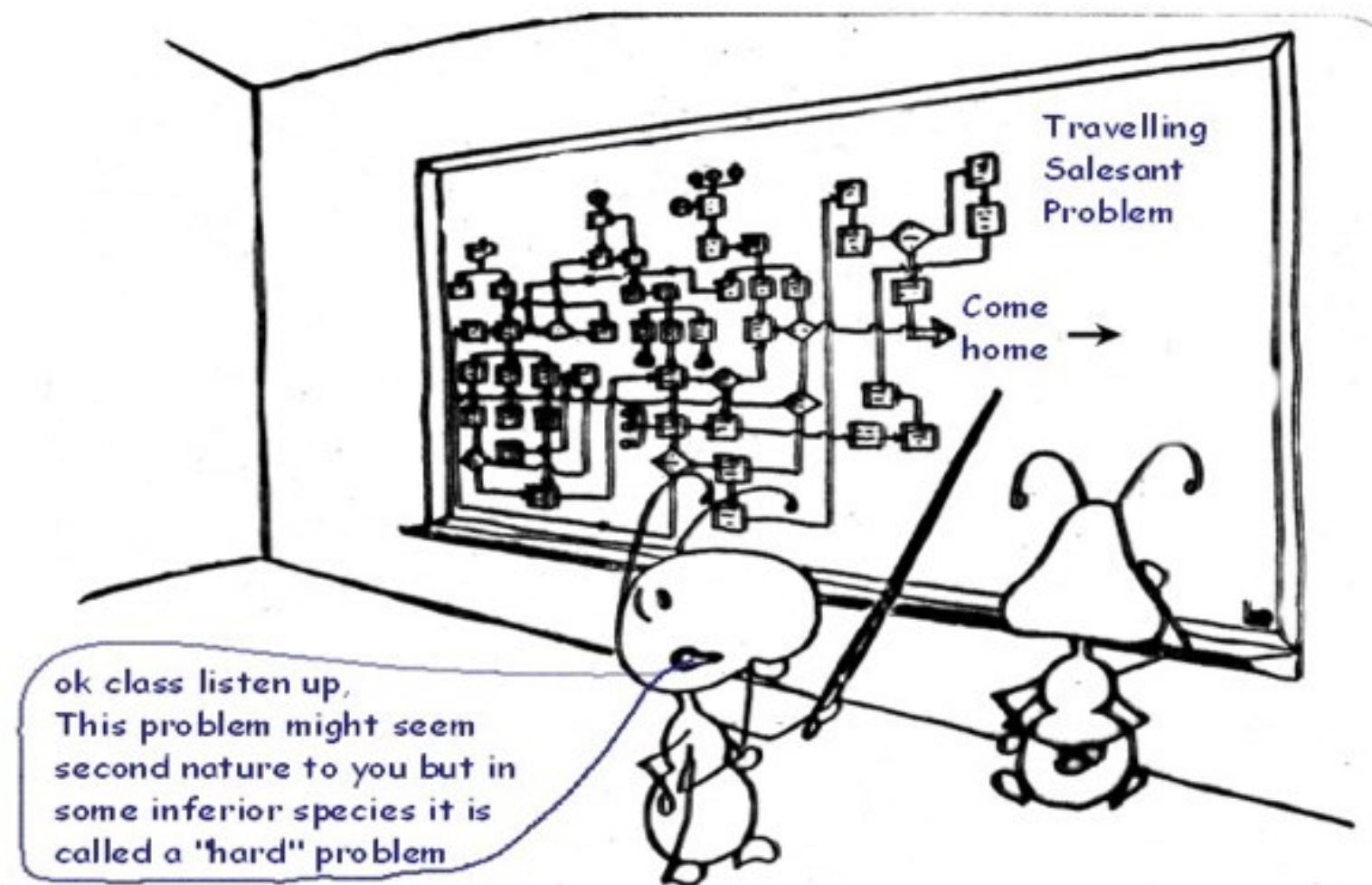


自然中的算法

蚁群优化 1



群体智能

- 智能体的集体系统，能够在动态多变的环境中完成困难的任务，而几乎不需要集中协调。
- 完成单个智能体单独行动通常无法完成的任务。
- 求解优化问题的通用框架。

鸟群

- 鸟为何成群？觅食、避敌、减小阻力（空气动力学）
- 简单规则产生有序行为。
- 算法视角参见
[Chazelle 2009]



鱼群

- 鱼为何成群？觅食、避敌等。
- 速度与准确性随群体规模变化，见 [Ward 2011]。

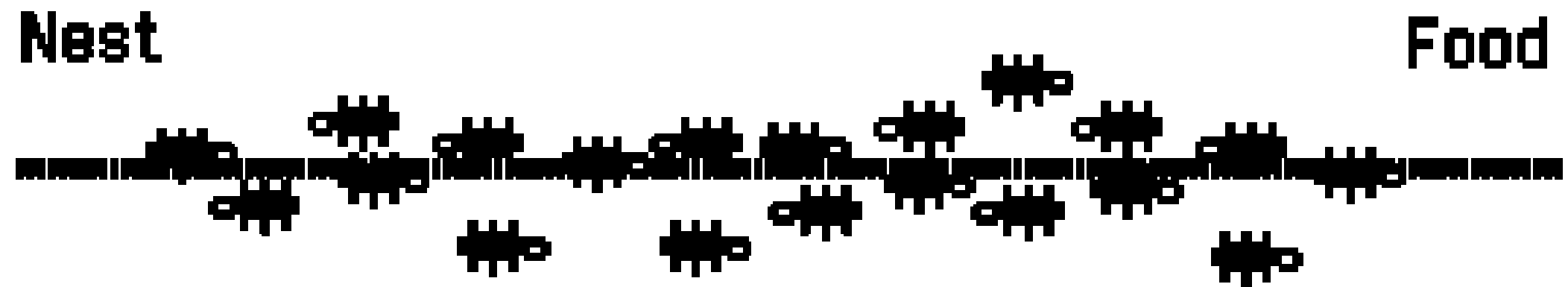


蚁群

- 单个蚂蚁几乎全盲，无法独自完成任务。
- 基本问题：
 - 找到食物。
 - 找到之后，招募其他蚂蚁。
 - 高效地把食物运回巢穴，供养整个群体。
- 难在哪里：
 - 食物可能在任何地方。
 - 通信能力有限。

