例,假定 14 个城市的位置坐标如表 4-1 所列。寻找出一条最短的遍历 14 个城市的路径。

表 4-1 14 个城市的位置坐标

城市编号	X 坐标	Y 坐标	城市编号	X 坐标	Y 坐标
1	16.47	96.10	8	17.20	96.29
2	16.47	94.44	9	16.30	97.38
3	20.09	92.54	10	14.05	98.12
4	22.39	93.37	11	16.53	97.38
5	25.23	97.24	12	21.52	95.59
6	22.00	96.05	13	19.41	97.13
7	20.47	97.02	14	20.09	92.55



14个城市旅行商问题

优化前的一个随机路线轨迹图如图 4-2 所示。

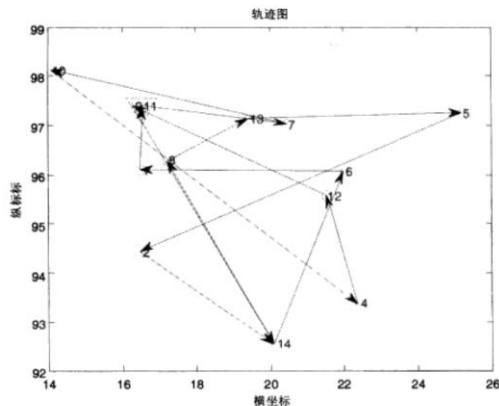


图 4-2 随机路线图

随机路线为 11→7→10→4→12→9→14→8→13→5→2→3→6→1→11。

总距离为 71.114 4。

优化后的路线图如图 4-3 所示。

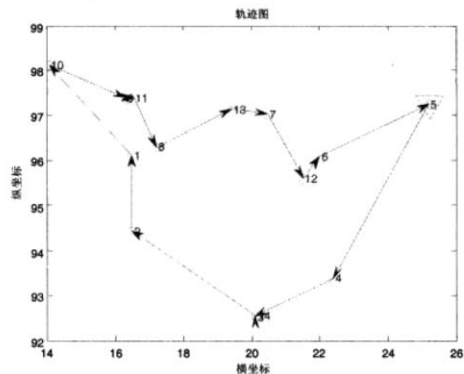


图 4-3 最优解路线图

最优解路线为 5→4→3→14→2→1→10→9→11→8→13→7→12→6→5。

总距离为 29.340 5。

-
- 蚂蚁觅食的秘密
 - 蚁群优化算法理论
 - 蚁群优化算法的数学模型及算法流程
 - 举例说明蚁群优化算法的工作机理
 - 蚁群优化算法的改进
 - 蚁群优化算法的参数设置及影响

蚂蚁的基本习性

- 蚂蚁是一种最古老的社会性昆虫，它的起源可追溯到1亿年前，大约与恐龙同一时代
- 楚汉相争之时，汉高祖刘邦的谋士张良用饴糖作为诱饵，使蚂蚁闻糖而聚，组成“霸王自刎乌江”，项羽以为是天意，“天之亡我，非战之罪也”，拔剑自杀而死。流传下“汉家天下，蚂蚁助成”的故事



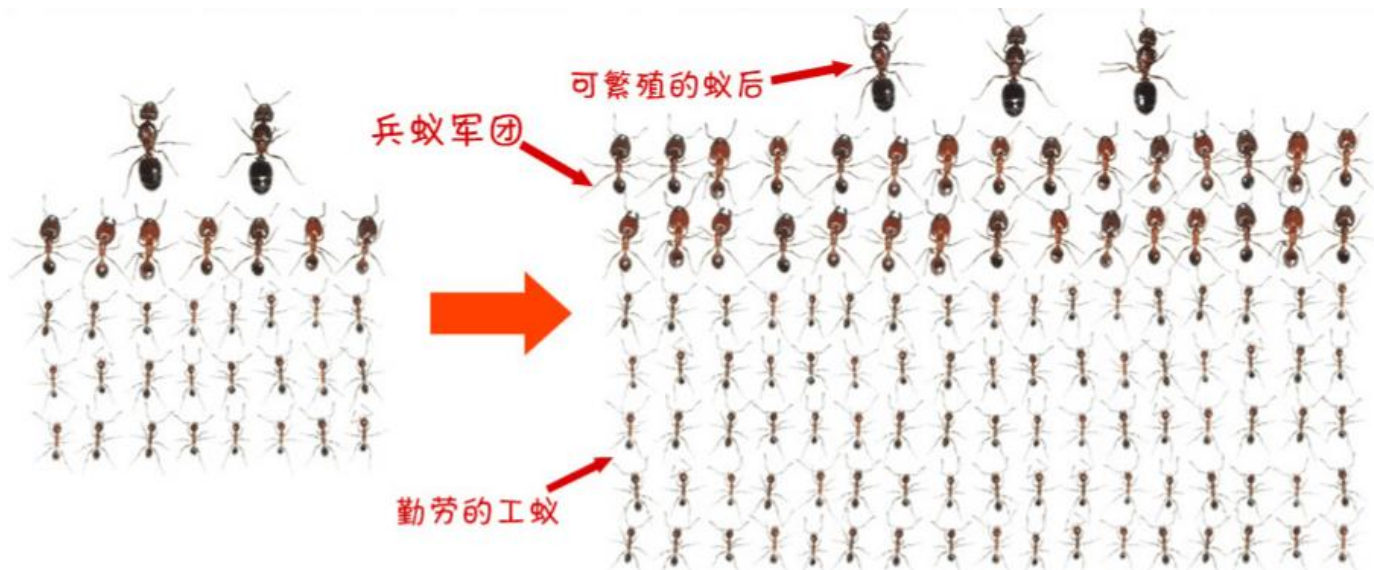
蚂蚁的基本习性

- 蚂蚁属膜翅类，蚁总科，已知360属，约9000种，估计应有12000~15000种。繁殖能力超强，世界上每诞生40个婴儿，诞生7亿只蚂蚁
- 数以百万亿计的蚂蚁悄悄布满了我们的星球，像人类一样，蚂蚁占据了几乎每一片适于居住的土地，只有永远雪封的南北两极未曾被其涉足（1%的生物量，1/3的动物量）。全球总数大约 2×10^{16} 只



蚂蚁的基本习性

- 蚂蚁虽然有成千上万中，但无一种是独居的，都是群体生活，建立了自己独特的蚂蚁社会
- 地球上有很多动物都具有一定的社会性，但是蚂蚁绝对是世界上社会性最强的种类



蚂蚁的基本习性

- 它们具有人类都无法比拟的“群体意识”，这种意识几乎达到了一种极端的境界。蚂蚁没有自己的自主意识，它们所做的一切完全服务于蚁群
- 在蚂蚁群体中，每一只蚂蚁个体，不论分工和职业，都是为了群体的利益。这种无私的奉献和牺牲精神，世界上任何一种生物都无法企及，所以在某种程度上，蚂蚁才是世界上进化得**最完美**的生物



蚂蚁群落的特点

➤ 单只蚂蚁能力有限

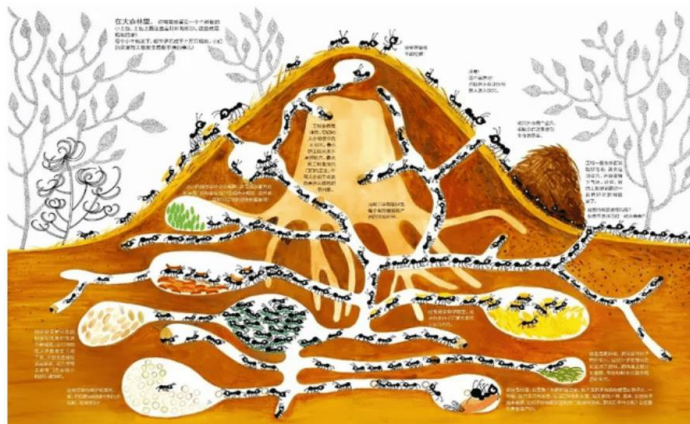
- ✓ 视力功能发育不完全，只能分辨出十几厘米以内物体的简单形状
- ✓ 视觉与听觉信号沟通能力有限
- ✓ 无法单独完成复杂的任务



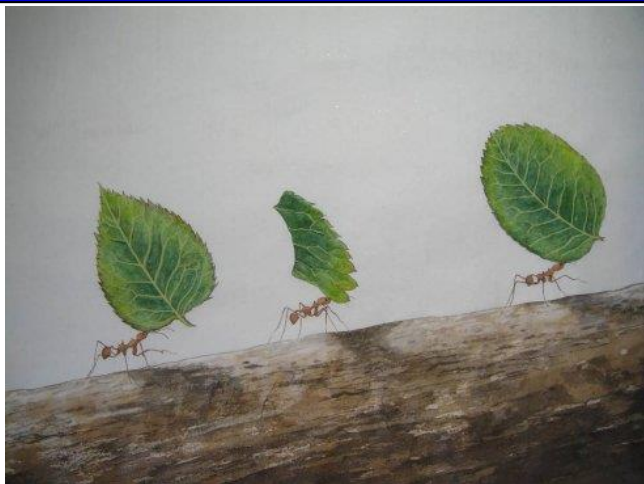
蚂蚁群落的特点

➤ 蚂蚁群落具有非凡的群体智慧

- ✓ 穴巢的建造与防卫
- ✓ 穴巢的温度控制
- ✓ 构建桥梁
- ✓ 合作搬运大型物品
- ✓ 食物分类
- ✓ 寻找食物来源（蚂蚁觅食）

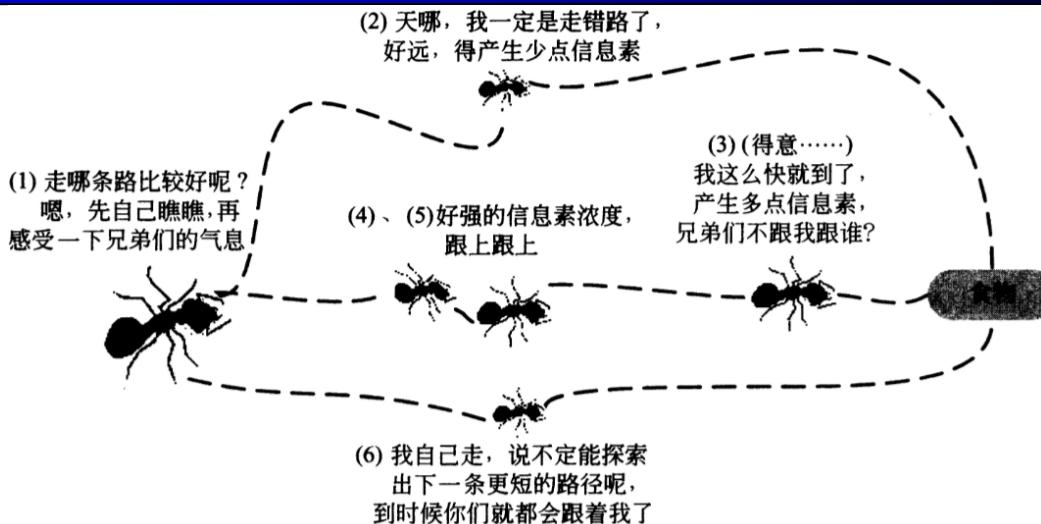


蚂蚁觅食的秘密



- 蚂蚁没有发育完全的视觉感知系统，甚至很多种类完全没有视觉，它们在寻找食物的过程中是如何选择路径的？
- 如何找到通往食物源的最短路径？在周围环境发生变化后，如何自适应地探索新的最佳路径
- 蚂蚁往往像军队般有纪律、有秩序地搬运食物，它们通过什么方式进行群体间的交流协作呢？

蚂蚁的协同工作机制



蚁群根据信息素觅食的过程

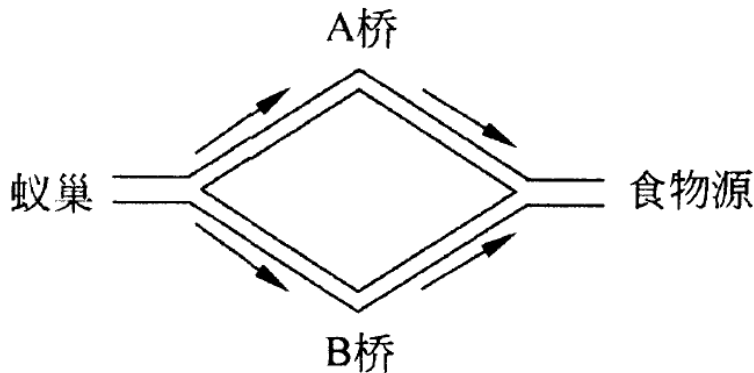
- 蚂蚁的协同工作机制通过改变周围环境实现
- 蚂蚁之间通过分泌的化学物质（信息素）进行沟通交流
- 蚂蚁的社会性行为依赖于信息素
 - ✓ 信息素可以用于标记路径，以便其他蚂蚁跟随
 - ✓ 在食物源和巢穴之间的路径上释放信息素

