



粒子速度更新: $\mathbf{v}_i(t) \rightarrow \mathbf{v}_i(t+1)$

$$\begin{aligned}\mathbf{v}_i(t+1) &= \text{Inertial} + \text{Cognitive} + \text{Social} \\ \mathbf{v}_i(t+1) &= w \times \mathbf{v}_i(t) + \\ &\quad c_1 \times \text{rndreal}_1() \times (\text{PBest}_i - \mathbf{x}_i(t)) + \\ &\quad c_2 \times \text{rndreal}_2() \times (\text{GBest} - \mathbf{x}_i(t))\end{aligned}$$

粒子位置更新: $\mathbf{x}_i(t) \rightarrow \mathbf{x}_i(t+1)$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$



智能优化技术

蚁群优化 (Ant Colony Optimization)

龚文引 (教授、博士生导师)

中国地质大学 (武汉) 计算机学院

1. 大纲

算法简介

蚂蚁系统

简单示例

小结

2. 算法简介

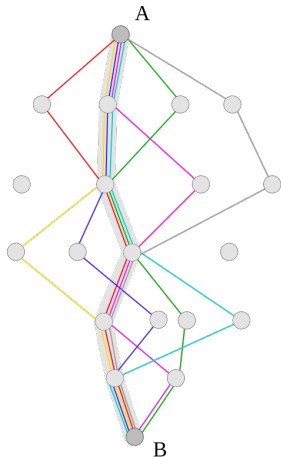
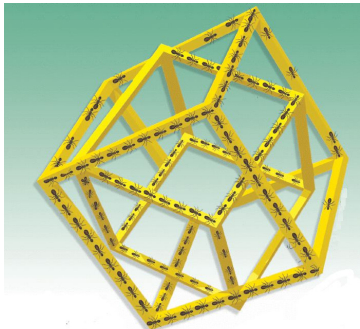
算法简介

蚂蚁系统

简单示例

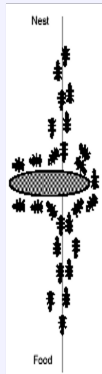
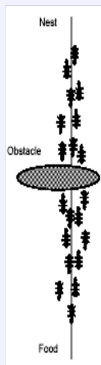
小结

2. 算法简介



2. 算法简介

蚂蚁的自然行为



2. 算法简介

蚁群优化（Ant Colony Optimization, 简称 ACO）起源

- 提出：
 - 20 世纪 90 年代意大利学者 M. Dorigo, V. Maniezzom, A. Colorni 等
- 启发：
 - 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
- 性质：
 - 属于群体智能（Swarm Intelligence）算法，是一种全局搜索方法
 - 能较好求解如 TSP 问题、分配问题、job-shop 调度问题等组合优化问题

- M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997.
- M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, Nov. 2006.

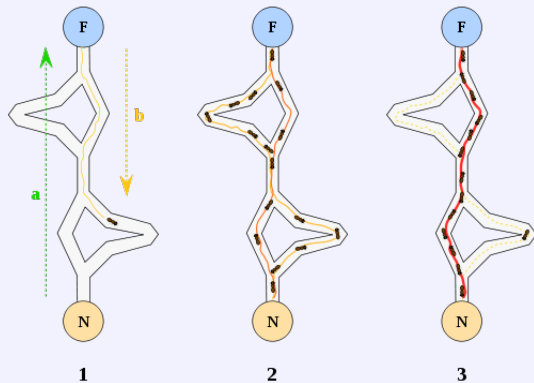
2. 算法简介

原理

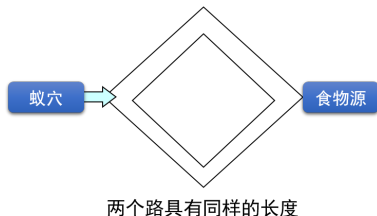
- 外激素 (pheromone)或信息素
 - 蚂蚁在运动过程中，在它所经过的路径上留下一种信息传递的物质，即信息素。信息素会随时间挥发
 - 蚂蚁在运动过程中能够感知这种物质，并以此指导自己的运动方向
 - 路径越长，释放的激素浓度越低
 - 蚂蚁随机选择路径，但激素浓度较高的路径被选中概率就会相对较大
- 信息正反馈
 - 某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大
 - 最优路径上的激素浓度越来越大，而其它的路径上激素浓度却会随着时间的流逝而消减
 - 最终整个蚁群会找出最优路径