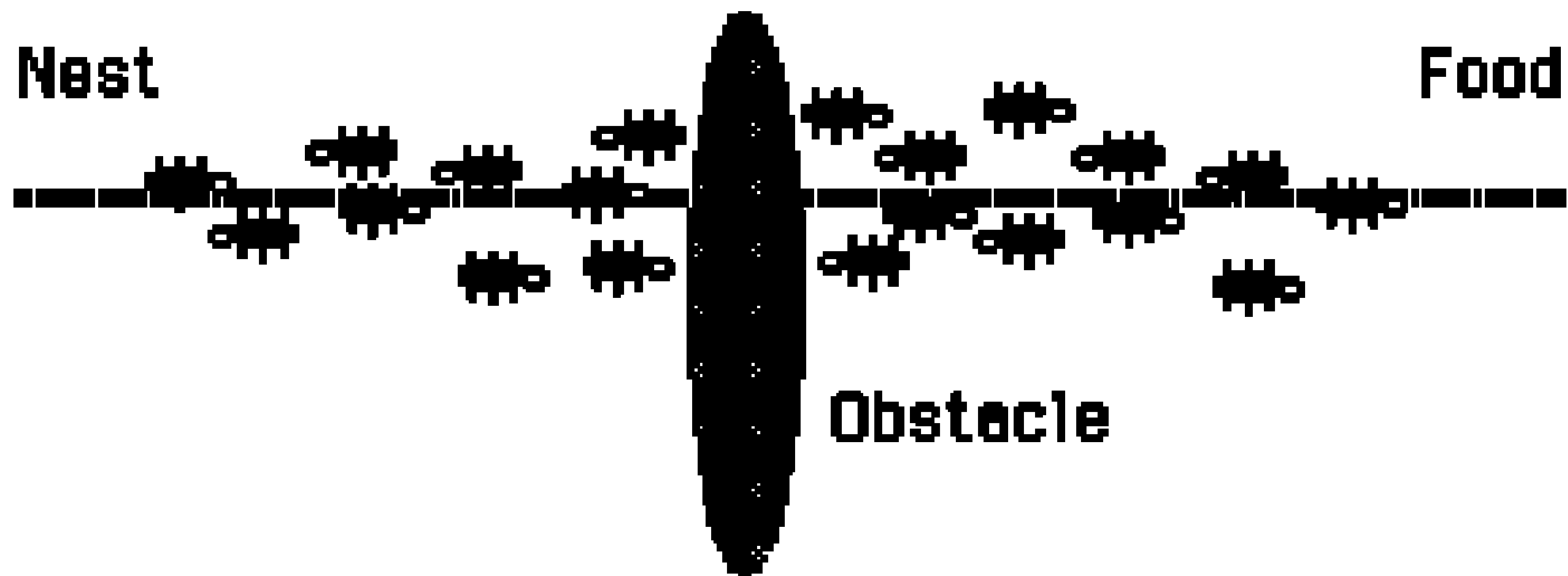


自然观察到的蚂蚁 行为



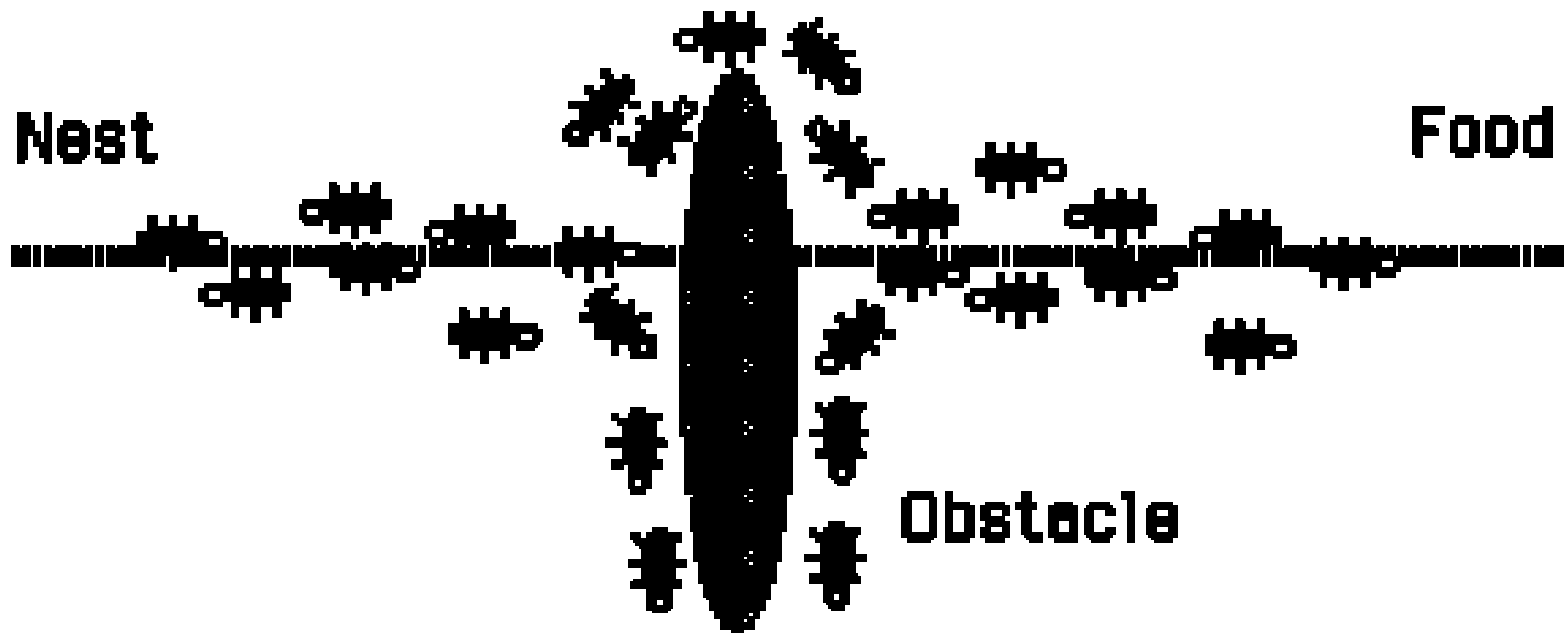
在蚂蚁的世界里，一切安好.....