蚁群觅食行为与觅食策略

*对称二元桥实验如图所示，目的是让一些蚂蚁从蚁巢处出发，分别通过A桥、B桥到达食物源。

*设起初两个桥上都没有信息素，走两个分支蚂蚁的概率相同。

*实验中有意选择A桥的蚂蚁数多于B桥，由于蚂蚁在行进中要释放信息素，因此A桥的信息素多于B桥，从而使更多蚂蚁走A桥。



蚁群觅食行为与觅食策略

Deneubourg开发了一个信息素模型如下

设 A_i 和 B_i 是第 i 只蚂蚁过桥后已经走过A桥和B桥的蚂蚁数，

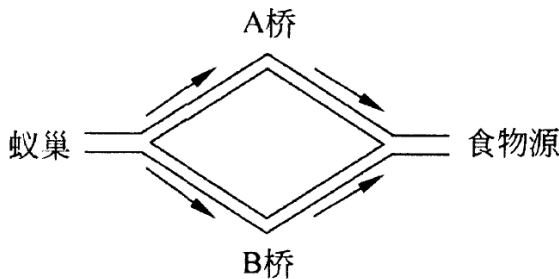
第 $i+1$ 只蚂蚁选择A桥（或B桥）的概率为

$$P_A = \frac{(A_i + K)^n}{(A_i + K)^n + (B_i + K)^n} = 1 - P_B$$

其中， n 及 K 是实验确定的参数

这个模型没有考虑信息素的挥发， K 和 n 的值通常为 $K \approx 20$ $n \approx 2$

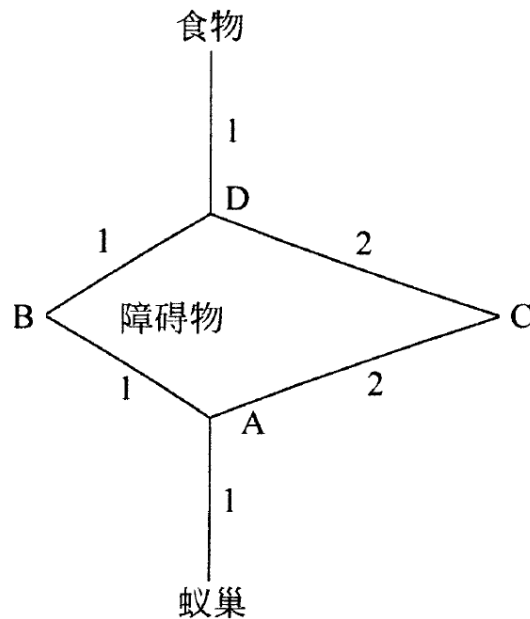
上式表明，如果A桥的蚂蚁越多，选择A 桥的概率越高



蚁群觅食行为与觅食策略

不对称二元桥实验如图所示，隐含的假设：

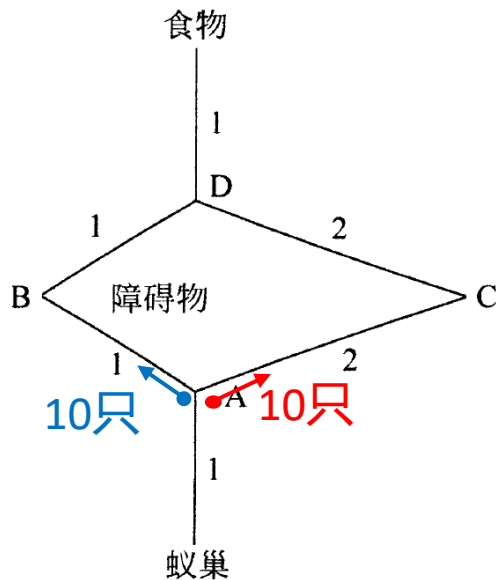
- (1) 每只蚂蚁移动速度相等；
- (2) 每只蚂蚁在单位长度路径上留下的信息素的数量相等；
- (3) 在正反两个移动方向都会留下信息素



蚁群觅食行为与觅食策略

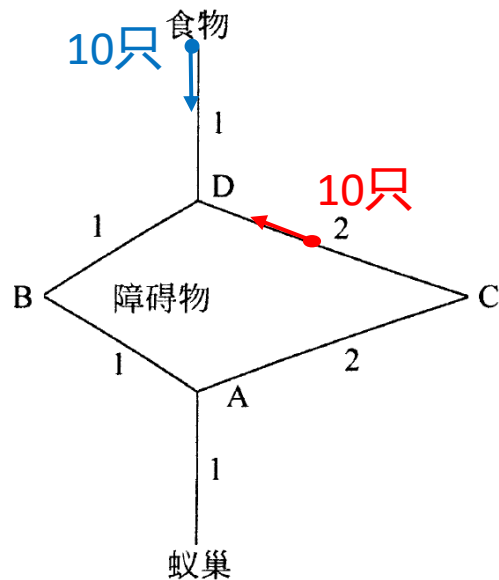
在 $t = 0$ 时刻，第一组有20只蚂蚁从蚁巢出发

在 $t = 1$ 时刻,20只蚂蚁移动到A，由于所有的道路上都没有信息素，它们以相同的概率选择左侧（ ABD ）路径或右侧（ ACD ）路径。因此，有10只选择左侧（ ABD ），10只走右侧（ ACD ）



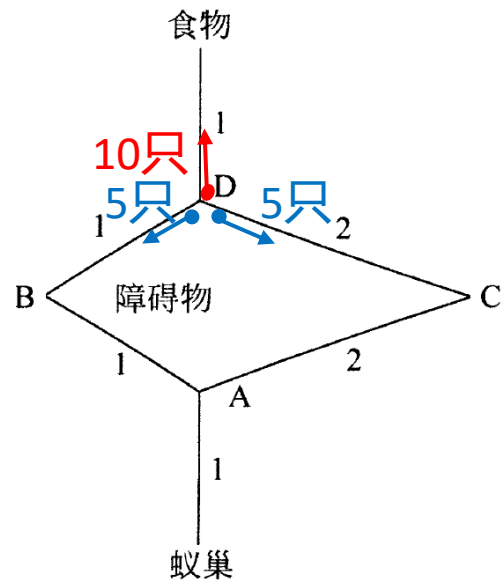
蚁群觅食行为与觅食策略

在 $t = 4$ 时刻，走左侧（ ABD ）路径的10只蚂蚁将折回，
此时走右侧（ ACD ）路径蚂蚁到达 CD 中点处



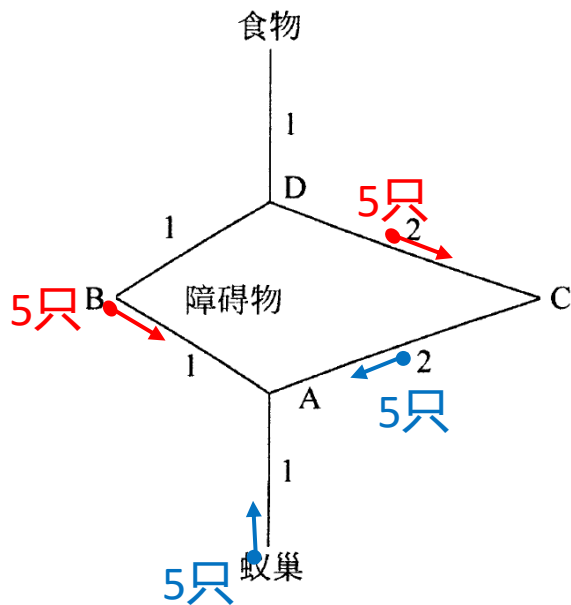
蚁群觅食行为与觅食策略

在 $t = 5$ 时刻，两组蚂蚁将在 D 点相遇。此时 BD 上的单位长度信息素数量（浓度）和 CD 上的相同，因为各有10 只蚂蚁选择了相应的路径，从而有5只返回的蚂蚁选择 BD 而另5 只将选择 CD ，走右侧（ ACD ）路径的蚂蚁继续向食物方向移动



蚁群觅食行为与觅食策略

在 $t = 8$ 时刻，前5只蚂蚁将返回蚁巢，此时在 AC 中点处、 CD 中点处及 B 点上各有5只蚂蚁



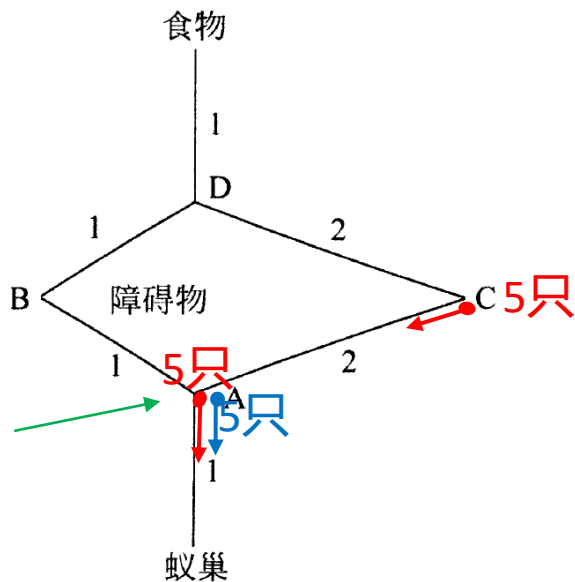
蚁群觅食行为与觅食策略

*在 $t = 9$ 时刻，前5只蚂蚁又返回A，并且再次面对往左还是往右的路径选择。这时，AB上的轨迹数是20而AC上是15，因此将有较为多数的蚂蚁选择往左，从而增强了该路线的信息素

*随着上述过程的继续，两条路径上的信息素浓度的差距将越来越大，直至绝大多数蚂蚁都选择了最短的路径。

*这就是蚂蚁从蚁巢到食物源的觅食过程中能够找到最优路径的原理

t=9时刻，5只已返回蚁巢的蚂蚁再次去寻找食物在A点进行路径选择，另外15只蚂蚁的位置如图所示



通往食物源的最优路径

隐含的假设：（1）每只蚂蚁移动速度相等；（2）每只蚂蚁在单位长度路径上留下的信息素的数量相等；（3）在正反两个移动方向都会留下信息素

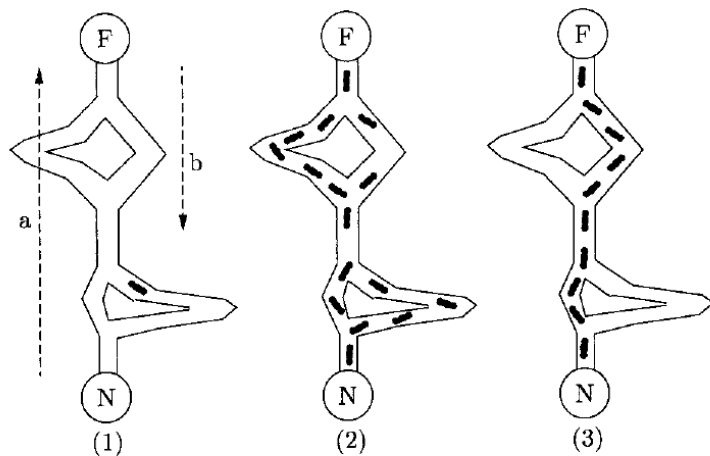


图 10.1 蚂蚁留下并跟随信息素。(1) 第一只蚂蚁朝着 a 所示的方向移动, 找到食物源 F, 并返回由 b 所示方向上的蚁巢 N, 在它的移动过程中会留下信息素的痕迹。(2) 蚂蚁沿着从 N 到 F 的 4 条可能路径中的一条移动, 但信息素强化会让最短路径更具吸引力。(3) 蚂蚁倾向于沿着信息素最多的路径移动, 继续强化这条路径的吸引力, 同时在较长路径上的信息素会挥发。(本图由 Johann Dréo 绘制, 复制自 http://en.wikipedia.org/wiki/File:Aco_branches.svg, 依照 GNU 自由文档协议的规定发布。)

蚂蚁觅食的秘密

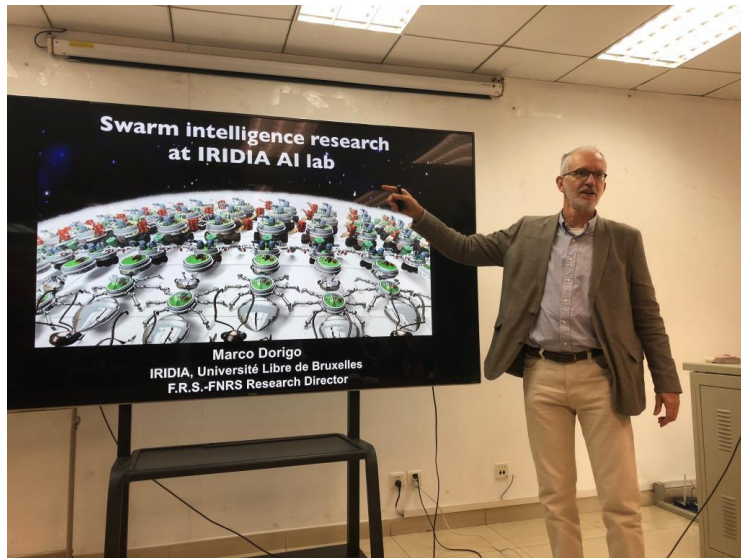
- 蚂蚁会在经过的路径上留下信息素
- 相对于信息素水平较低的路径，蚂蚁选择信息素水平高的路径**概率**更大----**多样性**（创造能力）
- 选择某条路径的蚂蚁的数量越多，该路径的信息素水平越高
- 蚂蚁的这种选择路径过程被称之为**自催化行为** (Autocatalytic Behavior)，越多蚂蚁选择的路径变得越来越有吸引力----**正反馈**（科学强化能力）
- **多样性**保证了蚂蚁在觅食的时候不致走进死胡同而无限循环
- **正反馈**机制则保证了相对优良的信息能够被保存下来

□ **这就是蚁群优化算法的灵感来源**

-
- 蚂蚁觅食的秘密
 - 蚁群优化算法理论
 - 蚁群优化算法的数学模型及算法流程
 - 举例说明蚁群优化算法的工作机理
 - 蚁群优化算法的改进
 - 蚁群优化算法的参数设置及影响

蚁群优化算法

- 意大利学者Marco Dorigo马可·多里戈1992年在其博士论文中提出，现任比利时布鲁塞尔自由大学教授，依托其所提出的蚁群算法书写了仿生智能计算领域中的一段传奇
- 蚁群优化算法是从真实蚂蚁群落觅食过程中寻找路径的行为获得灵感，是一种寻找优化路径的概率型方法



蚁群优化算法

➤ 1996年Marco Dorigo等在《IEEE Transactions on Systems, Man, and Cybernetics - Part B》发表了“The Ant System: Optimization by a colony of cooperating agents”一文，奠定了蚁群优化算法的基础

IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol.26, No.1, 1996, pp.1-13 1

The Ant System: Optimization by a colony of cooperating agents

Marco Dorigo^{*}, Member, IEEE, Vittorio Maniezzo^{**}, and Alberto Colomi[†]

^{*} IRIDIA, Université Libre de Bruxelles, Avenue Franklin Roosevelt 50, CP 194/6, 1050 Brussels, Belgium.
Email: mdorigo@ulb.ac.be URL: <http://iridia.ulb.ac.be/dorigo@dorigo.tnln>

^{**} Politecnico di Milano Artificial Intelligence and Robotics Project, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy.

[†] Dipartimento di Scienze dell'Informazione, Università di Bologna, Via Sacchi 3, 47023 Cesena, Italy. Email: maniezzo@cse.unibo.it

[‡] Centro di Teoria dei Sistemi del CNR, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133 Milano, Italy. Email: colomi@cet.polimi.it

Abstract

An analogy with the way ant colonies function has suggested the definition of a new computational paradigm, which we call *Ant System*. We propose it as a viable new approach to stochastic combinatorial optimization. The main characteristics of this model are positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process. We apply the proposed methodology to the classical Traveling Salesman Problem (TSP), and report simulation results. We also discuss parameter selection and the early setups of the model, and compare it with tabu search and simulated annealing using TSP. To demonstrate the robustness of the approach, we show how the Ant System (AS) can be applied to other optimization problems like the asymmetric traveling salesman, the quadratic assignment and the job-shop scheduling. Finally we discuss the salient characteristics – global data structure revision, distributed communication and probabilistic transitions of the AS.