2. 算法简介

原理



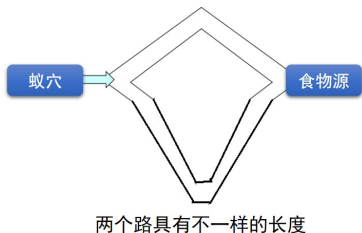
2. 算法简介



双桥实验 1

- 起初两条分支上不存在信息素，蚂蚁以相同的概率进行选择
- 随机波动的出现，选择某一条分支的蚂蚁数量可能比另外一条多
- 实验最终结果：
 - 几乎所有的蚂蚁都会选择同一分支
 - 但多次试验后发现选择两条路径的概率几乎相等

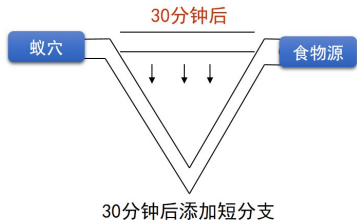
2. 算法简介



双桥实验 2

- 起初两条分支上不存在信息素，蚂蚁以相同的概率进行选择
- 短分支上的信息素积累速度比长分支的快
- 实验最终结果：几乎所有的蚂蚁都会选择较短的分支，只有很小比例的蚂蚁会选择较长的分支

2. 算法简介



双桥实验 3

- 实验最终结果：除了极少的蚂蚁选择较短的分支以外，整个群体几乎都困在较长的分支上
- 长分支上的信息素浓度高，而信息素的蒸发速度过于缓慢

2. 算法简介

双桥实验总结

- ① 选择路径是一个概率随机过程，启发式信息多以及信息浓度大的路径被选中概率更大
- ② 被选中路径的信息素会得到增强
- ③ 信息素会不断的蒸发
- ④ 路径探索也是必需的，否则容易陷入局部最优

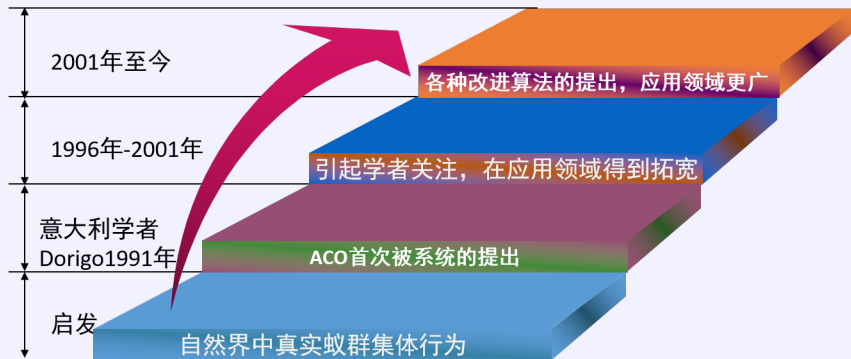
2. 算法简介

蚁群觅食现象和 ACO 的基本定义对照表

蚁群觅食现象	蚁群优化算法
蚁群	搜索空间的一组有效解（种群规模 m ）
觅食空间	问题的搜索空间（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	问题的最优解

