第八章　蚁群优化算法

一. 导言

1. 起源

- **Ant Colony Optimization (ACO)**
- 意大利学者**Dorigo**于**1992**年提出
- 灵感来源于蚂蚁在寻找食物过程中发现路径的行为

M. Dorigo. Optimization, learning and natural algorithms [D]. Italy: Politecnico diMilano, Department of Electronics, 1992.

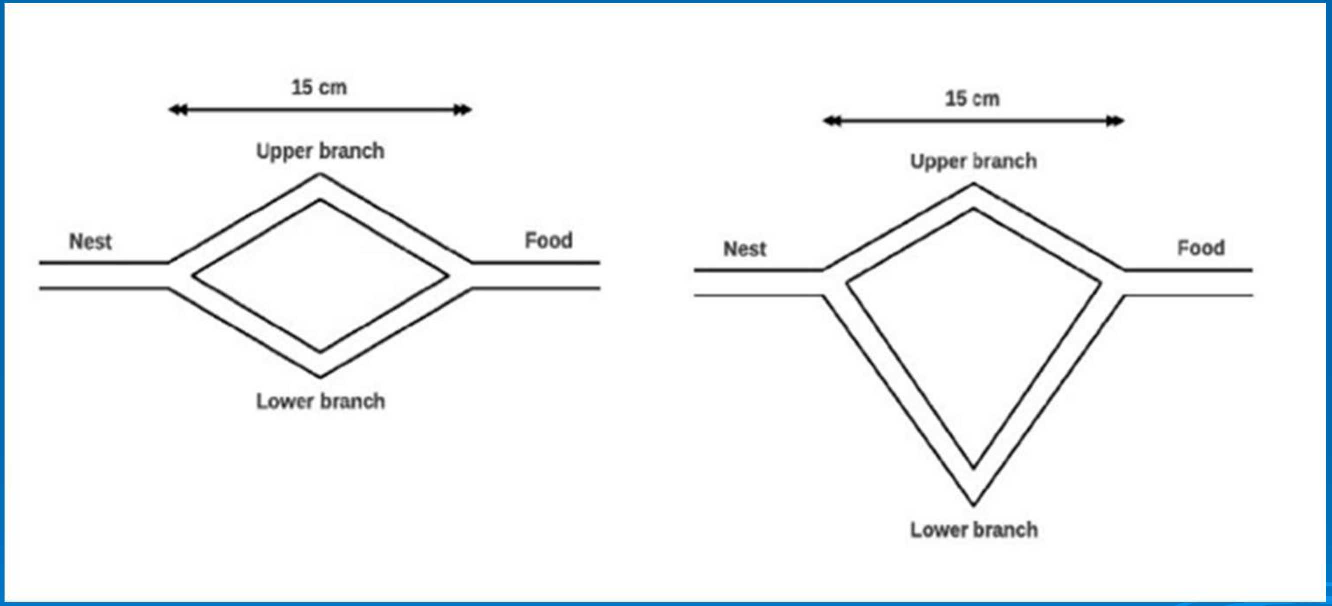
一. 导言

1. 起源

- 蚂蚁的社会行为—共识主动性(Stigmergy)
  - 真实蚂蚁几乎是全盲的
  - 蚂蚁可以通过一种特有分泌物（信息素）的气味进行间接通信
  - 当一只蚂蚁发现食物时，它就会在回家的路上留下一路的气味，其他的蚂蚁就会沿着这条路线去找食物，并不断地加强气味。如果食物被采集完了，没有蚂蚁再来，气味就会逐渐消散。