Copyright notice

Copyright 1996 Institute of Electrical and Electronics Engineers. Published in the IEEE Transactions on Systems, Man, and Cybernetics, Part-B, Vol.26, No.1, pp.1-13.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of [company's] products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to info.pub.permission@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Ant system: optimization by a colony of cooperating agents

来自 ACM | 喜欢 0 | 阅读量: 1967

作者: M Dorigo, V Maniezzo

摘要: An analogy with the way ant colonies function has suggested the definition of a new computational paradigm, which we call ant system (AS). We propose it as a viable new approach to stochastic combinatorial optimization. The main characteristics of this model are positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process. We apply the proposed methodology to the classical traveling salesman problem (TSP), and report simulation results. We also discuss parameter selection and the early setups of the model, and compare it with tabu search and simulated annealing using TSP. To demonstrate the robustness of the approach, we show how the ant system (AS) can be applied to other optimization problems like the asymmetric traveling salesman, the quadratic assignment and the job-shop scheduling. Finally we discuss the salient characteristics. 收起

关键词: feedback optimisation robust control search problems simulated annealing stochastic programming travelling salesman problems ant colonies ant system computational paradigm

DOI: 10.1109/3477.484436

被引量: 2.0万

年份: 1996

☆ 收藏

<> 引用

批量引用

报错

分享

蚁群优化算法

- 2000年，Dorigo和Bonabeau等在国际顶级学术刊物Nature发表了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
- 2004年Marco Dorigo等人出版了专著《Ant Colony Optimization》，经过20多年的发展，蚁群算法在理论以及应用研究上已经得到巨大的进步

nature

[Explore content](#) ▾ [About the journal](#) ▾ [Publish with us](#) ▾

[nature](#) > [ecology](#) > article

Published: 05 July 2000

Inspiration for optimization from social insect behaviour

[E. Bonabeau](#)  [M. Dorigo](#) & [G. Theraulaz](#)

[Nature](#) 406, 39–42 (2000) | [Cite this article](#)

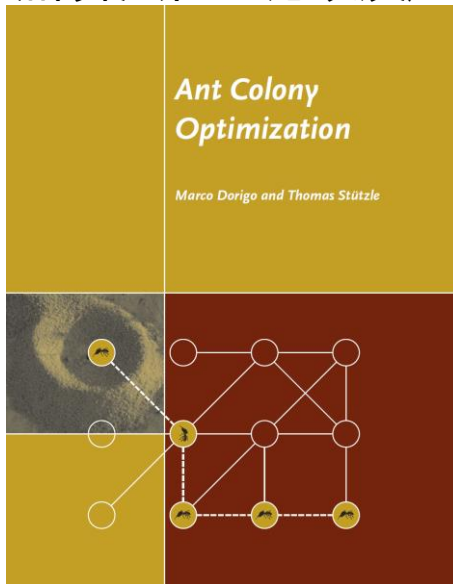
[4602](#) Accesses | [594](#) Citations | [17](#) Altmetric | [Metrics](#)

Abstract

Research in social insect behaviour has provided computer scientists with powerful methods for designing distributed control and optimization algorithms. These techniques are being applied successfully to a variety of scientific and engineering problems. In addition to achieving good performance on a wide spectrum of 'static' problems, such techniques tend to exhibit a high degree of flexibility and robustness in a dynamic environment.

Main

Ethologists use modelling to understand animal behaviour. Recent research in social insect behaviour suggests that models based on self-organization can help explain how complex colony-level behaviour emerges out of interactions among individual insects^{1,2,3}. Although the goal of a modeller is generally to understand the living, a model can in principle be explored beyond the biologically plausible. Although biology does not necessarily benefit from such an exploration, computer scientists and engineers have been able to transform models of social insect collective behaviour into useful optimization and control



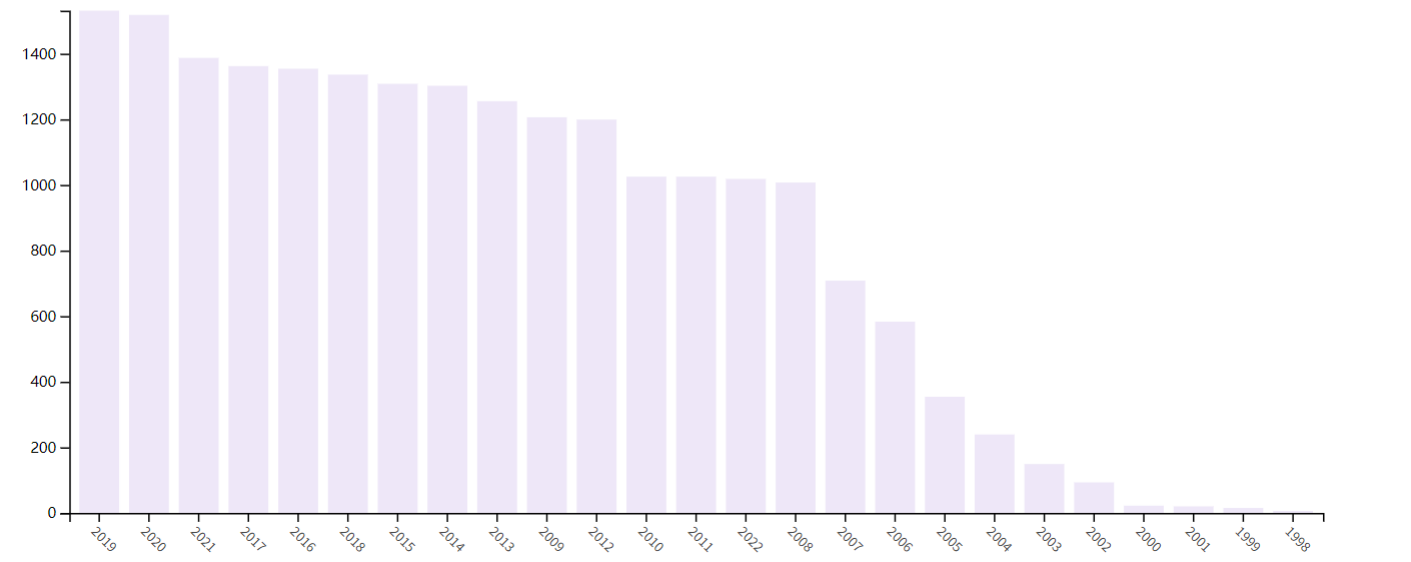
可视化数据:

柱状图

检索结果数:

25

下载



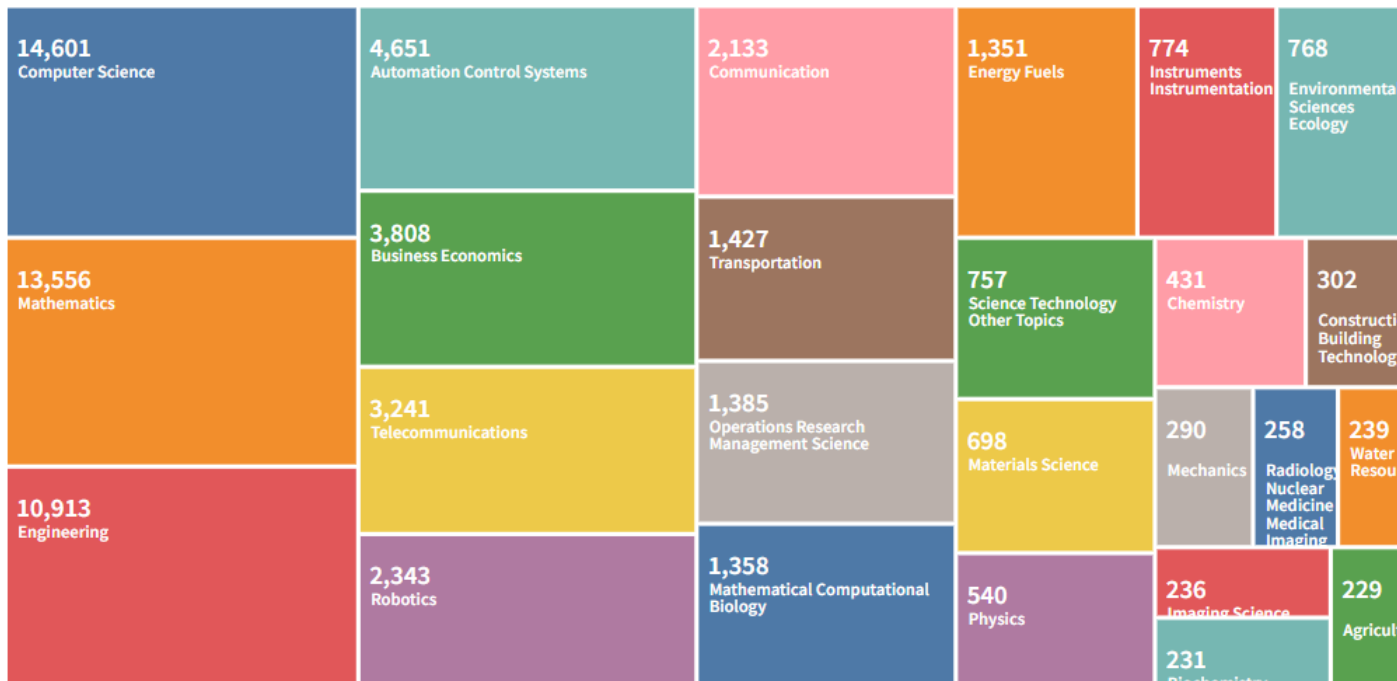
可视化数据:

检索结果数:

树状图

25

下载



蚁群优化算法的机制原理

➤ 蚁群优化算法是从蚂蚁觅食信息素沟通交流机制启发的优化算法，该算法基于如下基本假设：

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做出反应，也仅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而形成高度有序的群体行为

蚁群优化算法的机制原理

➤ 蚁群优化算法包含两个基本阶段：

1. 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，信息素的浓度越高，则该路径越容易被选择；时间越长，信息量会越小
2. 协作阶段：候选解之间通过信息交流，以期望获得性能更好的解

蚁群的抽象

➤ 蚂蚁个体的抽象：

- ✓ 为了能够从真实蚂蚁觅食行为中抽象出能够为算法借鉴的机理，需要将真实蚂蚁抽象成人工蚂蚁
- ✓ 人工蚂蚁能够完成所求问题解的基本构造过程，利用当前环境的信息依概率进行路径选择，从而能够完成源节点到目标节点的遍历
- ✓ 人工蚂蚁具备一定的信息交流能力，人工蚂蚁改变所经路径或节点上存储的数字信息，该信息就是算法中所定义的信息素

蚁群的抽象

➤问题空间的抽象：

- ✓问题的解空间一般是一个平面，这与真实蚂蚁觅食的三维空间不同，需要进行抽象
- ✓真实蚂蚁的爬行轨迹是连续的，计算机处理的是离散事件，因此必须将连续的平面离散化为点阵组成的离散平面

蚁群的抽象

➤ 寻找路径的抽象

- ✓ 可把蚂蚁觅食过程抽象为算法中解的搜索过程，将路径上的信息素抽象为问题所对应的边上的信息素轨迹
- ✓ 当人工蚂蚁处于图中某一节点时，通过感知与该节点相连的边上的信息素轨迹，并根据各条边上的浓度依概率选择将要爬行的边，从而到达另一节点。这样由初始节点到目标节点的一条路径就是所求问题的一个解

蚁群的抽象

➤ 启发因子的引入

- ✓ 根据所求问题空间的具体特性，在蚂蚁进行路径选择时，可以引入启发因子，给予初始的引导，这样能有效地减少算法的搜索时间

蚂蚁觅食现象

蚁群优化算法

蚁群	↔	搜索空间的一组有效解（表现为种群规模 m ）
觅食空间	↔	问题的搜索空间（表现为问题的规模、解的维数 n ）
信息素	↔	信息素浓度变量
蚁巢到食物的一条路径	↔	一个有效解
找到的最短路径	↔	问题的最优解

人工蚂蚁与真实蚂蚁的相同点

1. 都是蚁群相互协作的个体

- ✓ 与真实蚂蚁群一样，蚁群算法由一群人工蚂蚁组成，人工蚂蚁之间通过同步/异步协作来寻找问题的最优解
- ✓ 虽然单只人工蚂蚁可以构造出问题的解，但只有当多只人工蚂蚁通过相互协作，才能发现问题的最优（次优）解
- ✓ 人工蚂蚁个体间通过写/读问题的状态变量来进行协作

人工蚂蚁与真实蚂蚁的相同点

2. 都使用信息素的迹和蒸发机制

- ✓ 如真实蚂蚁一样，人工蚂蚁通过改变所访问过的问题的数字状态信息来进行间接的协作。在蚁群算法中，信息素是人工蚂蚁之间进行交流的唯一途径。这种通信方式在群体知识的利用上起到了至关重要的作用
- ✓ 蚁群算法还用到了蒸发机制，这一点对应于真实蚂蚁中信息素的蒸发现象。蒸发机制使蚁群逐渐忘记过去的历史，使后来的蚂蚁在搜索中较少受到过去较差解的影响，从而更好地指导蚂蚁的搜索方向

人工蚂蚁与真实蚂蚁的相同点

3. 搜索最短路径与局部移动

- ✓ 人工蚂蚁和真实蚂蚁具有相同的任务，即以局部移动的方式构造出从原点（蚁巢）到目的点（食物源）之间的最短路径

4. 随机状态转移策略

- ✓ 人工蚂蚁和真实蚂蚁都按照概率决策规则从一种状态转移到另一种相邻状态。其中的概率决策规则是与问题相关的信息和局部环境信息的函数
- ✓ 在状态转移过程中，人工蚂蚁和真实蚂蚁都只用到了局部信息，没有使用前瞻策略来预见将来的状态，因此所使用的策略在时间和空间上是局部的

人工蚂蚁与真实蚂蚁的不同点

1. 人工蚂蚁生活在离散的时间环境下，其仅考虑自身位于哪个城市，而不考虑在城市间的移动过程，人工蚂蚁从一种离散状态到另一种离散状态。而真实蚂蚁则是处于一个连续的时间环境下
2. 人工蚂蚁具有一定的记忆能力，它可以通过禁忌列表记住已经走过的路径，以保证不会重复走相同的城市。真实蚂蚁是没有记忆的，蚂蚁间的信息交换主要靠留在所经过路径上的信息素，因此，真实蚂蚁在选择路径时有可能重复选择一段路，这不符合TSP问题的要求

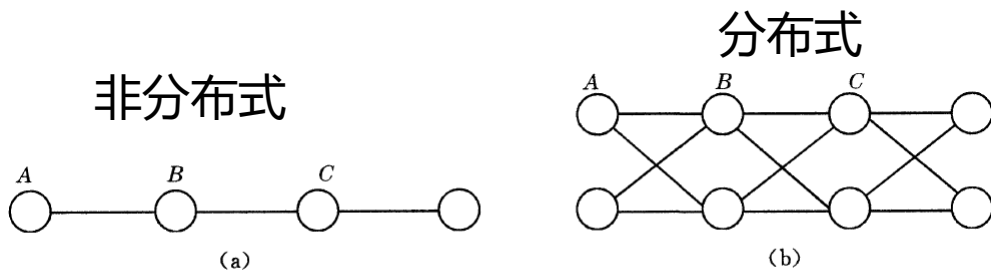
人工蚂蚁与真实蚂蚁的不同点

3. 人工蚂蚁不仅是依据信息素来确定要走的路径的，还会依据一定的启发信息，如相邻边的长度，这意味着人工蚂蚁具有一定的视觉能力，而真实蚂蚁几乎没有“视觉”
4. 人工蚂蚁释放信息素的数量是其生成解的质量的函数，人工蚂蚁更新信息素的时机依赖于特定的问题。例如，大多数人工蚂蚁仅仅在蚂蚁找到一个解之后才更新路径上的信息素