2. 算法简介

ACO 历史进展



2. 算法简介

ACO 算法进展



3. 蚂蚁系统

算法简介

蚂蚁系统

简单示例

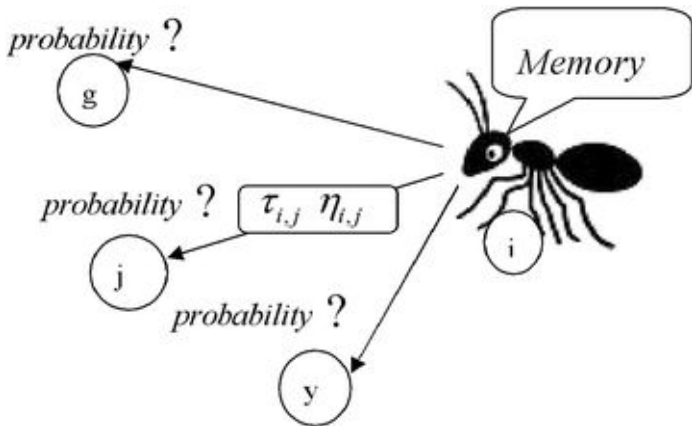
小结

TSP 问题

已知有 n 个城市的集合 $C_n = \{c_1, c_2, \dots, c_n\}$, 任意两个城市之间均有路径连接, d_{ij} ($i, j = 1, 2, \dots, n$) 表示城市 i 与 j 之间的距离. 旅行商问题就是需要寻找这样的一种周游方案: 周游路线从某个城市出发, 经过每个城市一次且仅一次, 最终回到出发城市, 使得周游的路线总长度最短.

第一个 ACO-蚂蚁系统, 就是以 NP 难的 TSP 问题作为应用实例提出的.

M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997.



重要步骤

AS 算法对 TSP 的求解流程主要有两大重要步骤：

- ① 路径构建
- ② 信息素更新

路径构建

定义：AS 中的随机比例规则

对每只蚂蚁 k ，路径记忆向量 \mathbf{R}^k (按照访问顺序记录了所有 k 已经经过的城市序号)。设蚂蚁 k 当前所在的城市为 i ，则其选择城市 j 作为下一个访问对象的概率为：

$$p^k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta}, & j \in J_k(i) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中； $J_k(i)$ 表示从城市 i 可以直接到达的且又不在蚂蚁访问过的城市序列 \mathbf{R}^k 中的城市集合； $\eta(i, j)$ 是一个启发式信息，通常由 $\eta(i, j) = 1/d_{ij}$ 直接计算； $\tau(i, j)$ 表示边 (i, j) 上的信息量。

路径构建

$$p^k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta}, & j \in J_k(i) \\ 0, & \text{otherwise} \end{cases}$$

由公式知，**长度越短、信息素浓度越大的路径被蚂蚁选择的概率越大。**

α 和 β 是两个预先设置的参数，用来控制**信息素浓度**与**启发式信息**的权重。

- 当 $\alpha = 0$ 时，算法演变成了传统的随机贪婪算法，最邻近城市被选中的概率最大；
- 当 $\beta = 0$ 时，蚂蚁完全只根据信息素浓度确定路径，算法快速收敛，构建出的最优路径与实际目标有较大的差异。

信息素更新

- 初始化: $\tau_0 = m/L^{nn}$
- 更新公式:

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \sum_{k=1}^m \Delta\tau^k(i, j) \quad (2)$$

$$\Delta\tau^k(i, j) = \begin{cases} (L_k)^{-1}, & (i, j) \in \mathbf{R}^k \\ 0, & \text{otherwise} \end{cases}$$

其中, m 是蚂蚁的个数; L^{nn} 是由贪心算法构造的路径的长度; $\rho \in (0, 1]$ 是信息素的蒸发率, 通常设置为 $\rho = 0.5$; $\Delta\tau^k(i, j)$ 是第 k 只蚂蚁在它经过的边上释放的信息素量; L_k 表示路径的长度, 它是 \mathbf{R}^k 中所有边的长度和.

蚂蚁系统（Ant System, AS）的三个版本

信息素的三种形式：

① 蚂蚁圈：

$$\Delta\tau^k(i, j) = \begin{cases} Q/L_k, & \text{if ant } k \text{ used edge}(i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases}$$

② 蚂蚁数量：

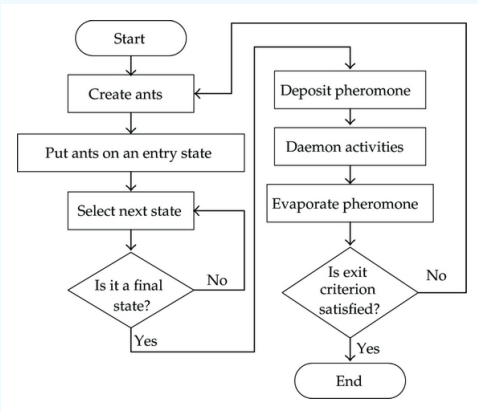
$$\Delta\tau^k(i, j) = \begin{cases} Q/d_{ij}, & \text{if ant } k \text{ used edge}(i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases}$$

③ 蚂蚁密度：

$$\Delta\tau^k(i, j) = \begin{cases} Q, & \text{if ant } k \text{ used edge}(i, j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases}$$

其中, Q 为常数, 表示蚂蚁循环一次所释放的信息素总量; L_k 为第 k 只蚂蚁经过路径的长度; d_{ij} 为第 i 个城市到第 j 个城市的长度.