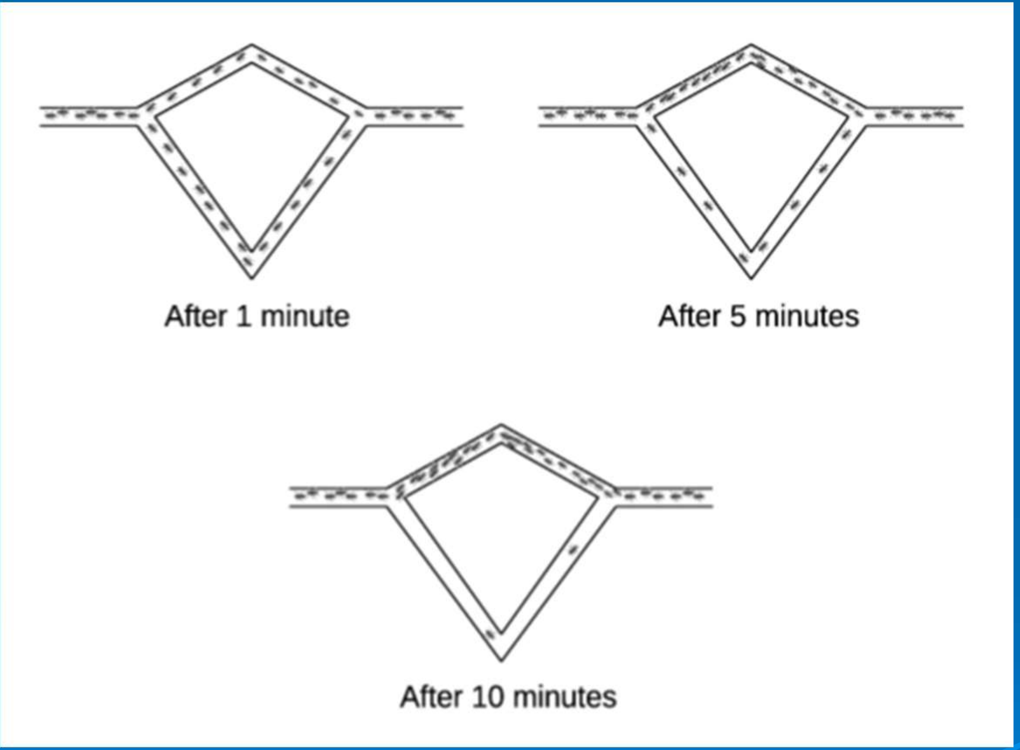
一. 导言

2. 一个小例子


15 cm
Upper branch
Nest
Food
Lower branch
15 cm
Upper branch
Nest
Food
Lower branch