蚁群优化算法的特点

➤ 蚁群算法是通过对生物特征的模拟得到的一种优化算法，它本身具有很多特点：

1. 蚁群算法是一种本质上的并行算法。每只蚂蚁搜索的过程彼此独立，仅通过信息激素进行通信。所以蚁群算法可以看作一个分布式的多智能体系统，它在问题空间的多点同时开始独立的解搜索，不仅增加了算法的可靠性，也使得算法具有较强的全局搜索能力



蚁群优化算法的特点

2. 蚁群算法是一种自组织的算法

- ✓ 所谓自组织，就是组织力或组织指令来自于系统的内部，以区别于其他组织
- ✓ 如果系统在获得空间、时间或者功能结构的过程中，没有外界的特定干预，就可以说系统是自组织的
- ✓ 简单地说，自组织就是在没有外界作用下使得系统从无序到有序（熵减少）的变化过程

蚁群优化算法的特点

3. 蚁群算法具有较强的稳健性

- ✓ 相对于其他算法，蚁群算法对初始路线的要求不高，即蚁群算法的求解结果不依赖于初始路线的选择，而且在搜索过程中不需要进行人工调整
- ✓ 蚁群算法的参数较少，设置简单，因而该算法易于应用到组合优化问题的求解

蚁群优化算法的特点

4. 蚁群算法是一种正反馈算法

- ✓ 从真实蚂蚁的觅食过程中不难看出，蚂蚁能够最终找到最优路径，直接依赖于其在路径上信息素的堆积，而信息素的堆积是一个正反馈的过程
- ✓ 蚁群算法中同样隐含着负反馈机制，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
- ✓ 正反馈缩小搜索范围，保证算法朝着最优解的方向进化；而负反馈保持搜索范围，避免算法过早收敛于不好的结果
- ✓ 正反馈是蚁群算法的重要特征，在正反馈和负反馈共同作用的影响下，蚁群算法得以自组织地进化

-
- 蚂蚁觅食的秘密
 - 蚁群优化算法理论
 - 蚁群优化算法的数学模型及算法流程
 - 举例说明蚁群优化算法的工作机理
 - 蚁群优化算法的改进
 - 蚁群优化算法的参数设置及影响

蚁群优化算法的数学模型

*TSP*问题首先引入城市*i*和*j*之间的距离 $d_{i,j}$ ($i, j = 1, 2, \dots, n$), 其在欧式空间中的定义如下式所示

$$d_{i,j} = \sqrt{\left[\left(x_i - x_j \right)^2 + \left(y_i - y_j \right)^2 \right]}$$

其中 (x_i, y_i) 为城市*i*的空间坐标, (x_j, y_j) 为城市*j*的空间坐标

蚁群优化算法的数学模型

第一个蚁群优化算法称为*Ant System*(AS)算法

*在AS算法中，首先将 m 个蚂蚁随机分配到 n 个不同的城市中，通常 $m \leq n$;

*然后 m 只蚂蚁同时由一个城市运动到另一个城市，逐步完成它们的搜索过程。用 m 只蚂蚁是为了进行 m 次并行计算以提高计算效率。

*整个算法的迭代过程以 N 为刻度, $1 \leq N \leq N_{\max}$,其中 N_{\max} 是预先设定的最大迭代次数。

*在每一次迭代过程中，算法以 t 为刻度, $0 \leq t \leq n$ ，蚂蚁 k ($k = 1, 2, \dots, m$)根据概率转换规则选择下一个城市，由此可以生成一个由 n 个城市组成的行动路线，并伴有信息素的更新

蚁群优化算法的数学模型

AS算法规定蚂蚁 k 由当前城市 i 转到下一个城市 j 的决定因素有以下4个

1.禁忌列表

2.能见度

3.虚拟信息素

4.概率转换规则

禁忌列表

**TSP*问题要求蚂蚁必须经过所有 n 个不同的城市，为了避免蚂蚁重复走入同一个城市，*AS*算法为每只蚂蚁配备了一个记忆空间，即在具体算法实现中设计一个数据结构，由这些数据结构组成的表（矩阵）称为禁忌列表

*禁忌列表的第 k 行（或列）用于存储第 k 只蚂蚁在当前时刻已访问过的所有城市，记为 J^k

禁忌列表

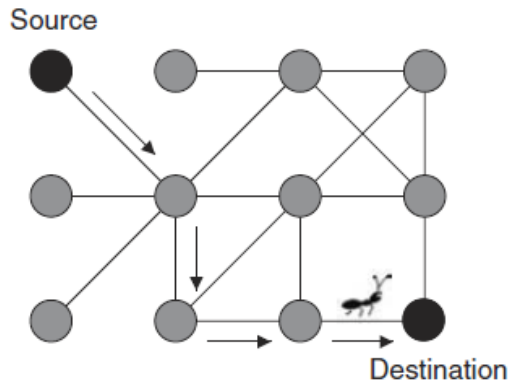
- *每只蚂蚁在选择城市前，先检索该表以确定下一步可能选择的的城市是否已经走过，如果走过，则其不在选择的范围内，这样可以避免重复访问同一个城市
- *当第 k 只蚂蚁在当前城市 i 选择了下一个城市 j ，则在 J^k 的相应位置加入城市 j 。因此，在一个完整的行程中，禁忌列表首先是空的，当选择所要经过的城市后算法将会在线更新禁忌列表，并在完成 n 个城市的遍历而形成一条完整路径后，清空禁忌列表，等待下一次迭代计算

能见度

*能见度定义为距离的倒数，其表示为 $\eta_{i,j} = 1 / d_{i,j}$

*其能见度与目标函数有关，是以本地信息为基础，代表由城市*i*到城市*j*的启发性愿望，两个城市之间的距离越短，能见度越大，被选择的愿望也会越大，由此引导蚂蚁的搜索

*这种信息是固定的，在问题求解过程中不发生变化，因此其又被称为启发信息



虚拟信息素

*虚拟信息素是模拟蚂蚁觅食过程的关键。当算法由城市 i 选择城市 j 后，其将在 i, j 路径上遗留虚拟信息素 τ_{ij}

*虚拟信息素（简称信息素）代表了从城市 i 到城市 j 的获知性愿望。它与能见度相反，是一种动态的全局信息，可反映蚂蚁在解决问题过程中的经验积累和向其他蚂蚁学习的能力，该参数是在线更新的

虚拟信息素

*信息素的更新方式体现在信息素增加和信息素挥发两个方面。每只蚂蚁在所经路径上都会增加一定的信息素，同时为了避免残留信息素过多导致启发信息被淹没，算法引入了信息素挥发机制

*这种更新策略模仿了人类大脑记忆的特点，即在新信息不断存入大脑的同时，存储在大脑中的旧信息会随着时间的推移逐渐淡化，甚至忘记。在算法模型中，通常会利用信息素挥发系数 ρ ($0 < \rho < 1$) 来模拟蚂蚁信息素的挥发过程，以使算法具有负反馈能力

虚拟信息素

城市*i*和城市*j*之间的信息素更新为

$$\tau_{ij}^{new} = (1-\rho) \tau_{ij}^{old} + \sum_{k=1}^m \Delta\tau_{ij}^{(k)} \quad \text{其中}\rho\text{为信息素的挥发率} \in (0, 1)$$

$\Delta\tau_{ij}^{(k)}$ 为蚂蚁*k*在它走过的路径上留下信息素，在这一代中所有蚂蚁

在 $edge_{ij}$ 上总共留下的信息素为 $\sum_{k=1}^m \Delta\tau_{ij}^{(k)}$

虚拟信息素

城市*i*和城市*j*之间的信息素更新为

$$\tau_{ij}^{new} = (1-\rho) \tau_{ij}^{old} + \sum_{k=1}^m \Delta\tau_{ij}^{(k)} \quad \text{其中}\rho\text{为信息素的挥发率} \in (0, 1)$$

信息素更新公式等号右边可以理解为两部分：

- (1) 第一部分是原有信息素的挥发，它均匀地衰减所有的信息素。它执行一个遗忘功能，使得尚未被探索到的区域有机会被开发，有助于丢弃那些构建过的较差的路径，避免算法过快地收敛于一个次优区域
- (2) 第二部分是蚂蚁经过（*i, j*）释放的新生信息素，给评价“好”（有蚂蚁走过）的边增加信息素

虚拟信息素

蚂蚁 k 在城市 i 到城市 j 之间释放的信息素 $\Delta\tau_{ij}^{(k)}$ 的浓度常用三种模型来模拟：

- (1) *Ant – Cycle System*(ACS)蚁周模型
- (2) *Ant – Quantity System*(AQS)蚁量模型
- (3) *Ant – Density System*(ADS)蚁密模型

Ant-Cycle System(ACS) 蚁周模型

*Dorigo*给出 $\Delta\tau_{ij}^{(k)}$ 的三种不同模型:

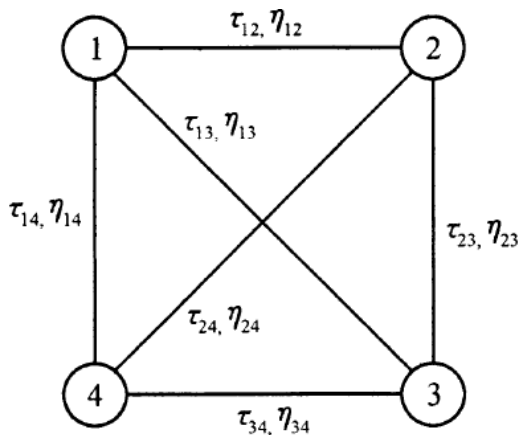
1. *Ant - Cycle System(ACS)*蚁周模型: 利用蚂蚁经过路径的整体信息 (经过路径的总长) 计算释放的信息素浓度

$$\tau_{i,j}(N+1) = (1-\rho)\tau_{i,j}(N) + \sum_{k=1}^m \Delta\tau_{ij}^{(k)}$$
$$\Delta\tau_{ij}^{(k)} = \begin{cases} Q/L_k & \text{若第}k\text{只蚂蚁在本次循环经过}ij \\ 0 & \text{其他} \end{cases}$$

其中 Q 为一个适当的正常数, L_k 为第 k 只蚂蚁在本次访问城市中所走过路径的总长度, 信息素的数量反比于 L_k

Ant-Cycle System(ACS) 蚁周模型

一般选用ACS模型计算释放的信息素浓度，即蚂蚁经过的路径越短，释放的信息素浓度越高



旅行商问题的构建图

- 蚁周模型利用的是整体信息，即蚂蚁完成一个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，通常采用蚁周模型为蚁群算法的基本模型

Ant-Quantity System(AQS)蚁量模型

2. *Ant-Quantity System(AQS)* 蚁量模型：利用蚂蚁经过路径的局部信息（经过各个城市间的距离）计算释放的信息素浓度

$$\tau_{ij}(t+1) = (1-\rho) \tau_{ij}(t) + \sum_{k=1}^N \Delta \tau_{ij}^{(k)}$$

$$\Delta \tau_{ij}^{(k)} = \begin{cases} Q/d_{ij} & \text{若第} k \text{只蚂蚁在时刻} t \text{和} (t+1) \text{之间经过} ij \\ 0 & \text{其他} \end{cases}$$

其中， Q 为一个适当的正常数； d_{ij} 表示第 k 只蚂蚁在本次访问中城市 i 和城市 j 的距离

➤ 蚁量模型利用的是局部信息，即蚂蚁完成一步后更新路径上的信息素

Ant-Density System(ADS)蚁密模型

3. *Ant - Density System(ADS)* 蚁密模型：将信息素释放的浓度取为恒值，并没有考虑不同蚂蚁经过路径长短的影响

$$\tau_{ij}(t+1) = (1-\rho) \tau_{ij}(t) + \sum_{k=1}^N \Delta \tau_{ij}^{(k)}$$

$$\Delta \tau_{ij}^{(k)} = \begin{cases} Q & \text{若第} k \text{只蚂蚁在时刻} t \text{和} (t+1) \text{之间经过} ij \\ 0 & \text{其他} \end{cases}$$

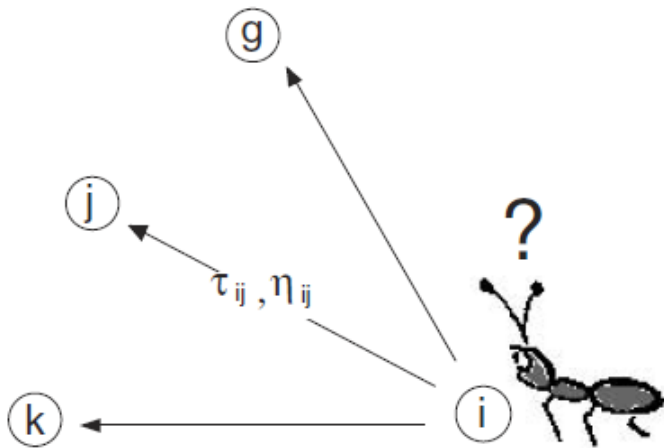
其中， Q 为一个适当的正常数，即在整个访问城市的过程中，第 k 只蚂蚁释放的信息素始终保持不变

➤ 蚁密模型利用的是局部信息，即蚂蚁完成一步后更新路径上的信息素

概率转换规则

状态转移规则规定了蚂蚁 k 从节点 i 移动到下一个节点 j 的规则

- (1) 可选择的一个节点必须是该蚂蚁未曾访问过的
- (2) 优先选择信息素残留量大的路径
- (3) 优先选择主观期望程度高的路径



概率转换规则

在算法的初始时刻，将 m 只蚂蚁随机地放到 n 座城市同时，将每只蚂蚁的禁忌表 $tabu$ 的第一个元素设置为它当前所在的城市。此时各路径上的信息素量相等，

设 $\tau_{ij}(0) = c$ （ c 为一较小的常数），

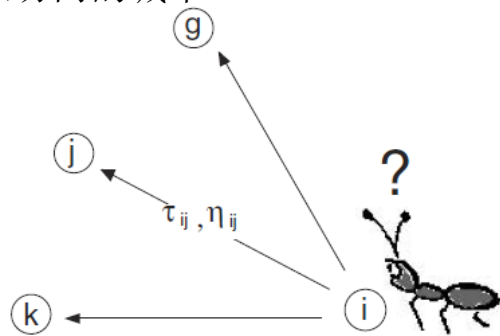
接下来，每只蚂蚁根据路径上残留的信息素量和启发式信息（两城市间的距离）独立地选择下一座城市，在时刻 t ，第 k 只蚂蚁从城市 i 移动到城市 j 的概率由下式给出

$$\text{概率 } p_{ij}^{(k)} = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,j}]^\beta}{\left(\sum_{m=1, m \notin J^k}^n [\tau_{i,m}(t)]^\alpha [\eta_{i,m}]^\beta \right)} & \text{如果城市 } j \text{ 属于第 } k \text{ 只蚂蚁可以访问的城市} \\ 0 & \text{如果城市 } j \text{ 不属于第 } k \text{ 只蚂蚁可以访问的城市} \end{cases}$$