流程



改进版本

- 精华蚂蚁系统
- 基于排列的蚂蚁系统
- 最大最小蚂蚁系统
- 蚁群系统
- 连续正交蚁群系统
- ...

D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens, "Classification with Ant Colony Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 651-665, 2007.

4. 简单示例

算法简介

蚂蚁系统

简单示例

小结

4. 简单示例

问题

给出用蚁群算法求解一个四城市的 TSP 问题的执行步骤，四个城市 A、B、C、D 之间的距离矩阵如下：

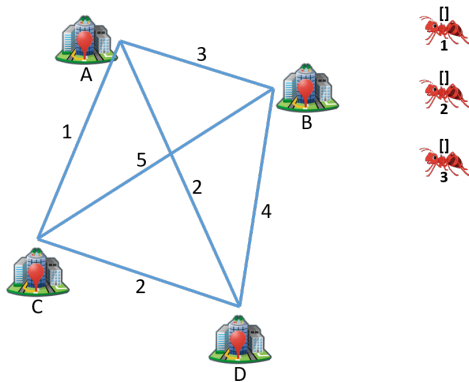
$$W = d_{ij} = \begin{bmatrix} \infty & 3 & 1 & 2 \\ 3 & \infty & 5 & 4 \\ 1 & 5 & \infty & 2 \\ 2 & 4 & 2 & \infty \end{bmatrix}$$

算法参数设置

假设蚂蚁种群的规模 $m = 3$, 参数 $\alpha = 1, \beta = 2, \rho = 0.5$.

4. 简单示例

A simple TSP example



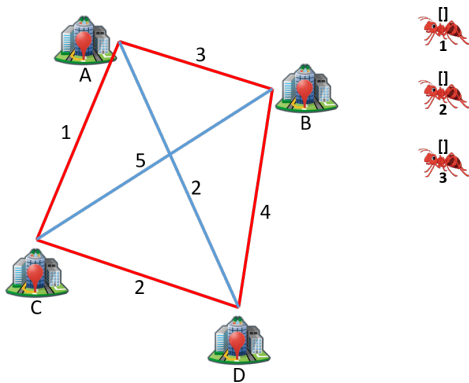
4. 简单示例

Step 1: 初始化

- 1) 首先用贪心算法得到路径 (ACDBA);
- 2) 计算 $L^{nn} = f(\text{ACDBA}) = 1 + 2 + 4 + 3 = 10$;
- 3) 求得 $\tau_0 = m/L^{nn} = 3/10 = 0.3$;
- 4) 初始化所有边上的信息素 $\tau(i, j) = \tau_0$.

4. 简单示例

初始化



4. 简单示例

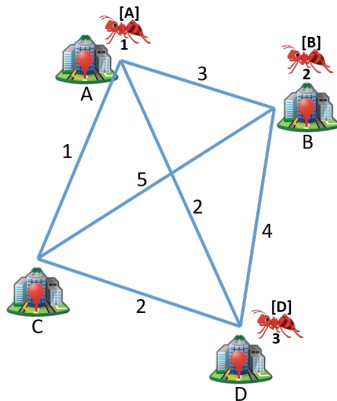
Step 2.1: 放置蚂蚁

为每只蚂蚁随机选择出发的城市，假设：

- 1) 蚂蚁 1 选择城市 A;
- 2) 蚂蚁 2 选择城市 B;
- 3) 蚂蚁 3 选择城市 D.