自然观察到的蚂蚁 行为



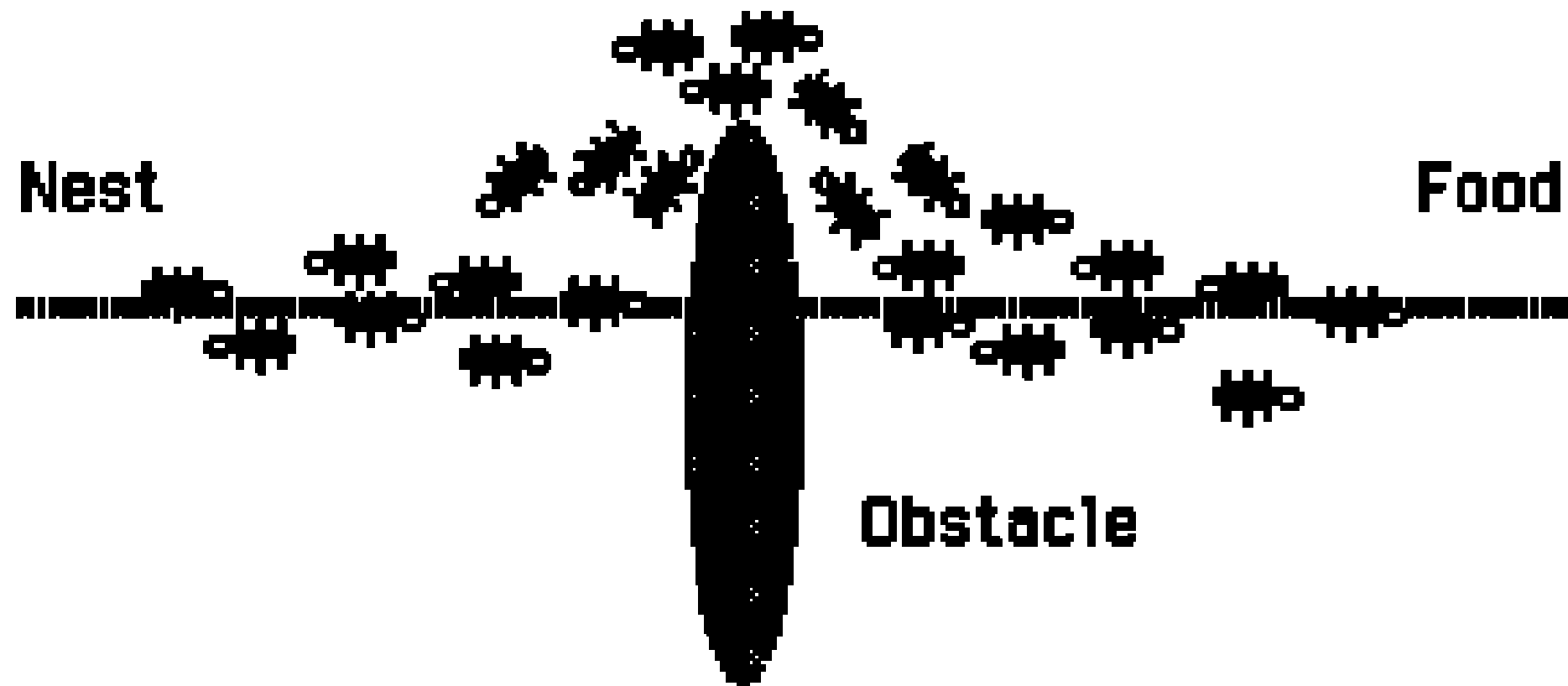
但现在，一个障碍物挡住了
去路

自然观察到的蚂蚁 行为



我们该怎么办？每只蚂蚁都掷一次
硬币。

自然观察到的蚂蚁 行为



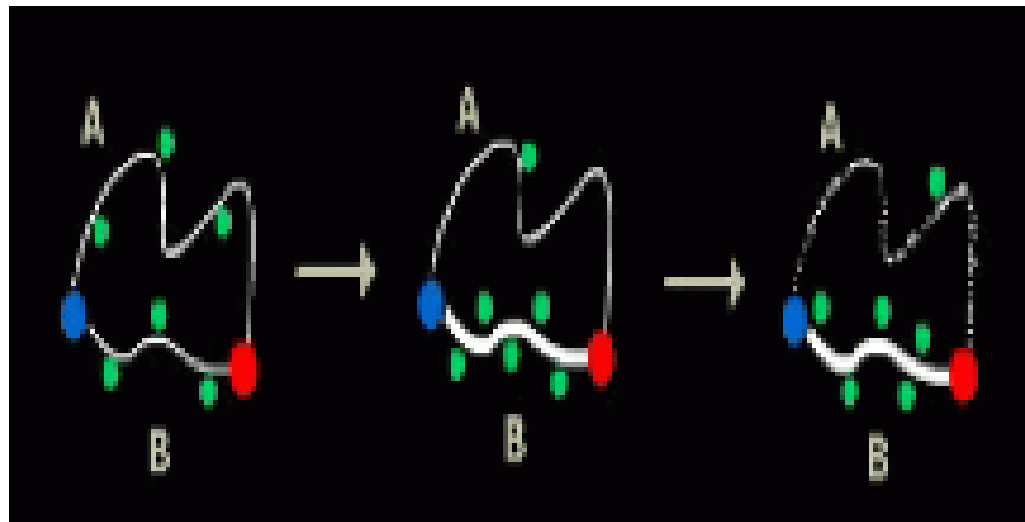
较短的路径被强化。这是如何发生的？

蚂蚁的通信

- 共识主动性（stigmergy）：由法国生物学家 Pierre-Paul Grasse 提出的术语，意为通过环境进行交互。
- 智能体之间的一种间接协调方式
- 当一个个体改变了环境，而另一个个体在稍后对这个新环境作出反应时，两者之间就发生了间接交互。
- 真实蚂蚁通过化学信息素实现共识主动性



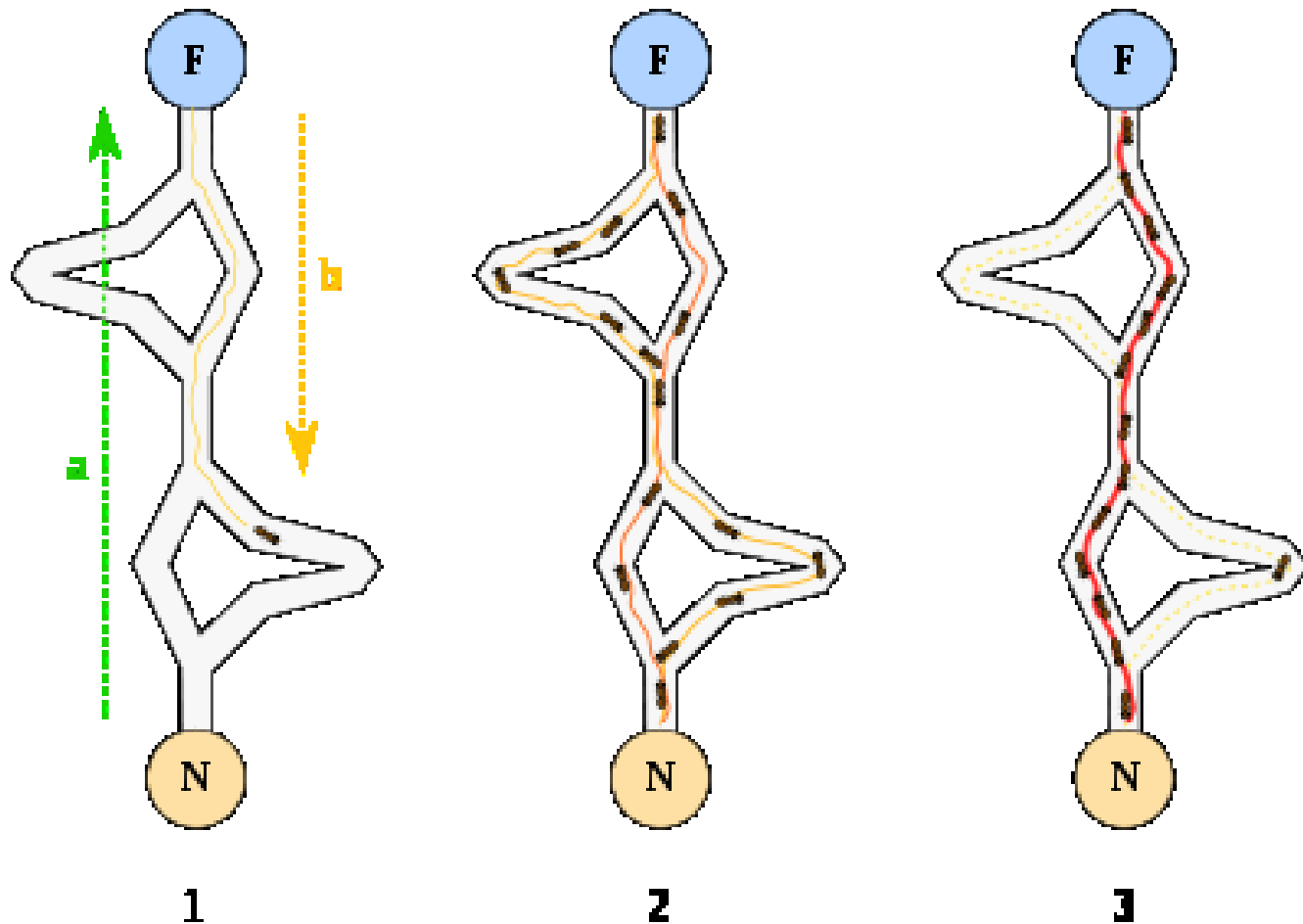
Natural ants: How do they do it?