$\tau_{i,j}(t)$ 为城市 i 和城市 j 之间的信息素， $\eta_{i,j} = 1/d_{i,j}$,

其中 $d_{i,j}$ 为城市 i 到城市 j 的距离

α, β = 信息素与启发式信息（能见度）的相对重要性



概率转换规则

$$\text{概率} p_{ij}^{(k)} = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,j}]^\beta}{\sum_{m=1, m \notin J^k}^n [\tau_{i,m}(t)]^\alpha [\eta_{i,m}]^\beta} & \text{如果城市} j \text{属于第} k \text{只蚂蚁可以访问的城市} \\ 0 & \text{如果城市} j \text{不属于第} k \text{只蚂蚁可以访问的城市} \end{cases}$$

α, β = 信息素与启发式信息（能见度）的相对重要性

α 和 β 是两个可以被调整的参数，用于控制信息素和能见度在 $p_{i,j}^k(t)$ 中所占的权重，它们的取值通常来自于经验或反复的实验。

*参数 α 表示残留信息的相对重要程度，其大小反映了蚂蚁在路径搜索过程中确定性因素作用的强弱。如果 $\alpha = 0$ ，则最近的城市会被选择，这类似于经典的随机贪婪算法

概率转换规则

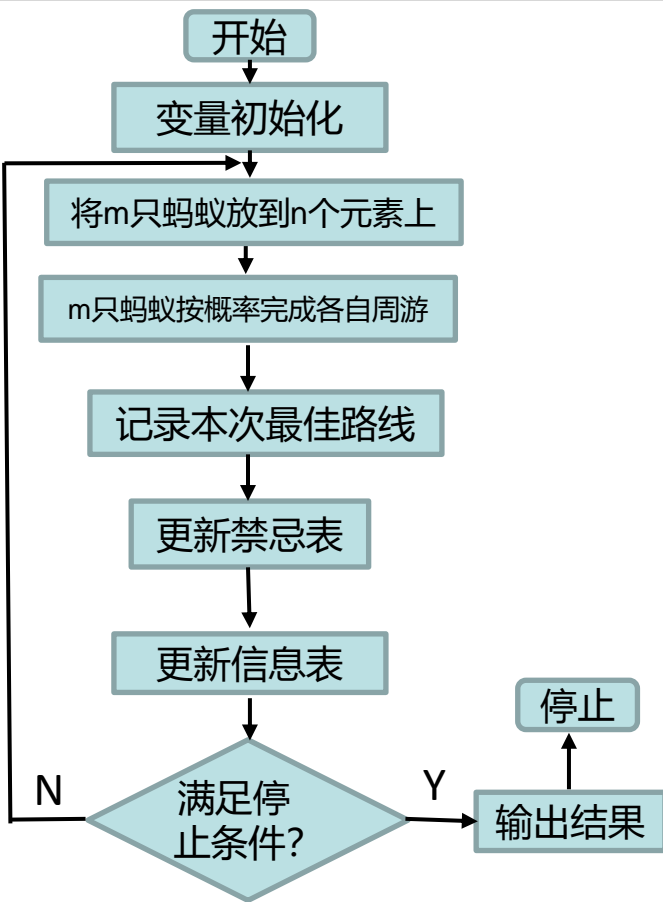
$$\text{概率} p_{ij}^{(k)} = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,j}]^\beta}{\sum_{m=1, m \notin J^k}^n [\tau_{i,m}(t)]^\alpha [\eta_{i,m}]^\beta} & \text{如果城市} j \text{属于第} k \text{只蚂蚁可以访问的城市} \\ 0 & \text{如果城市} j \text{不属于第} k \text{只蚂蚁可以访问的城市} \end{cases}$$

* β 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，则该状态概率越接近于贪婪选择。

如果 $\beta = 0$ ，则只有信息素放大机制在独自工作，此时算法会陷入一个范围窄小的搜索空间，这将导致算法可能会迅速获得一个非最优解

因此在信息素浓度和能见度之间确定一种折中关系是非常必要的

蚁群优化算法流程



蚁群优化算法的具体实现步骤如下：

Step1.初始化参数：给定蚂蚁个数 m ；城市节点个数 n ；迭代计数 $N_c = 0$ ，最大迭代次数 $MAXN_c$ ，信息素启发因子 α 、期望启发因子 β 、信息素挥发因子 ρ 、信息素总量 Q ；信息素矩阵 $Tau = \{1\}_{n \times n}$ ；期望值启发矩阵 $E_{ta} = \{1 / d_{ij}\}_{n \times n}$ ；蚂蚁禁忌表 $Tabu = \{0\}_{m \times n}$ ，蚂蚁 k 爬过某个节点之后，该节点被放入禁忌表，不能再次爬过。将 m 个蚂蚁随机置于 n 个节点（城市）上（ $m \leq n$ ），令有向图上每条边（ i, j ）的初始化信息量 $\tau_{ij}(t) = c$ ，其中 c 表示常数，且初始时刻 $\Delta\tau_{ij}(0) = 0$

Step2.循环次数 $N_c = N_c + 1$

Step3.蚂蚁的禁忌表索引号 $k = 1$

Step4.蚂蚁数目 $k = k + 1$

Step5.蚂蚁个体根据状态转移概率公式计算的概率选择元素 j 并前进

Step6.修改禁忌表指针，即选择好之后将蚂蚁移动到新的元素，并把该元素移动到该蚂蚁个体的禁忌表中

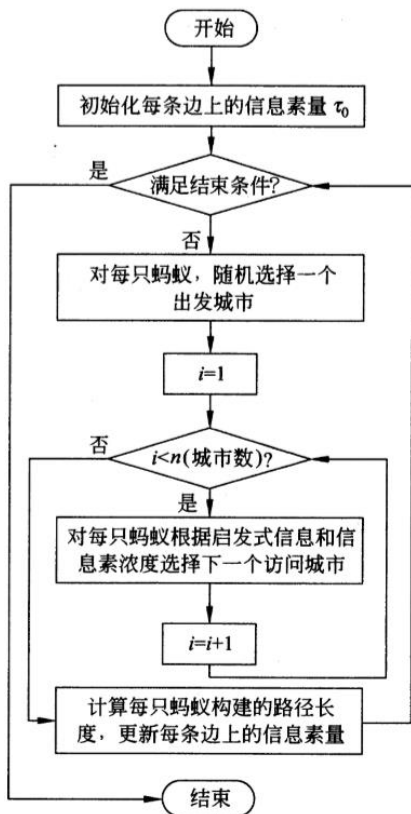
Step7.若禁忌表集合中元素未遍历完，即 $k < m$ ，则跳转到第4步；否则执行第8步

Step8.记录本次最佳路线

Step9.更新每条路径上的信息量

Step10.若满足结束条件，即如果循环次数 $N_c \geq MAXN_c$ ，则循环结束并输出程序优化结果；否则清空禁忌表并跳转到第2步

蚁群优化算法求解TSP的流程图和伪代码



//功能：蚂蚁系统伪代码

//说明：本例以求解 TSP 问题为目标

//参数：N 为城市规模

procedure AS

for each edge

set initial pheromone value τ_0 .

end for

while not stop

for each ant k

randomly choose an initial city.

for i = 1 to n

choose next city j with the probability given by Eq. (5.1).

end for

end for

compute the length C_k of the tour constructed by the kth ant.

for each edge

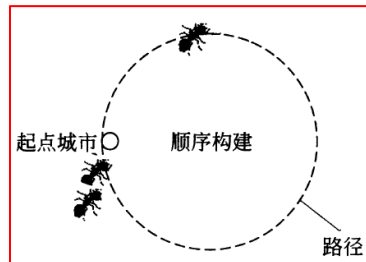
update the pheromone value by Eq. (5.2).

end for

end while

print result.

end procedure



-
- 蚂蚁觅食的秘密
 - 蚁群优化算法理论
 - 蚁群优化算法的数学模型及算法流程
 - 举例说明蚁群优化算法的工作机理
 - 蚁群优化算法的改进
 - 蚁群优化算法的参数设置及影响

蚁群优化算法

➤ 接下来，我们将利用旅行商问题来阐述蚁群优化算法，主要原因：

1. 蚁群优化算法是根据模拟蚂蚁寻找食物的最短路径行为设计的仿生算法，TSP问题与真实蚂蚁选择最短路径的相似
2. TSP问题是一个NP-hard优化问题，具有代表性

蚁群优化算法

3. 对于一个新算法，TSP问题是一个标准测试问题。在TSP问题中如果所需遍历的城市数发生变化，那么算法复杂度和TSP问题的解空间也会随之发生变化。通过简单地改变所设定的城市数，就可以模拟不同难易程度的优化问题，因此TSP问题适合用于测试计算智能算法的性能。目前，许多计算智能算法之间的比较都将TSP问题作为测试函数
4. TSP问题很容易理解，算法的编码较为简单，不会有技术和专业上的难度
5. 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

旅行商问题

用旅行商问题能够说明蚂蚁系统。

*每只蚂蚁从一个城市移动到另一个城市，信息素将留在蚂蚁经过的路径上，信息素不只被存放，还会挥发。

*蚂蚁从当前所在的城市移动到其他城市的概率与城市之间信息素的量成正比，与城市之间的距离成反比。

旅行商问题

在生物学的蚂蚁系统中，节点 i 和节点 j 之间信息素的量与节点 j 和节点 i 之间的相同

但在蚂蚁系统的仿真中，从节点 i 移动到节点 j 可能得到好的解，从节点 j 移动到节点 i 却可能得到差的解。

因此 $\tau_{ij} \neq \tau_{ji}$ ，它相应于非对称的旅行商问题。

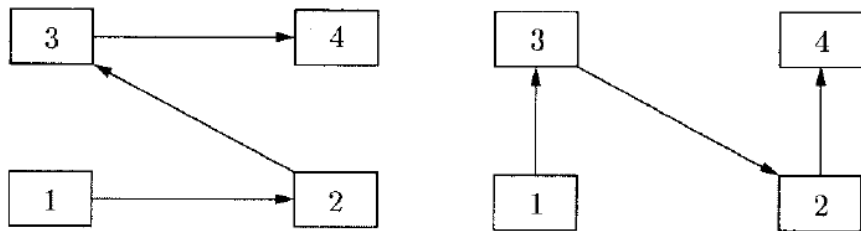


图 10.6 本例假定行程从节点 1 开始. 左边的行程比右边的差很多 (即它的距离较长), 但是两个行程都含有节点 2 和节点 3 之间的路径. 由于左边的行程长而右边的行程短, 对于有效的蚁群优化算法, τ_{23} 应该比 τ_{32} 小; 也就是说, 从节点 2 到节点 3 的吸引力应该比从节点 3 到节点 2 的吸引力低.

算法10.1

算法 10.1 解旅行商问题的简单蚂蚁系统. 在每一代, 城市 i 和城市 j 之间的一些信息素会挥发, 但信息素也会因蚂蚁在这两个城市之间移动而增加.

n = 城市的个数

α, β = 信息素与启发式信息的相对重要性

Q = 沉积常数

ρ = 挥发率 $\in (0, 1)$

$\tau_{ij} = \tau_0$ (城市 i 和城市 j 之间的初始信息素), $i \in [1, n], j \in [1, n]$

d_{ij} = 城市 i 和城市 j 之间的距离, $i \in [1, n], j \in [1, n]$

While not (终止准则)

For $q = 1$ to $n - 1$

For 每只蚂蚁 $k \in [1, N]$

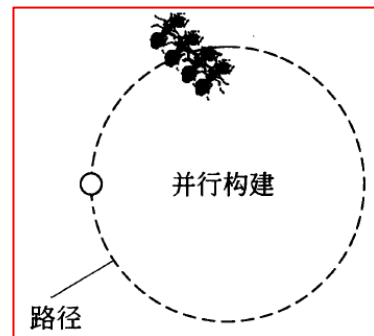
初始化每只蚂蚁 $k \in [1, N]$ 的起点城市 c_{k1}

初始化蚂蚁 k 已访问的城市的集合: $C_k \leftarrow \{c_{k1}\}$, $k \in [1, N]$

For 每一个城市 $j \in [1, n], j \notin C_k$

$$\text{概率 } p_{ij}^{(k)} \leftarrow (\tau_{ij}^\alpha / d_{ij}^\beta) \bigg/ \left(\sum_{m=1, m \notin C_k}^n \tau_{im}^\alpha / d_{im}^\beta \right)$$

下一个 j



旅行商问题

第 k 只蚂蚁从城市 i 移动到城市 j 的概率由下式给出

$$\text{概率 } p_{ij}^{(k)} = \begin{cases} \frac{(\tau_{ij}^\alpha / d_{ij}^\beta)}{\left(\sum_{m=1, m \notin C_k}^n \tau_{im}^\alpha / d_{im}^\beta \right)} & \text{如果城市 } j \text{ 属于第 } k \text{ 只蚂蚁可以访问的城市} \\ 0 & \text{如果城市 } j \text{ 不属于第 } k \text{ 只蚂蚁可以访问的城市} \end{cases}$$

τ_{ij} 为城市 i 和城市 j 之间的信息素， d_{ij} 为城市 i 到城市 j 的距离

$\alpha, \beta =$ 信息素与启发式信息的相对重要性

算法10.1

让蚂蚁 k 以概率 $p_{ij}^{(k)}$ 到城市 j

用 $c_{k,q+1}$ 表示在上一行中选择的城市

$$C_k \leftarrow C_k \cup \{c_{k,q+1}\}$$

下一只蚂蚁

下一个 q

$L_k \leftarrow$ 由蚂蚁 k 构建的路径总长度, $k \in [1, N]$

For 每一个城市 $i \in [1, n]$ 和每一个城市 $j \in [1, n]$

For 每一只蚂蚁 $k \in [1, N]$

If 蚂蚁 k 由城市 i 到城市 j then

$$\Delta\tau_{ij}^{(k)} \leftarrow Q/L_k$$

else

$$\Delta\tau_{ij}^{(k)} \leftarrow 0$$

End if

下一只蚂蚁

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)}$$

下一对城市

下一代

更新信息素

每一只蚂蚁 k 在它走过的路径上留下信息素 $\Delta\tau_{ij}^{(k)} = Q/L_k$

其中 Q 为一个适当的常数， L_k 为该路径的总长度，信息素的数量反比于 L_k

在这一代中所有蚂蚁在 $edge_{ij}$ 上总共留下的信息素为 $\sum_{k=1}^N \Delta\tau_{ij}^{(k)}$

城市 i 和城市 j 之间的信息素更新为

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)}$$

其中 ρ 为信息素的挥发率 $\in (0, 1)$

算法10.1的实施细节

实施细节包括智能初始化、精英和变异。

(1) 蚁群优化的性能和进化算法的性能一样，在很大程度上取决于初始化。对于旅行商问题，可以强制蚂蚁在每个决策点一定要访问最近的城市，来初始化某些个体。

(2) 蚁群优化可以用精英，可以通过记录在每一代中最好的几只蚂蚁并强制它们在下一代重复同一条路径（在该路径上增强额外的信息素）来实施精英，保证从一代到下一代不会丢掉最好的路径。

算法10.1的实施细节

带精英的蚂蚁系统算法也被称为精英蚂蚁系统。

$$\tau_{ij} = (1-\rho) \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)} + e\Delta\tau_{ij}^{(bs)}$$

其中

$$\Delta\tau_{ij}^{(bs)} = \begin{cases} Q / L^{bs} & \text{如果}(i, j)\text{边属于历史最优路径} \\ 0 & \text{otherwise} \end{cases}$$

(3) 蚁群优化也可以用变异，用某个变异概率随机转换路径就可以将变异纳入算法中。

算法10.1的调试参数

算法 10.1 显示蚂蚁系统中需要调试的几个参数. 它们包括:

- 蚂蚁的个数 N , 种群规模;
- α 与 β , 信息素的量与启发式信息的相对重要性;
- Q , 为沉积常数;
- ρ , 为挥发速率;
- τ_0 , 为每个城市之间信息素初始的量.