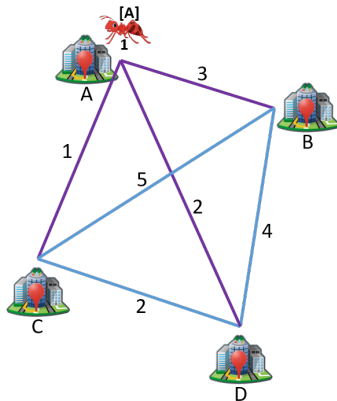
4. 简单示例

放置蚂蚁



4. 简单示例

移动蚂蚁-1



4. 简单示例

路径构建

$$p^k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta}, & j \in J_k(i) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

4. 简单示例

Step 2.2-a: 移动蚂蚁

为每只蚂蚁选择下一城市：仅以蚂蚁 1 为例，当前城市 $i = A$ ，则可访问城市集合为 $J_1(i) = \{B, C, D\}$ 。计算蚂蚁 1 选择 B, C, D 作为下一访问城市的概率：

$$A \rightarrow \begin{cases} B: & \tau_{AB}^{\alpha} \cdot \eta_{AB}^{\beta} = 0.3^1 \cdot (1/3)^2 = 0.033 \\ C: & \tau_{AC}^{\alpha} \cdot \eta_{AC}^{\beta} = 0.3^1 \cdot (1/1)^2 = 0.3 \\ D: & \tau_{AD}^{\alpha} \cdot \eta_{AD}^{\beta} = 0.3^1 \cdot (1/2)^2 = 0.075 \end{cases}$$

$$\begin{cases} p(B) &= 0.033 / (0.033 + 0.3 + 0.075) &= 0.081 \\ p(C) &= 0.3 / (0.033 + 0.3 + 0.075) &= 0.74 \\ p(D) &= 0.075 / (0.033 + 0.3 + 0.075) &= 0.18 \end{cases}$$

4. 简单示例

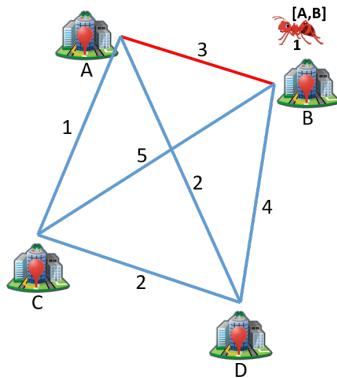
Step 2.2-b: 移动蚂蚁

用轮盘赌法选择下一城市，假设产生的随机数 $r = \text{rndreal}(0, 1) = 0.05$ ，则

- 蚂蚁 1 将会选择城市 B;
- 蚂蚁 2 将会选择城市 D;
- 蚂蚁 3 将会选择城市 A.

4. 简单示例

移动蚂蚁-2



4. 简单示例

Step 2.3: 继续移动蚂蚁

当前蚂蚁 1 所在城市 $i = B$ ，路径记忆向量 $\mathbf{R}^1 = (AB)$ ，则可访问的城市集合为 $J_1(i) = \{C, D\}$ 。计算蚂蚁 1 选择 C, D 作为下一访问城市的概率：

$$B \rightarrow \begin{cases} C : & \tau_{BC}^\alpha \cdot \eta_{BC}^\beta = 0.3^1 \cdot (1/5)^2 = 0.012 \\ D : & \tau_{BD}^\alpha \cdot \eta_{BD}^\beta = 0.3^1 \cdot (1/4)^2 = 0.019 \end{cases}$$

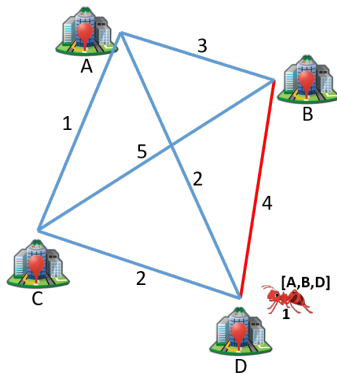
$$\begin{cases} p(C) & = & 0.012 / (0.012 + 0.019) & = & 0.39 \\ p(D) & = & 0.019 / (0.012 + 0.019) & = & 0.61 \end{cases}$$

用轮盘赌法选择下一城市，假设产生的随机数 $r = \text{rndreal}(0, 1) = 0.67$ ，则

- 蚂蚁 1 将会选择城市 D;
- 蚂蚁 2 将会选择城市 C;
- 蚂蚁 3 将会选择城市 C.