一. 导言

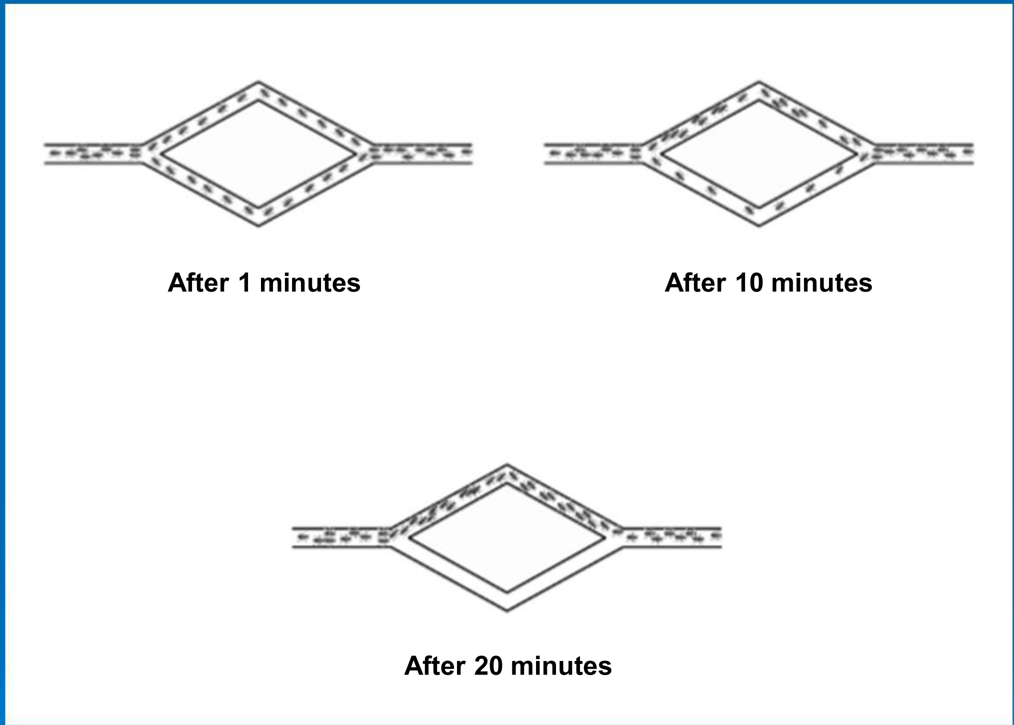
2. 一个小例子



不等分支时，蚂蚁很快聚集到短路径分支

一. 导言

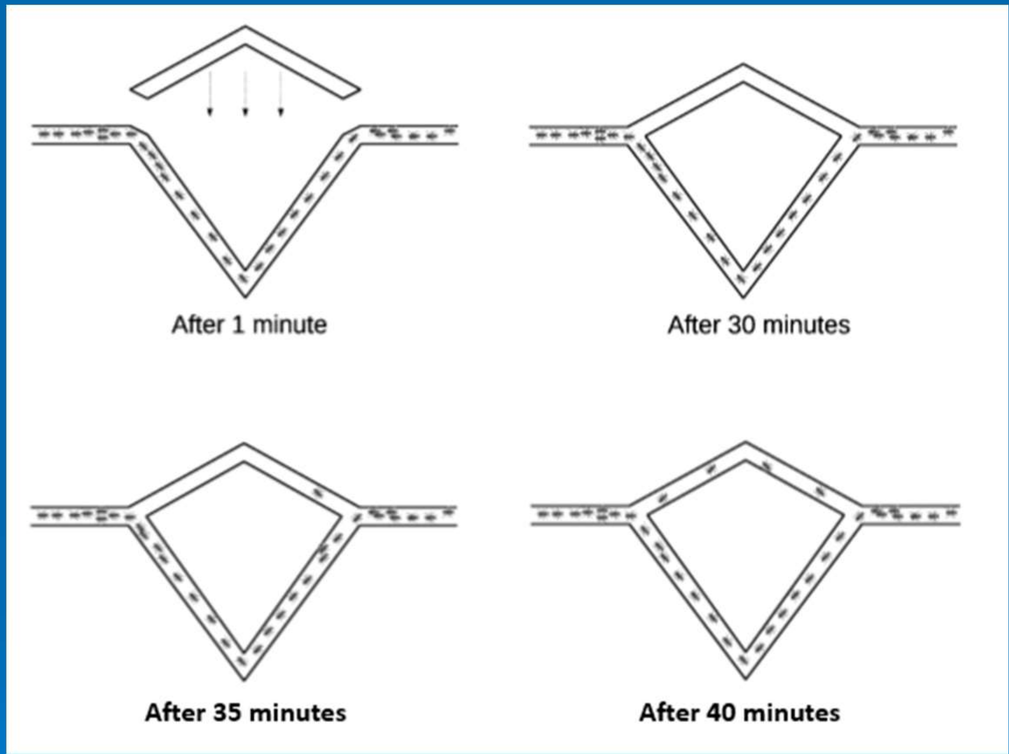
2. 一个小例子



相等分支时，蚂蚁最终聚集到一个分支

一. 导言

2. 一个小例子



收敛后提供最短分支时，最短分支仅是偶尔被选择

一. 导言

3. 基本思想

- 从真实蚂蚁到人工蚂蚁
  - 优化：能够发现最短路径
  - 探索能力：即使群体收敛于一条路径，仍然会有少量蚂蚁随机地选择别的路径
  - 开发能力：大部分蚂蚁更愿意选择并收敛于信息素浓度大的路径
  - 记忆性：蚂蚁移动时会用信息素标记走过的路径
  - 迭代：多次往返巢穴和食物源

二. 基本算法

1. **ACO metaheuristic**
   - **A swarm intelligence algorithm rather than evolutionary algorithm**
   - **Does not use any search operators inspired by evolution**
   - **Inspired by the collective behavior of ant colonies**
   - **A population of ants interacts locally with each other and with their environment**

二. 基本算法

1. ACO metaheuristic

- **Developed especially for discrete/combinatorial optimization problems**
- **The problem consists of finite set of components**
- **Initial application: TSP**
- **Consists of two main procedures: Construct Solutions (Forward Mode) and Pheromone Update (Backward Mode)**