- Since the route B is shorter, the ants on this path will complete the travel more times and **thereby** lay more pheromone **over** it.
- The pheromone concentration on trail B will increase at a higher rate than on A, and soon the ants on route A will choose to follow route B
- Since most ants will no longer travel on route A, and since the pheromone is volatile, trail A will start evaporating
- **Only** the shortest route will remain!



自然蚂蚁：它们是如何做到的？



1. 第一只蚂蚁随机游走，直到找到食物源 (F)，然后返回巢穴 (N)，沿途留下信息素。

2. 其他蚂蚁随机地沿 4 条路径中的一条前进，同时留下信息素

3. 短路径会比长路径获得更多的信息素强化，从而使其对后来的蚂蚁更有吸引力。

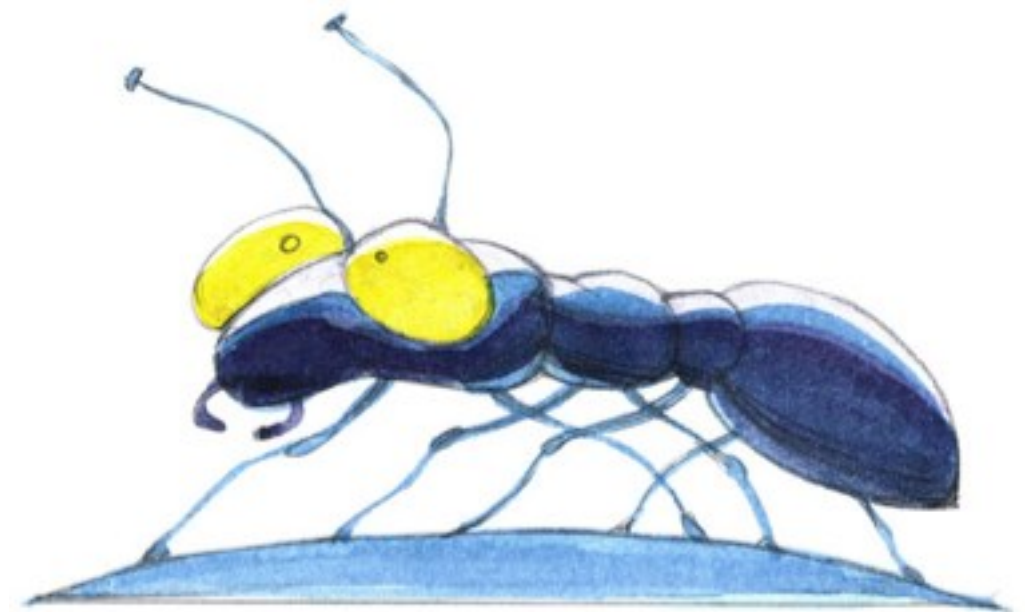
4. 蚂蚁越来越有可能走最短路径，因为按定义它就被强化最多的路径。

5. 长路径上的信息素会挥发。

自催化

即正反馈

- 以较高的概率沿已有的信息素轨迹前进。跟随某条轨迹的蚂蚁越多，该轨迹就越有吸引力。
- 因此这一过程具有正反馈回路的特征：某条离散路径此前被选择的次数越多，再次选择它的概率就越大。
- 总体而言，用计算机科学的术语说，该过程是并行与分布式的、随机的，并基于反馈自适应。



蚁群优化

- 由 Marco Dorigo 在其 1992 年的博士论文中提出
- 求解优化问题的通用框架，用于在图中
寻找路径，其灵感来自
蚁群的行为。
- 最初用于旅行商问题，后来用于
各种其他的 NP 难优化问题。

旅行商

问题

- 一个经典的 NP 难组合优化问题
- 给定 n 个城市以及它们之间两两的距离：访问每个城市恰好一次的最短路线是什么？



假定距离为
欧氏距离。

Performance Chart

总距离

迭代次数

占最优解的比例

Problem Name	MST	AS	ACS _{R&D}	ACS _D	GA	EP	SA	Optimum
Eil50 (50-city problem)	615 [1] 44.71%	450 [36] 5.89%	463.423 [3] 9.04%	425 [1,830] 0.00%	428 [25,000] 0.71%	426 [100,000] 0.23%	443 [68,512] 4.24%	425 [N/A]
Eil75 (75-city problem)	740 [1] 38.31%	570 [238] 6.5%	576.749 [10] 7.80%	545 [3,840] 1.87%	545 [80,000] 1.87%	542 [325,000] 1.31%	580 [173,250] 8.41%	535 [N/A]
KroA100 (100-city problem)	30517 [1] 43.39%	22,943 [228] 7.81%	24497.6 [37] 15.11%	21,282 [4,820] 0.00%	21,761 [103,000] 2.25%	N/A [N/A] N/A	N/A [N/A] N/A	21,282 [N/A]

Our Results

MST 2-approximation TSP algorithm

AS Ant System

($\alpha = 1$, $\beta = 5$, $\rho = .5$)

ACS_{R&D} Ant Colony System

($\alpha = 0.1$, $\beta = 2$, $\rho = .1$, $m = 50$)

Published Results

ACS_D Ant Colony System

GA Genetic Algorithm

EP Evolutionary Programming

SA Simulated Annealing

作为计算主体的 蚂蚁

- 蚂蚁从某个城市出发。
- 每只蚂蚁选择下一个城市的概率是城市间距离与该边上已有信息素量的函数。
- 为使蚂蚁走出合法的回路，在完成一次回路之前禁止转移到已访问的城市（由禁忌表控制）。
- 完成回路后，蚂蚁会在在经过的每条边留下信息素

自然	CS
自然栖息地	图（节点与边）
巢穴与食物	图中的节点
蚂蚁	分布式智能体
方向似然	边似然： $1/d_{ij}$
觅食行为	随机信息素 图中引导式游走

面向 TSP 的蚁群 优化

图 (V, E) : 其中 V = 城市 (节点) , E = 边

d_{ij} = 从城市 i 到城市 j 的欧氏距离

蚂蚁以某个转移概率从城市 i 移动到下一个城市 j 。

每条边 (r, s) 关联两个值:

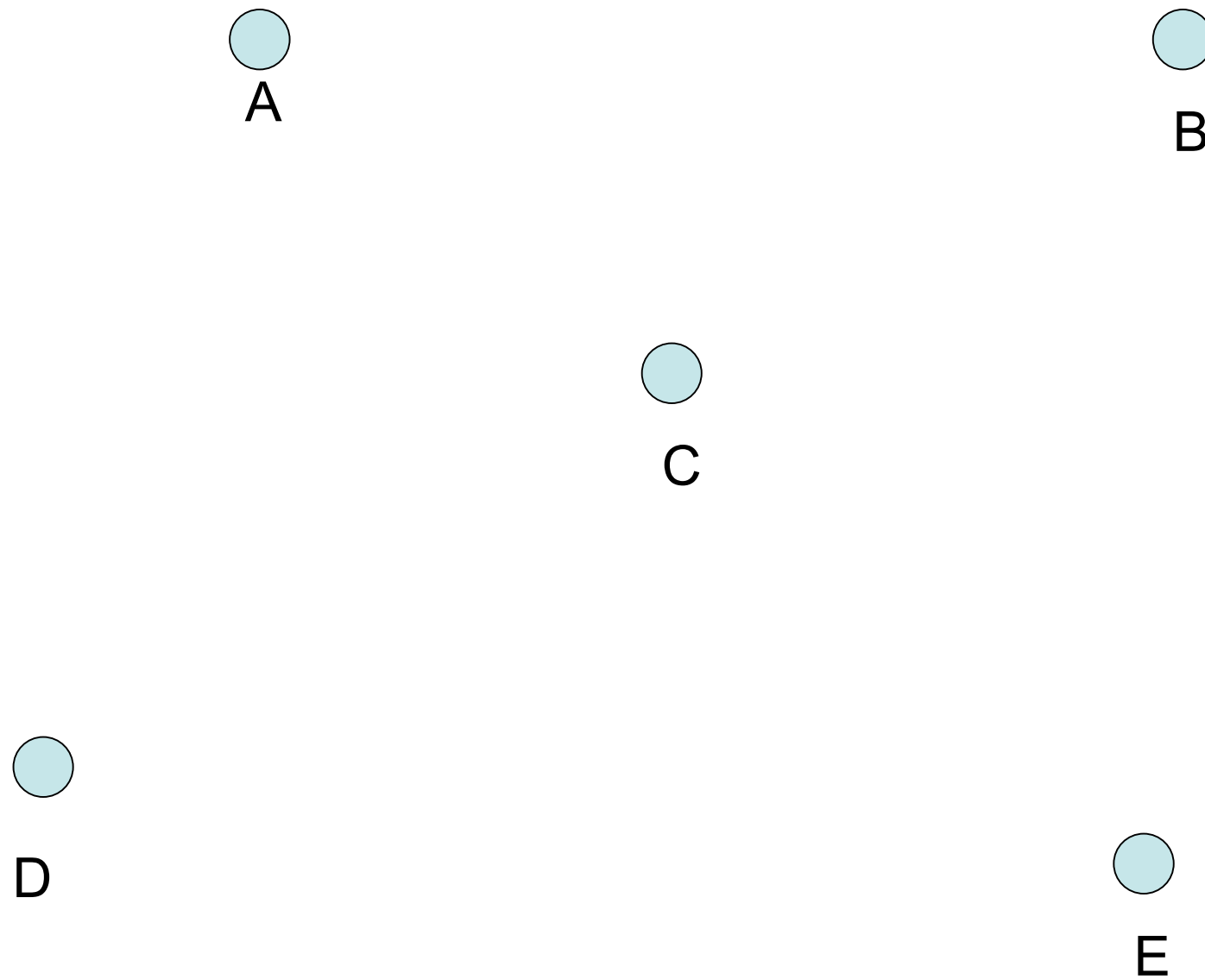
基于边代价的静态值 $\eta(r,s) = 1/d_{rs}$

以及基于所释放信息素量的动态值 $\tau(r,s)$

初始时 τ 为 0。

每只蚂蚁都试图生成一条完整回路,
依据 $\eta(r,s)$ 与 $\tau(r,s)$ 来选择下一个城市。

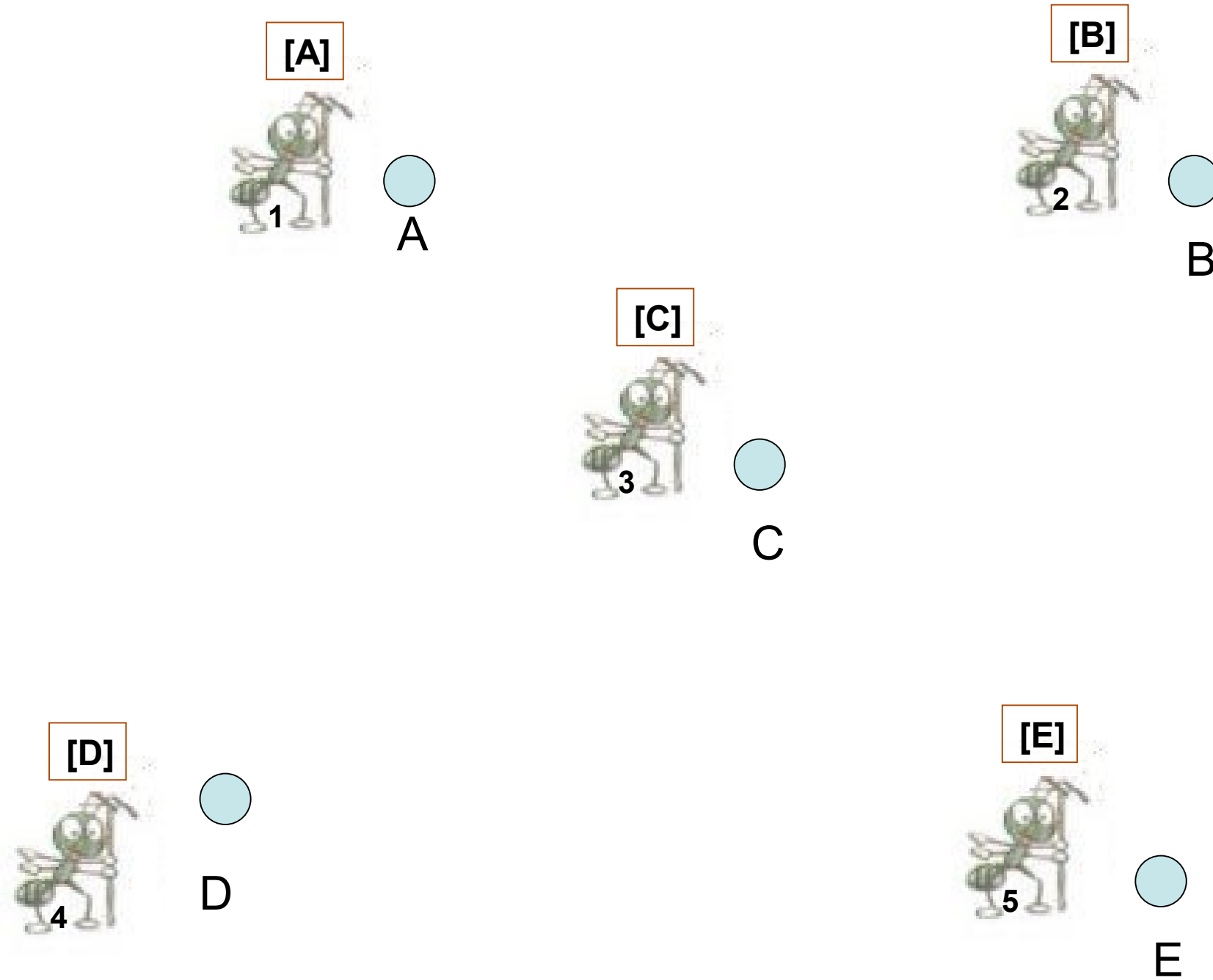
一个简单的 TSP 示例



$$d_{AB} = 100; d_{BC} = 60 \dots; d_{DE} = 150$$



第 1 次迭代



如何选择下一个城市?