4. 简单示例

移动蚂蚁-3



4. 简单示例

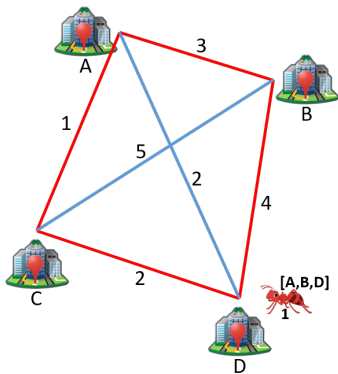
Step 2.4: 完成路径创建

实际此时路径已经创建完毕,

- 蚂蚁 1 创建的路径为 (ABDCA);
- 蚂蚁 2 创建的路径为 (BDCAB);
- 蚂蚁 3 创建的路径为 (DACBD).

4. 简单示例

完成路径创建



4. 简单示例

信息素更新

- 更新公式:

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \sum_{k=1}^m \Delta\tau^k(i, j) \quad (4)$$

$$\Delta\tau^k(i, j) = \begin{cases} (L_k)^{-1}, & (i, j) \in \mathbf{R}^k \\ 0, & \text{otherwise} \end{cases}$$

4. 简单示例

Step 3: 信息素更新

1) 计算每只蚂蚁创建的路径长度:

$$L_1 = 3 + 4 + 2 + 1 = 10$$

$$L_2 = 4 + 2 + 1 + 3 = 10$$

$$L_3 = 2 + 1 + 5 + 4 = 12$$

2) 更新每条边上的信息素:

$$\tau(A, B) = (1 - \rho)\tau(A, B) + \sum_{k=1}^3 \Delta\tau^k(A, B) = 0.5 \times 0.3 + (1/10 + 1/10) = 0.35$$

$$\tau(A, C) = (1 - \rho)\tau(A, C) + \sum_{k=1}^3 \Delta\tau^k(A, C) = 0.5 \times 0.3 + (1/12) = 0.1583$$

...

同理即可依次计算出问题空间内所有边更新后的信息素量.

4. 简单示例

Step 4: 判断

如果满足结束条件，则输出最优结果，并结束程序；否则，转向 Step 2.1 继续执行。

4. 简单示例

课外作业

在网上下载蚁群优化算法相关程序，读懂，并尝试独立实现，思考以下问题：

- 对ACO的代码添加中文注释。

4. 简单示例

思考

- ACO 的产生与测试在何处？
- ACO 的搜索方向和搜索步长如何确定？
- ACO 如何实现信息共享？
- ACO 如何实现资源竞争？
- ACO 的勘探与开采如何实现？

5. 小结

算法简介

蚂蚁系统

简单示例

小结

5. 小结

本章小结

- ① ACO 简介
- ② 蚂蚁系统介绍
- ③ 简单示例

5. 小结

思考

在网上下载蚁群优化算法相关程序，读懂，并尝试独立实现，思考以下问题：

- ACO 求解较大规模 TSP 问题的性能如何？
- ACO 除了求解 TSP 问题外，求解其他组合优化问题的效果如何？

网络参考报告

在网上利用ACO 算法求解 TSP 问题的报告：

- <https://www.cnblogs.com/twzh123456/p/11798800.html>
- <https://www.cnblogs.com/cenyan/p/11798802.html>

5. 小结

进一步阅读资料

- <http://www.aco-metaheuristic.org/>
- M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997.
- M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, Nov. 2006.
- E. Bonabeau, M. Dorigo & G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, pp. 39-42, 2000.
- D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens, "Classification with Ant Colony Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 651-665, 2007.
- M. Wittlinger, R. Wehner, H. Wolf, "The Ant Odometer: Stepping on Stilts and Stumps," *Science*, vol. 312, pp. 1965-1967, 2006.
- S. Schwarz, M. Mangan, J. Zeil, Barbara Webb, A. Wystrach, "How Ants Use Vision When Homing Backward," *Current Biology*, vol. 27, no. 3, pp. 401-407, 2017.

6. 致谢

Thank you!

AUTHOR: GONG, Wenyin
ADDRESS: School of Computer Science,
China University of Geosciences,
Wuhan, 430074, China
E-MAIL: wygong@cug.edu.cn
HOMEPAGE: <http://grzy.cug.edu.cn/gongwenyin>