二. 基本算法

2. ACO for TSP

- ACO can be applied to TSP easily
- Pheromone trails are associated with the arcs
- Initially all pheromone trails are equal
  - Based on the length of a tour generated by the nearest-neighbor heuristic
- Each ant is placed to a city randomly
- Ants construct a feasible solution by adding cities to its memory until all cities are visited

二. 基本算法

2. ACO for TSP

- The construction procedure is based on a probabilistic rule
- Uses existing pheromone trails and heuristic information
  - Heuristic information = the inverse of the cities' distance
- After all ants finish the construction procedure, the path is retraced and update the pheromone trails accordingly

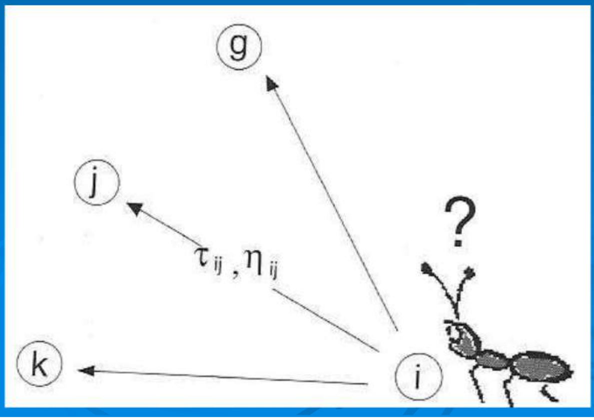
二. 基本算法

3. Construct Solutions

- **Ant k selects the next city j probabilistically from city i**

$$p_{ijk} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_{ik}}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \text{ if } j \in N_{ik}$$

- **τ_{ij} is the existing pheromone trail**
- **η_{ij} is the heuristic information**
- **N_{ik} is the neighborhood of unvisited cities for ant k ant city i**



二. 基本算法

3. Construct Solutions

- **The probabilistic rule is based on the roulette wheel technique**
- **The probability of choosing a particular city increases with the value of pheromone trails and heuristic information**
  - **When $\alpha = 0$, the closest cities are more likely to be selected**
  - **When $\beta = 0$, random cities are more likely to be selected**
- **Recommended values are $\alpha = 1$ and $\beta = 5$**

二. 基本算法

4. Pheromone Update (Evaporate)

- **With pheromone evaporation, a small amount of pheromone is deducted from all trails**

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \forall (i, j) \in A$$

  - **ρ is the evaporation constant**

- **Effect of pheromone evaporation**
  - **Helps ants to "forget" bad decisions (poor solutions) made in the past (previous iteration)**
  - **If an arc is not chosen by ants for a number of iterations, its associated pheromone value decreases exponentially**

二. 基本算法

4. Pheromone Update (Deposit)

- **Each ant updates its pheromone trails**
- **The amount of pheromone is proportional to the solution quality of each ant**
- **The higher the quality the more the pheromone**

$$\boldsymbol{\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ijk}, \forall(i,j) \in A}$$

- **m is the number of ants**
- **A is the set of arcs**
- **$\Delta\tau_{ijk}$ is the amount of pheromone to be deposited by ant k**

$$\Delta\tau_{ijk} = \frac{Q}{L_k}$$

三. 应用算例

1. ACO in Action

- TSP with 5 cities
- ACO with 2 ants

Distance Matrix

ID	1	2	3	4	5
1	-	6	3	8	7
2	6	-	4	5	2
3	3	4	-	3	1
4	8	5	3	-	8
5	7	2	1	8	-

三. 应用算例

2. Initial Phase

- Initial values of pheromone are set to 0.5

Heuristic Matrix

ID	1	2	3	4	5
1	-	0.16	0.33	0.12	0.14
2	0.16	-	0.25	0.20	0.50
3	0.33	0.25	-	0.33	1.00
4	0.12	0.20	0.33	-	0.12
5	0.14	0.50	1.00	0.12	-