有几位研究人员研究了这些参数的影响. 这些参数的常用推荐值如下 [Dorigo et al., 1996]:

- $N = n$ (即蚂蚁数 = 城市的个数);
- $\alpha = 1, \beta = 5$;
- $Q = 100$, 尽管它的影响不是很明显;
- $\rho \in [0.5, 0.99]$;
- $\tau_0 \approx 10^{-6}$.

旅行商 (TSP) 问题

旅行商 (TSP) 问题是一类受到广泛研究的组合优化问题。TSP问题描述很简单, 即旅行商要访问 n 个城市, 那他要以怎样的访问顺序来访问这 n 个城市才能使得行走路径最短。

TSP问题中常用的符号

(1) 索引

i, j : 城市索引, 其中 $i, j = 1, 2, \dots, n$

(2) 参数

n : 城市总数

d_{ij} : 城市 i 到城市 j 的距离, 由 d_{ij} 构成的矩阵是对称矩阵

(3) 决策变量

x_{ij} : 0, 1决策变量, 若选择路线 (i, j) , 则 $x_{ij} = 1$, 否则取值为0

旅行商 (TSP) 问题

对于含有4个城市的TSP问题，若按照 $Z \rightarrow X \rightarrow W \rightarrow Y$ 的顺序来访问，我们将4个城市的访问顺序反映在下表中

城市名	访问顺序			
	1	2	3	4
X	0	1	0	0
Y	0	0	0	1
Z	1	0	0	0
W	0	0	1	0

从上面表格中可以清楚地看到，每个城市只被访问一次，且每次只访问一个城市

TSP问题的数学模型

*TSP*问题一般可以表述为如下形式：

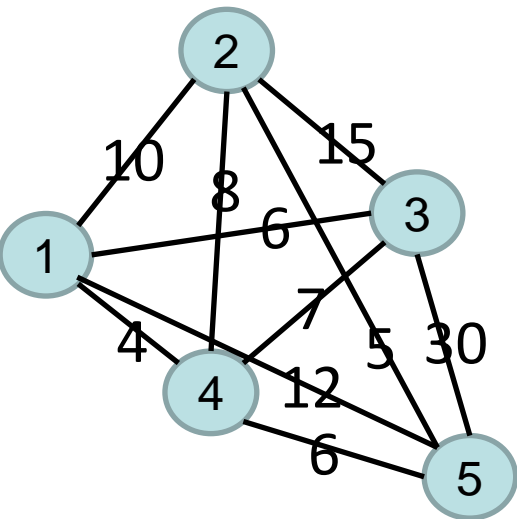
$$\min \quad z = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} x_{ij} + d_{n1} x_{n1})$$

$$s.t. \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, 3, \dots, n$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, 3, \dots, n$$

$$\text{其中, } x_{ij} = \begin{cases} 1, & \text{若选中 } route(i, j) \\ 0, & \text{否则} \end{cases}$$

示例：5个城市的TSP问题



d_{ij}	1	2	3	4	5
1	0	10	6	4	12
2	10	0	15	8	5
3	6	15	0	7	30
4	4	8	7	0	6
5	12	5	30	6	0

* 对称 *TSP* 问题: $d_{ij} = d_{ji}$

* 不对称 *TSP* 问题: 至少一条边 $d_{ij} \neq d_{ji}$

示例：5个城市的TSP问题

TSP问题一般可以表述为如下形式：

$$\min \quad Z = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} x_{ij} + d_{n1} x_{n1})$$

第一步：确定种群规模, 迭代次数, 信息素启发式因子 α , 期望启发因子 β , 信息素挥发因子 ρ , 信息素强度（沉积系数） Q , 初始信息素量 τ_0

$$N_p = 3 \quad T = 10 \quad \alpha = 1 \quad \beta = 1 \quad \rho = 0.5 \quad Q = 100$$

$$\tau_{i,j}(0) = 2$$

示例：5个城市的TSP问题

TSP问题一般可以表述为如下形式：

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} x_{ij} + d_{n1} x_{n1})$$

第一步：确定种群规模, 迭代次数, 信息素启发式因子 α , 期望启发因子 β , 信息素挥发因子 ρ , 信息素强度（沉积系数） Q , 初始信息素量 τ_0

$$N_p = 3 \quad T = 10 \quad \alpha = 1 \quad \beta = 1 \quad \rho = 0.5 \quad Q = 100$$
$$\tau_{i,j}(0) = 2$$

第二步：将3只蚂蚁随机分配到5个城市，分配的城市作为该蚂蚁的起点城市

蚂蚁	1	2	3
城市	3	4	1
禁忌列表	3	4	1

第一只蚂蚁

第三步：对每只蚂蚁计算转移概率 p ，采用“轮盘赌”选择下一个城市 j ，并将城市 j 加入到该蚂蚁的禁忌列表

第1只蚂蚁从城市3出发

$$\alpha = 1 \quad \beta = 1 \quad \rho = 0.5 \quad Q = 100 \quad \tau_{i,j}(0) = 2$$

$$p_{31}^{(1)} = \frac{\tau_{31}^{\alpha} / d_{31}^{\beta}}{\tau_{31}^{\alpha} / d_{31}^{\beta} + \tau_{32}^{\alpha} / d_{32}^{\beta} + \tau_{34}^{\alpha} / d_{34}^{\beta} + \tau_{35}^{\alpha} / d_{35}^{\beta}} = \frac{2^1 / 6^1}{2^1 / 6^1 + 2^1 / 10^1 + 2^1 / 7^1 + 2^1 / 30^1} \\ = 0.376$$

第 k 只蚂蚁从城市 i 移动到城市 j 的概率由下式给出

$$\text{概率 } p_{ij}^{(k)} = \begin{cases} \frac{(\tau_{ij}^{\alpha} / d_{ij}^{\beta})}{\left(\sum_{m=1, m \notin C_k}^n \tau_{im}^{\alpha} / d_{im}^{\beta} \right)} & \text{如果城市 } j \text{ 属于第 } k \text{ 只蚂蚁可以访问的城市} \\ 0 & \text{如果城市 } j \text{ 不属于第 } k \text{ 只蚂蚁可以访问的城市} \end{cases}$$

τ_{ij} 为城市 i 和城市 j 之间的信息素， d_{ij} 为城市 i 到城市 j 的距离

α, β = 信息素与启发式信息的相对重要性

第一只蚂蚁

$$p_{32}^{(1)} = \frac{\tau_{32}^{\alpha} / d_{32}^{\beta}}{\tau_{31}^{\alpha} / d_{31}^{\beta} + \tau_{32}^{\alpha} / d_{32}^{\beta} + \tau_{34}^{\alpha} / d_{34}^{\beta} + \tau_{35}^{\alpha} / d_{35}^{\beta}} = \frac{2^1 / 10^1}{2^1 / 6^1 + 2^1 / 10^1 + 2^1 / 7^1 + 2^1 / 30^1} = 0.226$$

$$p_{34}^{(1)} = \frac{\tau_{34}^{\alpha} / d_{34}^{\beta}}{\tau_{31}^{\alpha} / d_{31}^{\beta} + \tau_{32}^{\alpha} / d_{32}^{\beta} + \tau_{34}^{\alpha} / d_{34}^{\beta} + \tau_{35}^{\alpha} / d_{35}^{\beta}} = \frac{2^1 / 7^1}{2^1 / 6^1 + 2^1 / 10^1 + 2^1 / 7^1 + 2^1 / 30^1} = 0.323$$

$$p_{35}^{(1)} = \frac{\tau_{35}^{\alpha} / d_{35}^{\beta}}{\tau_{31}^{\alpha} / d_{31}^{\beta} + \tau_{32}^{\alpha} / d_{32}^{\beta} + \tau_{34}^{\alpha} / d_{34}^{\beta} + \tau_{35}^{\alpha} / d_{35}^{\beta}} = \frac{2^1 / 30^1}{2^1 / 6^1 + 2^1 / 10^1 + 2^1 / 7^1 + 2^1 / 30^1} = 0.075$$

$$\alpha = 1 \quad \beta = 1 \quad \rho = 0.5 \quad Q = 100 \quad \tau_{i,j}(0) = 2$$

第 k 只蚂蚁从城市 i 移动到城市 j 的概率由下式给出

$$\text{概率 } p_{ij}^{(k)} = \begin{cases} \frac{(\tau_{ij}^{\alpha} / d_{ij}^{\beta})}{\left(\sum_{m=1, m \notin C_k}^n \tau_{im}^{\alpha} / d_{im}^{\beta} \right)} & \text{如果城市 } j \text{ 属于第 } k \text{ 只蚂蚁可以访问的城市} \\ 0 & \text{如果城市 } j \text{ 不属于第 } k \text{ 只蚂蚁可以访问的城市} \end{cases}$$

τ_{ij} 为城市 i 和城市 j 之间的信息素, d_{ij} 为城市 i 到城市 j 的距离

α, β = 信息素与启发式信息的相对重要性

第一只蚂蚁

第三步：对每只蚂蚁计算转移概率 p ，采用“轮盘赌”选择下一个城市 j ，并将城市 j 加入到该蚂蚁的禁忌列表

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	1	3	3	1	2	0.376
				2	2	0.226
				4	2	0.323
				5	2	0.075

第一只蚂蚁

第三步：对每只蚂蚁计算转移概率 p ，采用“轮盘赌”选择下一个城市 j ，并将城市 j 加入到该蚂蚁的禁忌列表

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	1	3	3	1	2	0.376
				2	2	0.226
				4	2	0.323
				5	2	0.075

假设下一个城市依据“轮盘赌”选择为1

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	2	1	3,1	2	2	0.231
				4	2	0.577
				5	2	0.192

第一只蚂蚁

假设下一个城市依据“轮盘赌”选择为4

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	3	4	3,1,4	2	2	0.429
				5	2	0.571

第一只蚂蚁

假设下一个城市依据“轮盘赌”选择为4

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	3	4	3,1,4	2	2	0.429
				5	2	0.571

假设下一个城市依据“轮盘赌”选择为2

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
1	4	2	3,1,4,2	5	2	1

第一只蚂蚁的路径为3 - 1 - 4 - 2 - 5 - 3

$$\text{总距离 } L_1 = d_{31} + d_{14} + d_{42} + d_{25} + d_{53} = 6 + 4 + 8 + 5 + 30 = 53$$

第二只蚂蚁

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
2	1	4	4	1	2	0.365
				2	2	0.183
				3	2	0.209
				5	2	0.243
	2	5	4,5	1	2	0.263
				2	2	0.632
				3	2	0.105
	3	2	4,5,2	1	2	0.6
				3	2	0.4
	4	1	4,5,2,1	3	2	1.0
	5	3	4,5,2,1,3			

第二只蚂蚁的路径为4 - 5 - 2 - 1 - 3 - 4

$$\text{总距离 } L_2 = d_{45} + d_{52} + d_{21} + d_{13} + d_{34} = 6 + 5 + 10 + 6 + 7 = 34$$

第三只蚂蚁

k	t	当前城市	禁忌列表	下一个城市	信息素 τ_{ij}	p_{ij}
3	1	1	1	2	2	0.166
				3	2	0.278
				4	2	0.417
				5	2	0.139
	2	4	1,4	2	2	0.288
				3	2	0.329
				5	2	0.383
	3	5	1,4,5	2	2	0.857
				3	2	0.143
	4	2	1,4,5,2	3	2	1.0
	5	3	1,4,5,2,3			

第三只蚂蚁的路径为1 - 4 - 5 - 2 - 3 - 1

总距离 $L_3 = d_{14} + d_{45} + d_{52} + d_{23} + d_{31} = 4 + 6 + 5 + 15 + 6 = 36$

更新信息素

第四步：采用（ACS）蚁周模型，更新所有蚂蚁经过的各条路径上的信息素

第一只蚂蚁的路径为3 - 1 - 4 - 2 - 5 - 3

$$\Delta\tau_{ij}^{(1)} = Q/L_k = 100 / 53 = 1.887$$

总距离 $L_1 = d_{31} + d_{14} + d_{42} + d_{25} + d_{53} = 6 + 4 + 8 + 5 + 30 = 53$

第二只蚂蚁的路径为4 - 5 - 2 - 1 - 3 - 4

$$\Delta\tau_{ij}^{(2)} = Q/L_k = 100 / 34 = 2.941$$

总距离 $L_2 = d_{45} + d_{52} + d_{21} + d_{13} + d_{34} = 6 + 5 + 10 + 6 + 7 = 34$

第三只蚂蚁的路径为1 - 4 - 5 - 2 - 3 - 1

$$\Delta\tau_{ij}^{(3)} = Q/L_k = 100 / 36 = 2.778$$

总距离 $L_3 = d_{14} + d_{45} + d_{52} + d_{23} + d_{31} = 4 + 6 + 5 + 15 + 6 = 36$

每一只蚂蚁 k 在它走过的路径上留下信息素 $\Delta\tau_{ij}^{(k)} = Q/L_k$

其中 Q 为一个适当的常数， L_k 为该路径的总长度，信息素的数量反比于 L_k

在这一代中所有蚂蚁在 $edge_{ij}$ 上总共留下的信息素为 $\sum_{k=1}^N \Delta\tau_{ij}^{(k)}$

城市 i 和城市 j 之间的信息素更新为

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)} \quad \text{其中}\rho\text{为信息素的挥发率} \in (0, 1)$$