$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases}$$

第 k 只蚂蚁走到的
从节点 i 到
节点 j , 在时刻 t

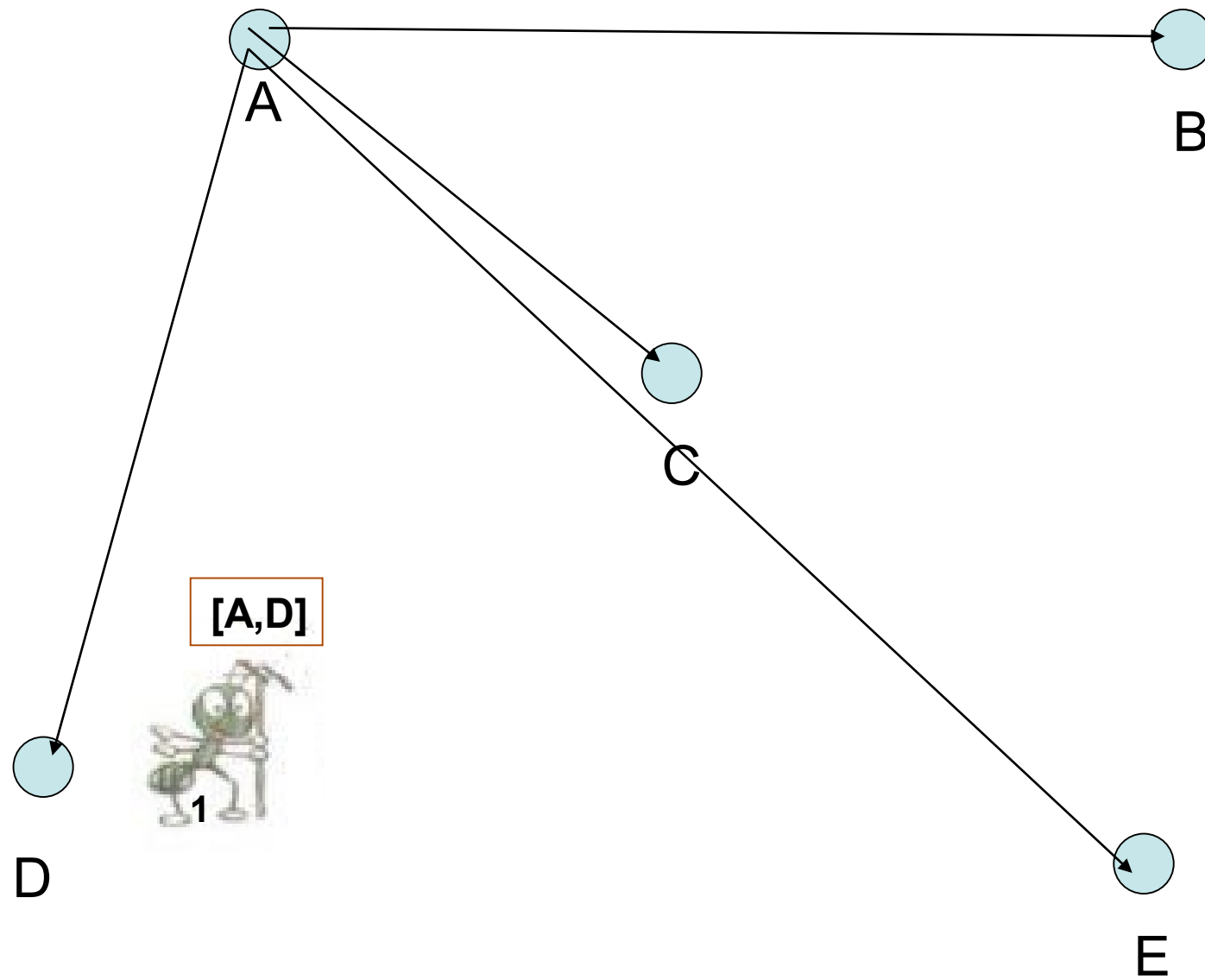
D

的禁忌表
允许的下一
城市

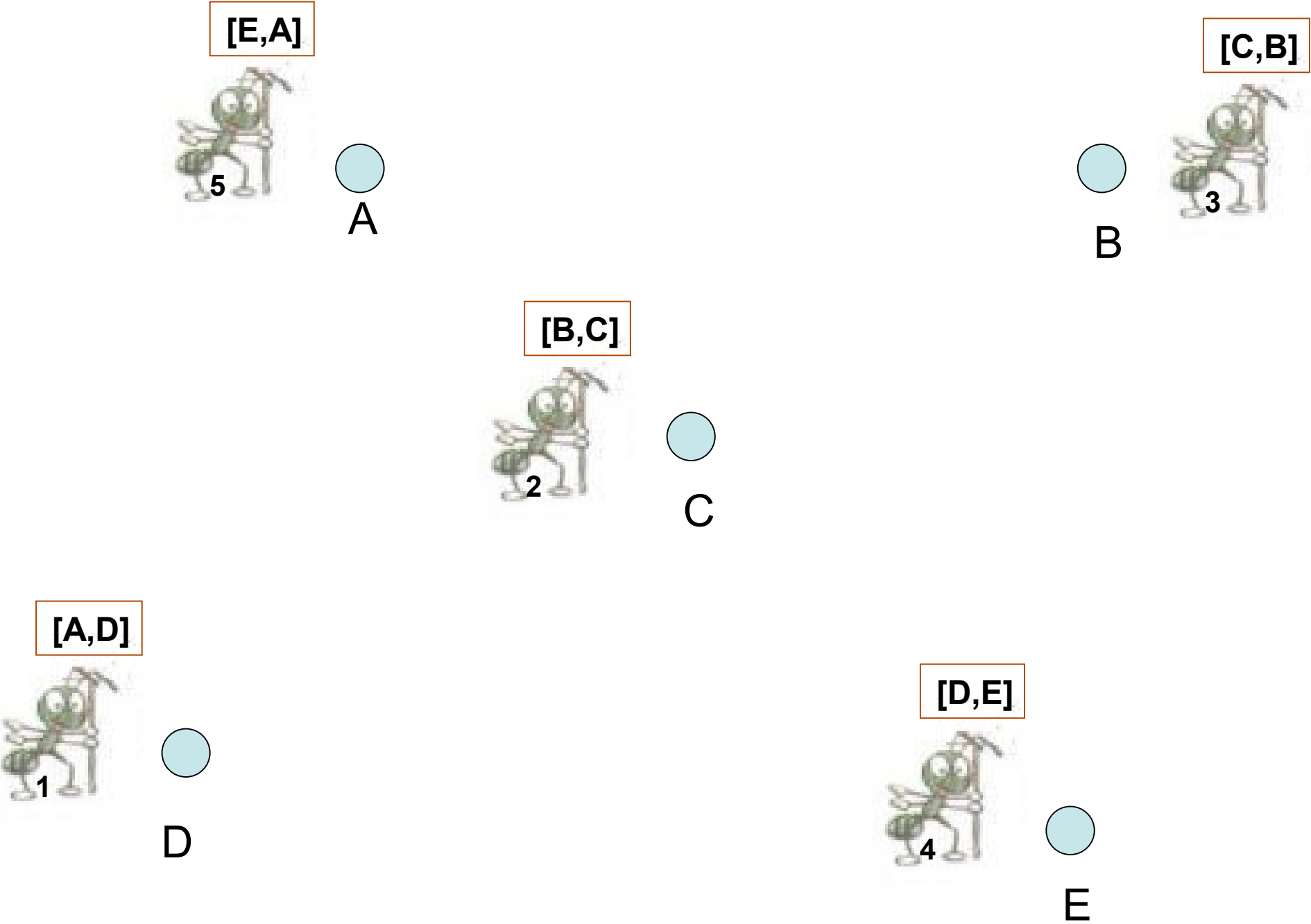
当前的信息素量
信息素, 从
节点 i 到节点 k

静态距离
从 i 到 k
E

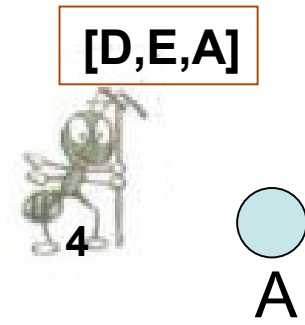
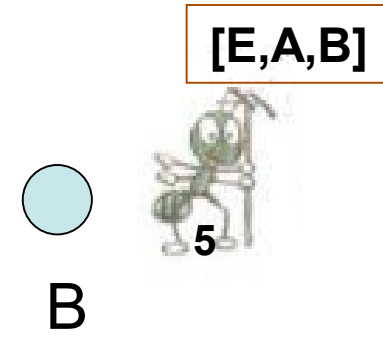
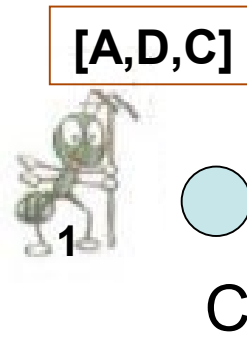
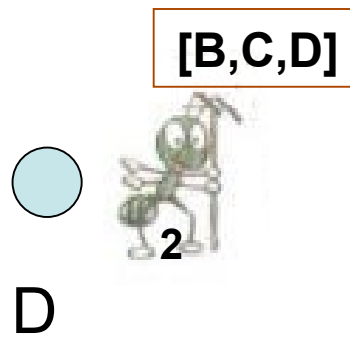
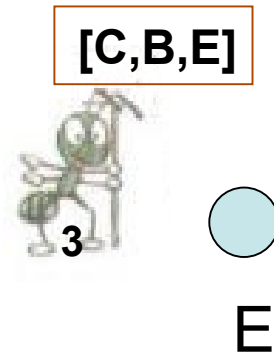