Pheromone Matrix

ID	1	2	3	4	5
1	-	0.5	0.5	0.5	0.5
2	0.5	-	0.5	0.5	0.5
3	0.5	0.5	-	0.5	0.5
4	0.5	0.5	0.5	-	0.5
5	0.5	0.5	0.5	0.5	-

三. 应用算例

3. First Iteration

- **Ant 1: How to construct a tour (Forward Mode)**
  - **Step 1: start from city 1 randomly;**
  - **Step 2: select the next city to visit from city 1;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
1	1	2	0.5	0.16	0.08	0.21
1	1	3	0.5	0.33	0.165	0.44
1	1	4	0.5	0.12	0.06	0.16
1	1	5	0.5	0.14	0.07	0.19
Sum					0.375	1.00

Here, city 3 is chosen by Ant 1 as the next city to visit from city 1

三. 应用算例

3. First Iteration

- **Ant 1：How to construct a tour (Forward Mode)**
  - **Step 3: select the next city to visit from city 3;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
1	3	2	0.5	0.25	0.125	0.16
1	3	4	0.5	0.33	0.165	0.21
1	3	5	0.5	1.00	0.5	0.63
Sum					0.79	1.00

Here, city 5 is chosen by Ant 1 as the next city to visit from city 3

三. 应用算例

3. First Iteration

- **Ant 1： How to construct a tour (Forward Mode)**
  - **Step 4: select the next city to visit from city 5;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
1	5	2	0.5	0.50	0.25	0.81
1	5	4	0.5	0.12	0.06	0.19
Sum					0.31	1.00

Here, city 2 is chosen by Ant 1 as the next city to visit from city 5

三. 应用算例

3. First Iteration

- **Ant 1: How to construct a tour (Forward Mode)**
  - **Step 5: select the next city to visit from city 2;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
1	2	4	0.5	0.20	0.1	1.00
Sum					0.1	1.00

Here, city 4 is chosen by Ant 1 as the next city to visit from city 2

Therefore, Ant 1 can construct a tour: 1-3-5-2-4-1.
The distance value of this tour is: 19

三. 应用算例

3. First Iteration

- **Ant 2：How to construct a tour (Forward Mode)**
  - **Step 1: start from city 3 randomly;**
  - **Step 2: select the next city to visit from city 3;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij}\cdot\eta_{ij}$	p_{ijk}
2	3	1	0.5	0.33	0.165	0.17
2	3	2	0.5	0.25	0.125	0.13
2	3	4	0.5	0.33	0.165	0.17
2	3	5	0.5	1.00	0.5	0.53
Sum					0.955	1.00

Here, city 5 is chosen by Ant 2 as the next city to visit from city 3

三. 应用算例

3. First Iteration

- **Ant 2：How to construct a tour (Forward Mode)**
  - **Step 3: select the next city to visit from city 5;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
2	5	1	0.5	0.14	0.07	0.18
2	5	2	0.5	0.50	0.25	0.66
2	5	4	0.5	0.12	0.06	0.16
Sum					0.38	1.00

Here, city 2 is chosen by Ant 2 as the next city to visit from city 5

三. 应用算例

3. First Iteration

- **Ant 2： How to construct a tour (Forward Mode)**
  - **Step 4: select the next city to visit from city 2;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
2	2	1	0.5	0.16	0.08	0.44
2	2	4	0.5	0.20	0.1	0.56
Sum					0.18	1.00

Here, city 1 is chosen by Ant 2 as the next city to visit from city 2

三. 应用算例

3. First Iteration

- **Ant 2：How to construct a tour (Forward Mode)**
  - **Step 5: select the next city to visit from city 2;**

k	i	j	τ_{ij}	η_{ij}	$\tau_{ij} \cdot \eta_{ij}$	p_{ijk}
2	1	4	0.5	0.12	0.06	1.00
Sum					0.06	1.00

Here, city 4 is chosen by Ant 2 as the next city to visit from city 1

Therefore, Ant 2 can construct a tour: 3-5-2-1-4-3.
The distance value of this tour is: 20

三. 应用算例

3. First Iteration

- **Update pheromone matrix (Backward Mode)**
  - **Step 1: Evaporation radio is 0.5**