更新信息素

初始信息素

τ_{ij}	1	2	3	4	5
1	0	2	2	2	2
2	2	0	2	2	2
3	2	2	0	2	2
4	2	2	2	0	2
5	2	2	2	2	0

更新信息素

更新后信息素

τ_{ij}	1	2	3	4	5
1	0	1	$1+2.941$	$1+1.887+2.778$	1
2	$1+2.941$	0	$1+2.778$	1	$1+1.887$
3	$1+1.887+2.778$	1	0	$1+2.941$	1
4	1	$1+1.887$	1	0	$1+2.941+2.778$
5	1	$1+2.941+2.778$	$1+1.887$	1	0

更新信息素

更新后信息素

τ_{ij}	1	2	3	4	5
1	0	1	3.941	5.665	1
2	3.941	0	3.778	1	2.887
3	5.665	1	0	3.941	1
4	1	2.887	1	0	5.665
5	1	5.665	2.887	1	0

第五步：记录当前最优解，判断是否满足程序终止条件，如果满足输出最优解，如果不满足程序终止条件，重新随机分配蚂蚁的初始城市，进入下一次迭代

蚁群优化算法存在的问题

蚁群算法在解决TSP问题上具有极大的优越性，但它还存在如下问题：

- ① 搜索时间长。如果问题规模较大，需要花费大量时间进行路径选择，相比其他优化算法，蚁群算法的时间复杂度较高，搜索时间较长
- ② 易陷入局部最优解。蚁群算法采用正反馈机制，随着搜索不断进行，路径上的信息素差异越来越大，所有蚂蚁搜索的路线基本趋于一致，在局部最优解附近徘徊，甚至出现停滞，此时，很难再搜索到更优解。需要采取有效策略减弱或消除停滞现象，如利用适当的信息素差异平滑策略和信息素挥发机制

蚁群优化算法存在的问题

- ③在其他实际应用中，如图像处理中寻求最优模板问题，并不要求所有蚂蚁都找到最优模板，而只需要一只找到最优模板即可。如果要求所有蚂蚁都找到最优模板，反而影响了计算效率

许多学者提出了各种改进算法，在一定程度上提高了蚁群算法的性能。下面介绍两种最具代表性的改进算法，

1. 蚁群系统 (Ant Colony System, ACS)
2. 最大-最小蚁群模型 (MAX-MIN Ant System, MMAS)

蚁群优化算法的改进

- 蚁群系统 (Ant Colony System, ACS)
- 最大-最小蚂蚁系统 (MAX-MIN Ant System, MMAS)

蚁群系统 (Ant Colony System, ACS)

- *ACS模型最早是由Dorigo多里戈等人在基本AS模型的基础上提出的，它解决了基本AS模型在求解过程中由随机选择策略造成的算法易陷入局部最优、收敛速度慢等问题
- *ACS模型较AS模型能更快地得到最优解，这是由于ACS模型在选择城市时，通过改变转换规则提高了算法的搜索能力
- *ACS模型采用局部信息素更新和全局信息素更新相结合的方式，有利于算法发现更多潜在的最优解，从而提高了算法的搜索质量

蚁群系统 (Ant Colony System, ACS)

ACS模型的基本过程为：

在初始时刻，在 n 个城市中随机放置 m 只蚂蚁，并使蚂蚁多次利用状态转移规则建立一条路径（即TSP问题一个可行解）。

在建立路径的过程中，蚂蚁会受信息素信息（信息强度高的边对蚂蚁更有吸引力）和启发信息（可使蚂蚁倾向于选择最短路径）的指导；

同时，蚂蚁会采用局部更新规则更新已经访问过路径上的信息素。当所有蚂蚁都建立了完整的路径后，再采用全局更新规则对路径上的信息素进行更新，直至整个搜索过程结束

蚁群系统 (Ant Colony System, ACS)

ACS模型与AS模型的主要区别有3点：

1.增加了使用随机比例规则进行状态转移

2.修改了全局信息素更新规则

3.ACS模型新增了对各条路径信息量调整的局部信息素更新规则

收敛于全局最优解及算法运行效率

*一般地，要使算法收敛于全局最优解，并获得较高的求解效率，就务必要求搜索范围尽可能大，而不能局限在已有路径这个空间周围，这不可避免地会使算法的运行效率降低

*反之，要想使算法的运行速度加快，以获得令人满意的运行效率，算法的搜索空间就不能太大，但这样会使算法产生收敛于局部最优解的风险

ACS模型的伪随机选取规则

*在AS模型中，蚁群完全依赖概率进行路径选择，并且使用的是随机选取规则，会有倾向地对新路径进行探索

*在ACS模型中，蚁群采用了不同的状态转换规则（伪随机选取规则），这使得关于问题的先验知识得到了更好的利用，强化了算法的局部搜索能力

用 $(a_k \rightarrow j)$ 记第 k 只蚂蚁在构造候选解时去到城市 j 这一事件。
记 $\Pr(a_k \rightarrow j)$ 为 $(a_k \rightarrow j)$ 的概率。

ACS模型的伪随机选取规则

*基于以上考虑，ACS模型采用的解决方式为：当 $q \leq q_0$ 时，增强已有的较好路径上的信息素，即选择当前转移概率最大的城市；当 $q > q_0$ 时，按照AS模型中的选路方式来选择下一个要去的城市，以扩大搜索空间。这样做的优点是保障算法收敛性的同时可加快算法的收敛速度

*通过调整 q_0 来影响搜索，以使模型集中于选择最好的解，而不是持续地搜索下去。当 q_0 接近于1时，只有本地的优化解会被选择，但这并不能保证算法获得全局最优解；反之，当 q_0 接近于0时，所有的局部解都要被检测，但其中的局部最优解将会被分配一个较大的权值

ACS模型的伪随机选取规则

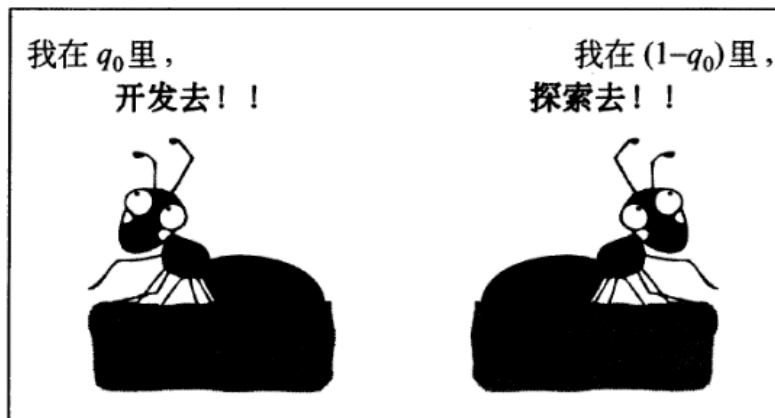
标准蚂蚁系统与蚁群系统在构造候选解时的差别如下：

$$\left. \begin{array}{l} \text{蚂蚁系统: } \Pr(a_k \rightarrow j) = p_{ij}^{(k)} \\ \text{蚁群系统: } \Pr(a_k \rightarrow j) = \begin{cases} 1, & \text{如果 } j = \arg \max_j p_{ij}^{(k)} \\ 0, & \text{其他} \end{cases} \quad \text{如果 } r < q_0 \\ p_{ij}^{(k)} & \quad \text{如果 } r \geq q_0 \end{array} \right\}$$

其中 r 是在 $[0, 1]$ 上均匀分布的一个随机数， $q_0 \in [0, 1]$ 是一个可调参数，通常 $q_0 = 0.9$

ACS模型中 q_0 的作用

q_0 是一个在算法的求解效率与运行效率之间起到平衡作用的可调整参数，它的大小决定了先验知识与探索路径之间的选择程度。较大的 q_0 侧重于开发；反之，侧重于探索



ACS 中的“开发”与“探索”

蚁群系统 (Ant Colony System, ACS)

ACS模型与AS模型的主要区别有3点：

1.增加了使用随机比例规则进行状态转移

2.修改了全局信息素更新规则

3.ACS模型新增了对各条路径信息量调整的局部信息素更新规则

ACS模型的全局信息素更新规则

*在AS模型中，所有的蚂蚁都会在经过的路径上留下信息素

*在ACS模型中，只有找到最优路径的蚂蚁才被允许释放信息素，且对于非最优路径上的信息素也不进行挥发操作，而是只更新属于最好路径的各条边的信息素浓度，即全局信息素更新。这样有利于诱使更多的蚂蚁搜索已发现最好路径周围的路径，进而增强了模型搜索的导向性

ACS模型的全局信息素更新规则

全局更新规则如下式所示

$$\tau_{i,j}(N+1) = (1-\rho)\tau_{i,j}(N) + \rho\Delta\tau_{i,j}(N), \quad (i,j) \in T^+$$

式中, $\Delta\tau_{i,j}(N) = 1/L^+$

T^+ 为本次迭代（或至今最优）得到的最优路径 L^+ 为

本次迭代（或至今最优）的最优路径长度

要强调的是: 不论是信息素的蒸发还是释放, 都只在属于最优路径的边上进行, 这里与AS有很大的区别

通过全局信息素的更新, 最优解得到了局部增强。为了发现其他解, 我们还必须进行局部信息素更新

蚁群系统 (Ant Colony System, ACS)

ACS模型与AS模型的主要区别有3点：

1.增加了使用随机比例规则进行状态转移

2.修改了全局信息素更新规则

3.ACS模型新增了对各条路径信息量调整的局部信息素更新规则

ACS模型的局部信息素更新规则

当蚂蚁 k 选择由城市 i 运动到城市 j 时, (i, j) 上的信息素更新规则如下式所示

$$\tau_{i,j}(t+1) = (1-\varphi)\tau_{i,j}(t) + \varphi\tau_0$$

其中 $\varphi \in [0, 1]$ 是局部信息素挥发（衰减）常数， τ_0 是初始信息素的量。
如果 $\varphi = 0$ 则 τ_{ij} 不变，这时回到最初的蚂蚁系统。

通过实验确定： $\varphi = 0.1$ ， $\tau_0 = (n \times L_{n,n})^{-1}$ 可得到较好的解，其中 n 是城市的数量， $L_{n,n}$ 是由贪婪算法构造的路径的长度

ACS模型的局部信息素更新规则

*通常 $\tau_0 = (n \times L_{n,n})^{-1} \leq \tau_{i,j}(t)$ ，上式表明当蚂蚁通过城市*i* 和城市*j*，这段路径上的信息素会衰减。

*这在生物学上并不准确，但会劝阻别的蚂蚁跟从相同的路径，因此它鼓励更多的探索和多样性。从而间接地促进蚂蚁探索那些仍未被访问过的城市。

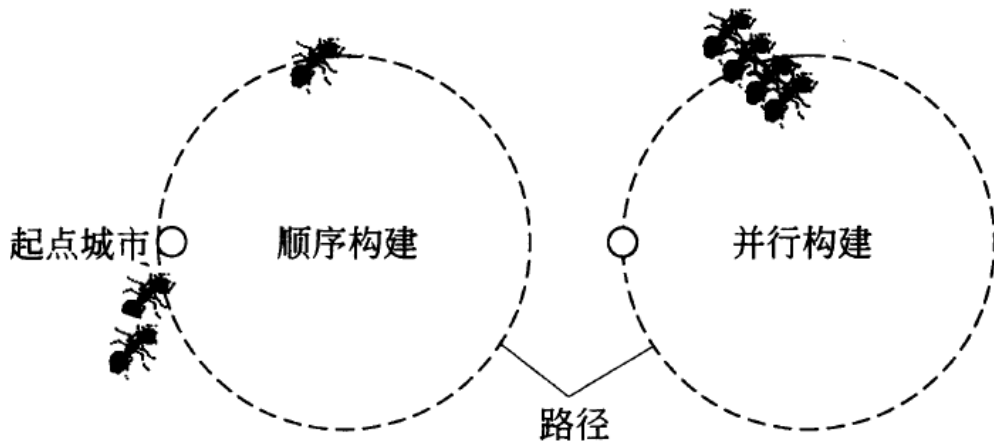
*这样操作将导致蚂蚁不会集中收敛于单一路径。如果没有局部更新过程，所有蚂蚁都可能会陷入一个范围窄小的搜索空间，从而使算法易于陷入局部最优

ACS模型的局部信息素更新规则

在前面对AS的介绍中我们曾提到过顺序构建和并行构建两种途径构建方式

*对于AS算法，不同的路径构建方式不会影响算法的行为。

*但对于ACS，由于信息素局部更新规则的引入，两种路径构建方式会造成算法行为的区别，通常我们选择让所有蚂蚁并行地工作



ACS 中的顺序构建与并行构建

ACS模型小结

*在实际应用中，研究者们遇到的大多数问题是小规模优化问题，即变量空间的范围较小，此时ACS模型和AS模型都能取得令人满意的结果。但ACS模型较为复杂，其不像AS模型那样易于实现，所以对于小规模优化问题而言，一般采用AS模型来求解。

*但当优化问题的设计变量的维度很高（城市数量很多）时，如75个城市的TSP问题，ACS模型较AS模型更快地得到最优解。这是由于ACS模型在选择城市时通过转换规则的改变提高了算法的搜索能力；同时，采用局部信息素更新和全局信息素更新相结合的方式，有利于发现更多潜在的最优解，进而提高了算法的搜索质量

蚁群优化算法的改进

- 蚁群系统 (Ant Colony System, ACS)
- 最大-最小蚂蚁系统 (MAX-MIN Ant System, MMAS)

最大-最小蚂蚁系统 (MAX-MIN Ant System, MMAS)

*德国学者施蒂茨勒 (*Stutzle*) 和霍斯 (*Hoos*) 在1997年提出MMAS模型，其基本结构与AS模型类似。

MMAS模型对AS模型主要进行了以下4项改进

- 1.交替精英策略。它只允许一代中最好的或者目前为止最好的一只蚂蚁在本次循环后释放信息素，而其他路径只进行信息素挥发操作（但可能导致收敛到局部最优解）
- 2.限制区间策略。每条边上的信息量被限制在一个特定的范围 $[\tau_{\min}, \tau_{\max}]$ 内，超出这个范围的值将被强制设为 τ_{\min} 或 τ_{\max} ，这样可以有效地避免由于某条路径上的信息量远大于其他路径而导致所有的蚂蚁都集中到同一条路径上以及搜索过程停滞不前

最大-最小蚂蚁系统 (MAX-MIN Ant System, MMAS)

3.信息素平滑机制。各条边上的信息量初始化为区间上界值 τ_{\max} ，通过信息素更新使较差路径上的信息素逐渐减少，同时，信息素挥发系数取较小值，以保障不同路径上的信息素挥发不会太快，进而使算法在搜索初期具有较强的探索能力