如何选择下一个城市?



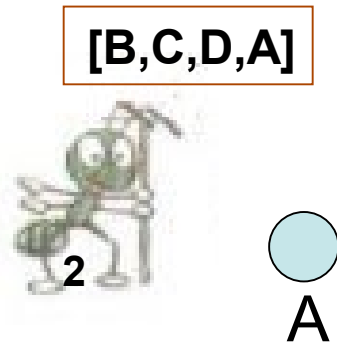
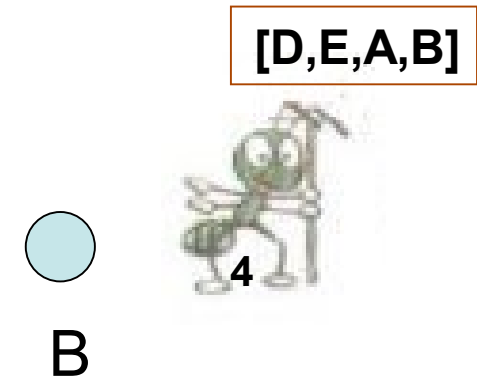
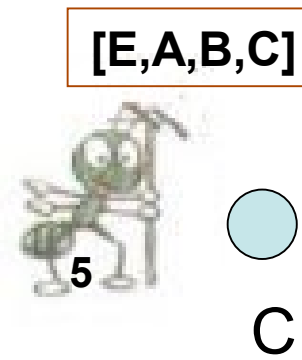
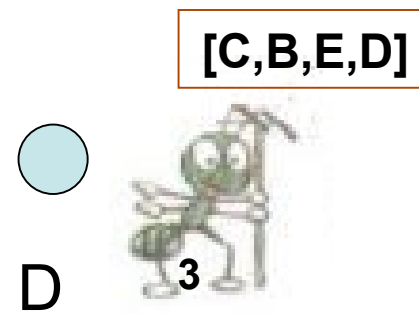
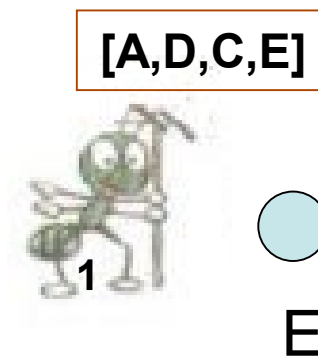
第 2 次迭代



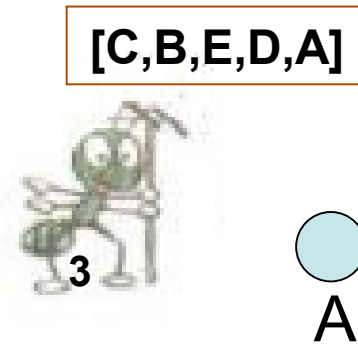
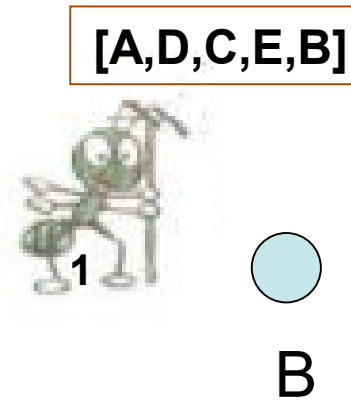
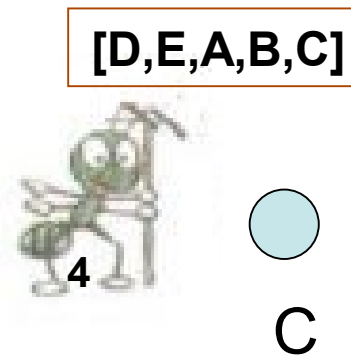
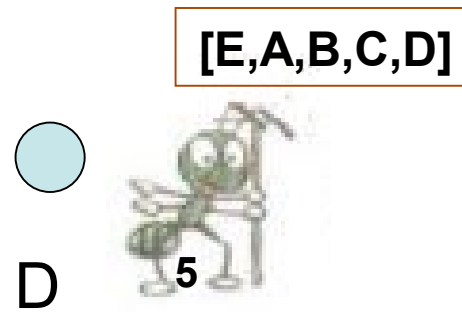
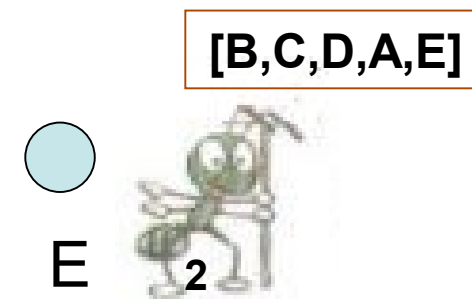
第 3 次迭代