Pheromone Matrix after evaporation

ID	1	2	3	4	5
1	-	0.25	0.25	0.25	0.25
2	0.25	-	0.25	0.25	0.25
3	0.25	0.25	-	0.25	0.25
4	0.25	0.25	0.25	-	0.25
5	0.25	0.25	0.25	0.25	-

三. 应用算例

3. First Iteration

- **Update pheromone matrix (Backward Mode)**
  - **Step 2: Update by ant 1 with $\Delta\tau = 0.053$**

Pheromone Matrix after ant 1

ID	**1**	**2**	**3**	**4**	**5**
1	-	0.25	0. 303	0.303	0.25
2	0.25	-	0.25	0.303	0.303
3	0. 303	0.25	-	0.25	0.303
4	0.303	0.303	0.25	-	0.25
5	0.25	0.303	0.303	0.25	-

Ant 1 constructs a tour: 1-3-5-2-4-1, and the distance value of this tour is: 19.

三. 应用算例

3. First Iteration

- **Update pheromone matrix (Backward Mode)**
  - **Step 3: Update by ant 2 with $\Delta\tau = 0.05$**

Pheromone Matrix after ant 2

ID	1	2	3	4	5
1	-	0.30	0. 303	0.353	0.25
2	0.30	-	0.25	0.303	0.353
3	0. 303	0.25	-	0.30	0.353
4	0.353	0.303	0.30	-	0.25
5	0.25	0.353	0.353	0.25	-

Ant 2 constructs a tour: 3-5-2-1-4-3, and the distance value of this tour is: 20.

三. 应用算例

3. First Iteration

- **Update pheromone matrix (Backward Mode)**

Final Pheromone Matrix

ID	1	2	3	4	5
1	-	0.30	0. 303	0.353	0.25
2	0.30	-	0.25	0.303	0.353
3	0. 303	0.25	-	0.30	0.353
4	0.353	0.303	0.30	-	0.25
5	0.25	0.353	0.353	0.25	-

四. 算法变型

1. ACO Variations

- So far we described the basic version of ACO known as the Ant System (AS)
- Several variations were proposed that mainly differ on the way pheromone trails are updated
- Their objective is to address stagnation behavior
  - When all ants follow the same path early

四. 算法变型

2. Elitist Ant System (EAS)

- **Same construction rule with AS**
- **Additional pheromone deposit to the best ant**

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ijk} + e\,\Delta\tau_{ij}^{bs}, \forall(i,j) \in A$$

  - **m, A and $\Delta\tau_{ijk}$ defined as in AS**
  - **$\Delta\tau_{ij}^{bs}$ is the amount of pheromone the best ant deposit**
  - **e defines the weight given to the best ant**
- **Finds better solution in a fewer iterations than AS**

四. 算法变型

3. Rank-based Ant System (ASrank)

- **Same construction rule with AS**
- **Each ant deposits an amount of pheromone that decreases with its rank**

$$\tau_{ij} = \tau_{ij} + \sum_{r=1}^{w-1}(w-r)\Delta\tau_{ijk} + w\,\Delta\tau_{ij}^{bs}, \forall(i,j) \in A$$

  - A **and** $\Delta\tau_{ij}^{bs}$ **defined as in EAS**
  - r **defines the rank of the first** w **best ants**
  - w **defines the number of ants allowed to deposit pheromone**
- **The best ant always has the strongest weight, i.e.,** w
- **Performs significantly better than AS and slightly better than EAS**

四. 算法变型

4. MAX-MIN Ant System (MMAS)

- Same construction rule with AS
- Only the best ant is allowed to deposit pheromone

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}^{best}, \forall(i,j) \in A$$

  - $\Delta\tau_{ij}^{best}$ is the amount the best ant deposits
- Lower and upper limits on the possible pheromone values are imposed to avoid stagnation
- When a new best solution is found the limits are updated
- Pheromone trails are occasionally re-initialized
- Triggered when stagnation behavior occurs

四. 算法变型

5. Ant Colony System (ACS)

- Different construction rule than the other variations
- A new parameter q0 is introduced that allows to tune the exploration and exploitation
- With probability 1-q0 the same probabilistic rule with AS is used
- With probability q0 the next city with maximum probability is selected
- q0 is set to 0.9 hence the 2nd option is frequently used