4.当认为算法已经停滞不前或者一定数量的迭代之后没有更好的解会产生时，将所有边上的信息素重新初始化

MMAS：交替精英策略

标准蚂蚁系统算法与最大最小蚂蚁系统之间的首要差别是在信息素的更新方式：

$$\left. \begin{array}{l} \text{标准蚂蚁系统: } \tau_{ij} \leftarrow (1-\rho) \tau_{ij} + \sum_{k=1}^N \Delta \tau_{ij}^{(k)} \end{array} \right\}$$

$$\left. \begin{array}{l} \text{最大最小蚂蚁系统: } \tau_{ij} \leftarrow (1-\rho) \tau_{ij} + \Delta \tau_{ij}^{(best)} \end{array} \right\}$$

$$\text{式中, } \Delta \tau_{i,j}^{best} = \begin{cases} 1/L^{best}, & (i,j) \in T^{best} \\ 0, & \text{其他} \end{cases}$$

T^{best} 表示到目前为止找到的最优路径或在本次迭代中找到的最优路径

L^{best} 表示到目前为止找到的最优解或在本次迭代中找到的最优解

MMAS：交替精英策略

如用到目前为止找到的最优解更新信息素，则 $\Delta\tau_{i,j}^{best} = 1/L^{gb}$,

L^{gb} 表示到目前为止找到的最优解

如用本次迭代找到的最优解更新信息素，则 $\Delta\tau_{i,j}^{best} = 1/L^{ib}$,

L^{ib} 表示本次迭代找到的最优解

*仅由每一代中最好的蚂蚁（或目前最好的蚂蚁）添加信息素。这样会让探索减少，对已知的最好解的开发增加。

MMAS：交替精英策略

*当对信息素进行更新时，通过使用 L^{gb} 可以使算法较快地收敛，但可能使搜索解的范围受到限制，从而导致算法易于陷入局部收敛

*通过使用 L^{ib} 可避免上述情况发生。因为一次迭代的最优解会随着迭代次数的不同而显著变化，并会有较大数量的解得到增强

*可以使用混合策略，如使用 L^{ib} 作为信息素更新的默认值，而仅在固定的迭代次数间隔内使用 L^{gb}

MMAS：交替精英策略

*研究表明，对于小规模的 TSP 问题，使用本次迭代找到的最优解较好，而对于大规模的 TSP 问题（数百个城市），则逐渐增加使用目前为止找到的最优解进行信息素更新的频率，可以取得较好的结果

*大量实验表明， $MMAS$ 模型的效率和求解能力都要明显优于基本的 AS 模型

MMAS: 限制区间策略

最大最小蚂蚁系统 (*Max-Min Ant System*) 的第二个特征: 信息素的量有上下界。这样会有相反的效果, 也就是会让探索增加, 即使是最差的行程也有非零的信息素的量, 最好的行程的信息素再多也不能完全支配蚂蚁决策。

*MMAS*和*AS*两者之间的一个差别是在更新 τ_{ij} 之后, 最大最小蚂蚁系统执行下面的式子

$$\begin{aligned}\tau_{ij} &\leftarrow \max \left\{ \tau_{ij}, \tau_{\min} \right\} \\ \tau_{ij} &\leftarrow \min \left\{ \tau_{ij}, \tau_{\max} \right\}\end{aligned}$$

其中 τ_{\min} 和 τ_{\max} 根据具体的优化问题调整

MMAS: 限制区间策略

有不同的方式可以让最大最小蚂蚁系统更一般化。

*例如，不只是最好的蚂蚁留下信息素，可以让最好的 M 只蚂蚁留下信息素，这里 M 是一个可调参数；

*或者允许第 m 只最好的蚂蚁以概率 p_m 留下信息素，这里 p_m 随费用的增大而减小。假定我们想在优化过程的初始阶段更多地探索，结束阶段更多地开发，可以随着代数的增加增大 $(\tau_{\max} - \tau_{\min})$

-
- 蚂蚁觅食的秘密
 - 蚁群优化算法理论
 - 蚁群优化算法的数学模型及算法流程
 - 举例说明蚁群优化算法的工作机理
 - 蚁群优化算法的改进
 - 蚁群优化算法的参数设置及影响

蚁群优化算法的关键参数设置

信息素启发式因子 α ：影响算法的搜索能力

- 1.代表信息量对是否选择当前路径的影响程度，即反映蚂蚁在运动中所积累的信息量在指导蚁群搜索中的相对重要程度
2. α 的大小反映了蚁群在路径搜索中随机性因素作用的强度
3. α 值越大，蚂蚁在选择以前走过的路径的可能性就越大，搜索的随机性就会减弱，易使蚁群的搜索过早陷入局部最优解

蚁群优化算法的关键参数设置

4. α 的值越小，最邻近城市被选中的概率越大，蚂蚁越注重“眼前利益”。 $\alpha=0$ 时，算法等同于随机贪婪算法

5. 根据经验， α 取值范围一般为 $[1, 4]$ 时，蚁群算法的综合求解性能较好

6. *Dorigo* 等人的实验表明，在各类ACO 算法中设置 $\alpha = 1$ ， $\beta = 2 \sim 5$ 比较合适

蚁群优化算法的关键参数设置

期望启发因子 β ：决定算法搜索的导向

1. β 的大小反映了蚁群在搜索最优路径的过程中的先验性和确定性因素的作用强度
2. β 值越小，蚂蚁越倾向于根据信息素浓度确定路径，算法收敛越快。
 $\beta = 0$ 时，构建出的最优路径与实际目标有着较大差异，算法的性能比较糟糕
3. β 值越大，蚂蚁在某个局部点上选择局部最短路径的可能性就越大，虽然这个时候算法的收敛速度得以加快，但蚁群搜索最优路径的随机性减弱，而此时搜索易于陷入局部最优解
4. 根据经验， β 取值范围一般为 $[3, 5]$ 时，蚁群算法的综合求解性能较好

蚁群优化算法的关键参数设置

实际上信息素启发式因子 α 和期望启发因子 β 是一对关联性很强的参数：

a.) 蚁群算法的全局寻优性能，首先要求蚁群的搜索过程必须要有很强的随机性；

b.) 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一定的确定性；

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，算法要获得最优解，就必须在这两者之间选取一个平衡点，只有正确选定它们之间的搭配关系，才能避免在搜索过程中出现过早停滞或陷入局部最优等情况的发生

蚁群优化算法的参数设置

信息素挥发因子 ρ ：反映了蚂蚁个体之间相互影响的强弱

- 1.信息素挥发因子 ρ 的取值范围应该是 $[0, 1]$ 之间的一个数，表示信息素的蒸发程度, $1-\rho$ 表示信息素持久性系数
- 2.信息素挥发因子 ρ 大小的选择将直接影响到整个蚁群算法的全局搜索能力和收敛速度
3. ρ 较大时，信息素挥发速率大，说明路径上的信息素挥发相对变多，那些从未被蚂蚁选择过的边上的信息素急剧减小到接近0，降低了算法的全局搜索能力

蚁群优化算法的参数设置

4. ρ 较小时，路径上遗留的信息量过量，减弱了正反馈的作用，算法收敛速度变慢

5. 因此， ρ 的选取值要综合考虑。从最短路径的选取和迭代次数可以知道，信息素挥发因子在由小到大的变化中，最短路径的获取和收敛性相对来说是比较好的

6. Dorigo等人的实验表明，对于AS和EAS， $\rho = 0.5$ ；

对于AS_{rank}， $\rho = 0.1$ ；对于MMAS， $\rho = 0.2$ ；

对于ACS， $\rho = 0.1$ ，算法的综合性能较高

7. 经验数据： $0.1 \leq \rho \leq 0.99$

蚁群优化算法的参数设置

蚂蚁数量 m ：影响着算法的搜索能力和计算量

1. 蚁群在搜索过程中之所以表现出复杂有序的行为，是因为个体之间的信息交流与相互协作起着至关重要的作用
2. 蚂蚁数量增多，可以提高蚁群算法的全局搜索能力以及算法的稳定性
3. 蚂蚁数量过多时，每轮迭代的计算量大、且被搜索过的路径上信息素变化比较平均，信息正反馈的作用不明显，此时算法的全局随机搜索能力得到增强，但收敛速度减慢

蚁群优化算法的参数设置

4.蚂蚁数量过少时，特别是当问题规模很大时，会使那些从未被搜索到的解（路径）上的信息素减小到接近于0，搜索的随机性减弱，虽然收敛速度加快了，但会使算法的全局寻优能力会变得十分糟糕，算法的稳定性差，容易出现过早停滞的现象

5.参数设定可以考虑 $n = 1.5m$ ，通常取值范围 $m = 10 \sim 50$

6.*Dorigo*等人的实验表明，在用AS、*EAS*、 AS_{rank} 和*MMAS*求解TSP问题时， m 取值等于城市数目 n 算法有较好的性能；而对于ACS， $m = 10$ 比较合适

蚁群优化算法的参数设置

信息素强度 Q 对算法性能的影响:

1.在蚁群算法中，各个参数的作用实际上是紧密联系的，其中对算法性能起着主要作用的是信息启发式因子 α 、期望启发式因子 β 和信息素挥发因子 ρ 这3个参数，总信息量 Q 对算法性能的影响有赖于上述3个参数的选取，以及算法模型的选取

2.在 $ant - cycle$ 模型和 $ant - quantity$ 模型中，总信息量 Q 所起的作用显然是有很大差异的，即随着问题规模的不同，其影响程度也将不同

蚁群优化算法的参数设置

3. 相关人员研究结果表明：总信息量 Q 对 $ant - cycle$ 模型蚁群算法的性能没有明显的影响。因此，在算法参数的选择上，参数 Q 不必作特别的考虑，可以任意选取

4. 经验数据： $10 \leq Q \leq 10000$

蚁群优化算法的参数设置

初始信息素量 τ_0 ：决定算法在初始化阶段的探索能力，影响算法的收敛速度

1. τ_0 太小，未被蚂蚁选择过的边上信息素太少，搜索方向容易受到蚂蚁初始选择路径的影响，蚂蚁很快就全部集中于一条局部最优的路径上，算法容易早熟

2. τ_0 太大，需要等待很多次循环，直到初始信息素量挥发一定程度，蚂蚁新增加的信息素能对搜索方向产生影响，因此新增加信息素对搜索方向的引导能力增长得十分缓慢，算法收敛慢

蚁群优化算法的参数设置

3.一种好的信息素初始化方法是将信息素的初始值设为略高于每一次迭代中蚂蚁释放信息素大小的期望值。可以用以下方式粗略地估算这个初始值： $\forall (i, j) \in L, \tau_{ij} = m / L^{nn}$ ，其中， m 为蚂蚁的个数， L^{nn} 为用最近邻启发式方法构造的路径的长度

4.*Dorigo*等人的实验推荐：对于AS, $\tau_0 = m / L^{nn}$;

对于EAS, $\tau_0 = (e + m) / (\rho L^{nn})$; 对于AS_{rank}, $\tau_0 = 0.5r(r-1) / (\rho L^{nn})$;

对于MMAS, $\tau_0 = 1 / (\rho L^{nn})$; 对于ACS, $\tau_0 = 1 / (n L^{nn})$

蚁群优化算法的参数设置

终止条件: 决定算法运行何时结束，由具体的应用和问题本身确定

- 1.将最大循环数设定为500、1000、5000，或者最大的函数评估次数，等等
- 2.也可以使用算法求解得到一个可接受的解作为终止条件
- 3.算法的运行时间达到预先指定的最大值
- 4.算法以探索的路径数目达到最大值
- 5.或者算法陷入停顿，当算法在很长一段迭代中没有得到任何改善时，可以终止算法

蚁群优化算法的参数设置

Box 3.1

Parameter Settings for ACO Algorithms without Local Search

Our experimental study of the various ACO algorithms for the TSP has identified parameter settings that result in good performance. For the parameters that are common to almost all the ACO algorithms, good settings (if no local search is applied) are given in the following table.

ACO algorithm	α	β	ρ	m	τ_0
AS	1	2 to 5	0.5	n	m/C^m
EAS	1	2 to 5	0.5	n	$(e+m)/\rho C^m$
AS _{rank}	1	2 to 5	0.1	n	$0.5r(r-1)/\rho C^m$
MMAS	1	2 to 5	0.02	n	$1/\rho C^m$
ACS	—	2 to 5	0.1	10	$1/n C^m$

Here, n is the number of cities in a TSP instance. All variants of AS also require some additional parameters. Good values for these parameters are:

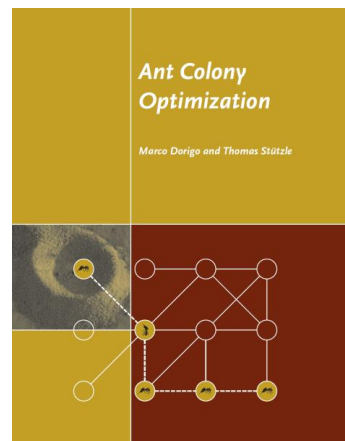
EAS: The parameter e is set to $e = n$.

AS_{rank}: The number of ants that deposit pheromones is $w = 6$.

MMAS: The pheromone trail limits are $\tau_{max} = 1/\rho C^{bs}$ and $\tau_{min} = \tau_{max}(1 - \sqrt[3]{0.05})/((avg - 1) \cdot \sqrt[3]{0.05})$, where avg is the average number of different choices available to an ant at each step while constructing a solution (for a justification of these values see Stützle & Hoos (2000)). When applied to small TSP instances with up to 200 cities, good results are obtained by using always the iteration-best pheromone update rule, while on larger instances it becomes increasingly important to alternate between the iteration-best and the best-so-far pheromone update rules.

ACS: In the local pheromone trail update rule: $\xi = 0.1$. In the pseudorandom proportional action choice rule: $q_0 = 0.9$.

It should be clear that in individual instances, different settings may result in much better performance. However, these parameters were found to yield reasonable performance over a significant set of TSP instances.



小结

- 蚁群优化算法是继遗传算法、模拟退火算法、禁忌搜索算法等计算智能算法之后又一种应用于组合优化问题的启发式随机搜索算法
- 蚁群优化算法不仅能够进行智能搜索、全局优化，而且具有稳健性强、正反馈性显著、可分布式计算、易与其他算法结合等优点
- 蚁群算法通过模拟蚂蚁觅食的过程，是一种天然的解决离散组合优化问题的方法，在解决典型组合优化问题，如旅行商（TSP）问题、车辆路径问题（VRP）、车间作业调度问题（JSP）时具有明显的优越性

小结

- 蚁群优化算法的研究已由当初单一的TSP领域渗透到了多个应用领域，由解决一维静态优化问题发展到解决多维动态组合优化问题，由离散域范围内研究逐渐拓展到连续域范围内研究
- 蚁群优化算法的硬件实现上取得了突破性进展
- 蚁群优化算法的模型改进及与其他仿生优化算法的融合方面也取得了相当丰富的研究进展
- 蚁群优化算法展现出前所未有的勃勃生机，并已经成为一种完全可与遗传算法相媲美的仿生优化算法