四. 算法变型

5. Ant Colony System (ACS)

- **Only the best ant is allowed to deposit pheromone**
- **Pheromone evaporation is applied only to the arcs of the best ants**

$$\boldsymbol{\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{best}, \forall(i,j) \in T^{best}}$$

- **ρ and $\Delta\tau_{ij}^{best}$ are defined as above**
- **T^{best} is the tour of the best ant**

四. 算法变型

5. Ant Colony System (ACS)

- **Each ant performs a local pheromone update immediately after having crossed arc (i,j)**

$$\tau_{ij} = (1 - \xi)\tau_{ij} + \xi\tau_0$$

  - **τ_0 is the initial pheromone trail value**
  - **ξ is a parameter similar with the evaporation rate**

- **Note the all other variations described above perform a global pheromone update after all ants construct their solutions**

四. 算法变型

6. Population-based ACO (P-ACO)

- **Same construction rule with ACS**
- **Uses an explicit memory that stores the best ant**
- **"First in first out" policy is used to update memory**
- **The pheromone trails are updated according to the ants currently stored in the memory**

四. 算法变型

6. Population-based ACO (P-ACO)

- When an ant enters the memory, a constant amount of pheromone is added to the trails

$$\tau_{ij} = \tau_{ij} + \Delta_{ij,constrant}$$

- When an ant leaves the memory, a constant amount of pheromone is removed from the trails

$$\tau_{ij} = \tau_{ij} - \Delta_{ij,constrant}$$

四. 算法变型

6. Remarks on the Variations

- **ACS uses a more greedy probabilistic rule**
- **Local pheromone update is closer to the behavior of real ants that deposit pheromone as they are moving**
- **ACS and MMAS are the state-of-the-art ACO algorithms**
- **P-ACO is competitive with MMAS**
- **P-ACO does not use any pheromone evaporation; trails are removed directly**
- **MMAS explicitly imposes pheromone trail limits whereas ACS and P-ACO implicitly impose pheromone trail limits**

五. 算法应用实例

ACO求解旅行商问题（**TSP**）：假设有一个旅行商人要拜访全国**31**个省会城市，他需要选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择要求是：所选路径的路程为所有路径之中的最小值。

- 全国**31**个省会城市的坐标数据如下：

C = [1304 2312; 3639 1315; 4177 2244; 3712 1399; 3488 1535; 3326 1556; 3238 1229; 4196 1044; 4312 790; 4386 570; 3007 1970; 2562 1756; 2788 1491; 2381 1676; 1332 695; 3715 1678; 3918 2179; 4061 2370; 3780 2212; 3676 2578; 4029 2838; 4263 2931; 3429 1908; 3507 2376; 3394 2643; 3439 3201; 2935 3240; 3140 3550; 2545 2357; 2778 2826; 2370 2975]

五. 算法应用实例

算法仿真过程：

（1）初始化蚂蚁个数$\boldsymbol{m}=\mathbf{50}$，信息素重要程度参数**Alpha= 1**，启发式因子重要程度参数**Beta= 5**，信息素蒸发系数**Rho= 0.1**，最大迭代次数$\boldsymbol{G}=\mathbf{200}$，信息素增加强度系数$\boldsymbol{Q}=\mathbf{100}$。

（2）将$\boldsymbol{m}$个蚂蚁置在$\boldsymbol{n}$个城市上，计算待选城市的概率分布，$\boldsymbol{m}$个蚂蚁按概率函数选择下一座城市，完成各自的周游。

（3）记录本次迭代最佳路线，更新信息素，禁忌表清零。

（4）判断是否满足终止条件：若满足，则结束搜索过程，输出优化值；若不满足，则继续进行迭代优化。

五. 算法应用实例

作业要求：

- 参考代码，调试程序；
- 分析不同算法参数（种群大小、迭代次数、信息素重要程度参数、启发式因子重要程度参数、信息素蒸发系数、信息素增加强度系数）对算法性能的影响；
- 尝试其他**ACO**模型，设计并检验算法求解效果（选作）。