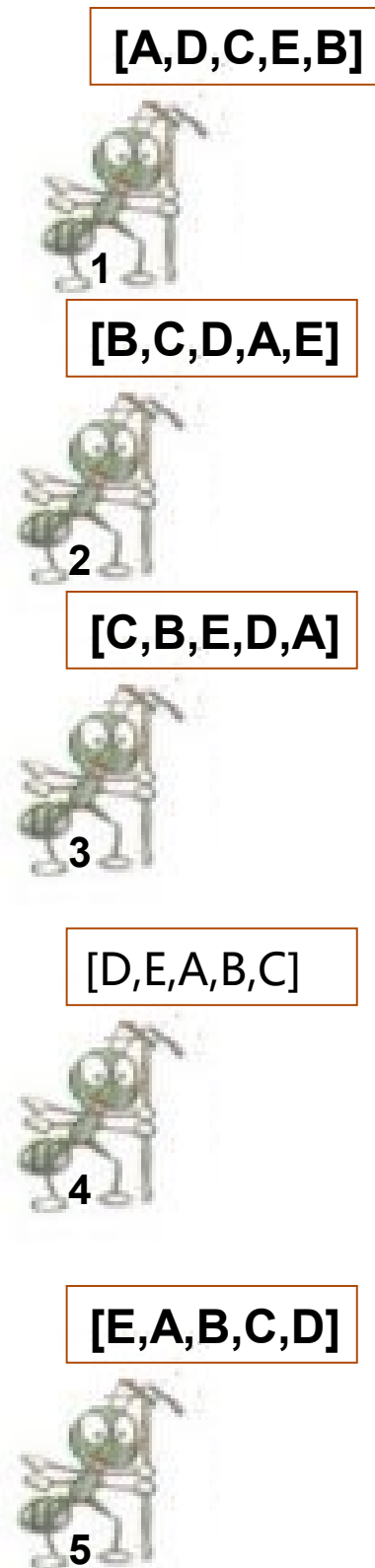
第 4 次迭代



第 5 次迭代



路径与轨迹更新



$L_1 = 300$

$L_2 = 450$

$L_3 = 260$

$L_4 = 280$

$L_5 = 420$

constant, typically = 1

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{Q}{L_k} & \text{if } (i,j) \in \text{bestTour} \\ 0 & \text{otherwise} \end{cases}$$

每只蚂蚁都会更新它所经过的所有边上的信息素；更新量为回路长度的函数（越短则越多）used)

那么信息素挥发呢？

常数，通常取 1

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{Q}{L_k} & \text{if } (i,j) \in \text{bestTour} \\ 0 & \text{otherwise} \end{cases}$$

先更新
所用的信息素

每只蚂蚁都会更新它经过的所有边上的信息素。
更新量取决于回路的长度
(回路越短，每条边信息素越多)

然后，挥发

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}$$

的量
信息素，沿
边，时刻 $t+1$

衰减率

所有蚂蚁加到该边上的量

信息素
沿边，在
时刻 t

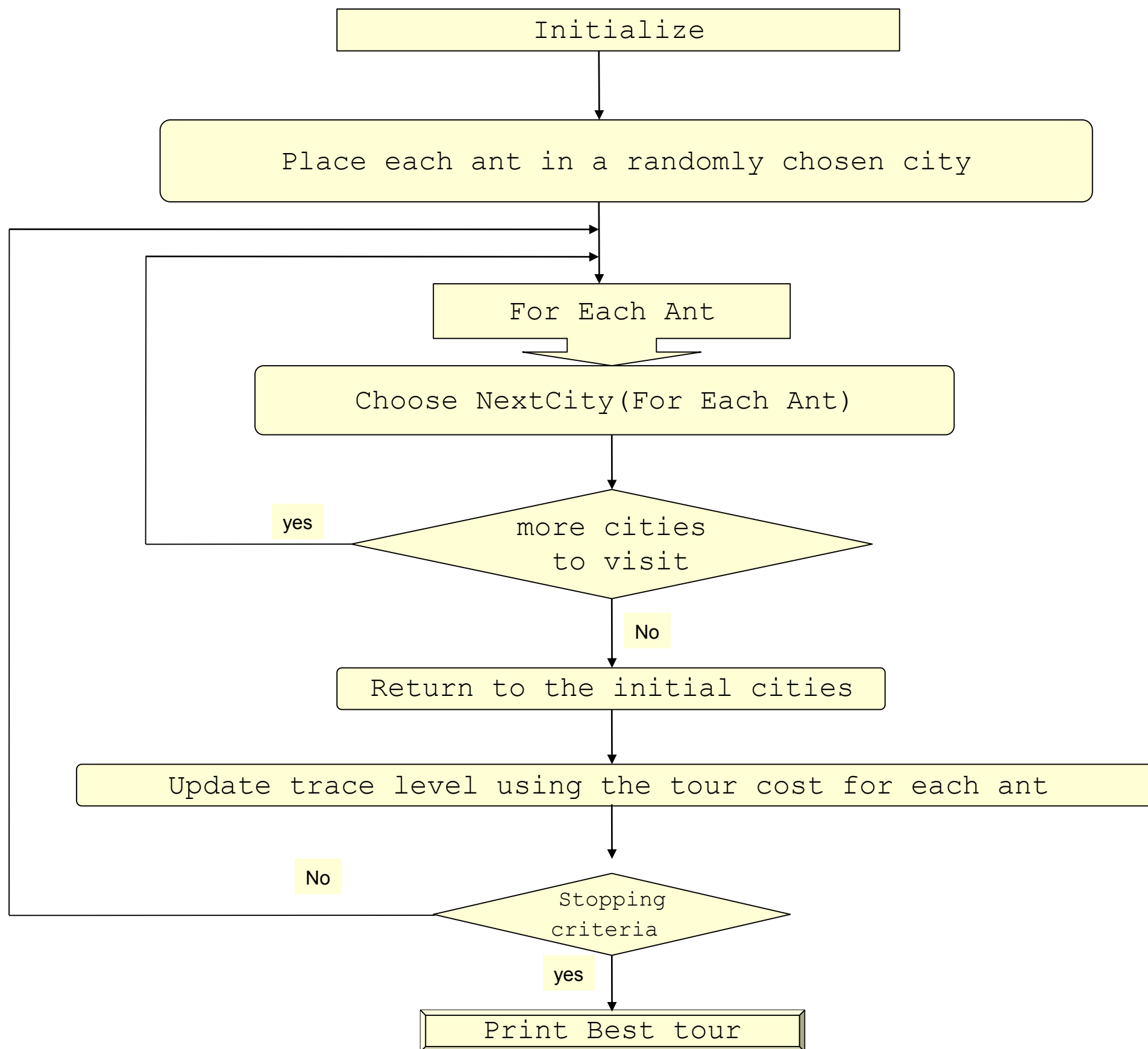
新增的信息素

第一轮结束

保存最优回路（序列与长度）

所有蚂蚁重新出发并重复

面向 TSP 的 ACO 算法



参数及其影响

1. α (信息素)

取值越大, 信息素越重要

2. β (边距离)

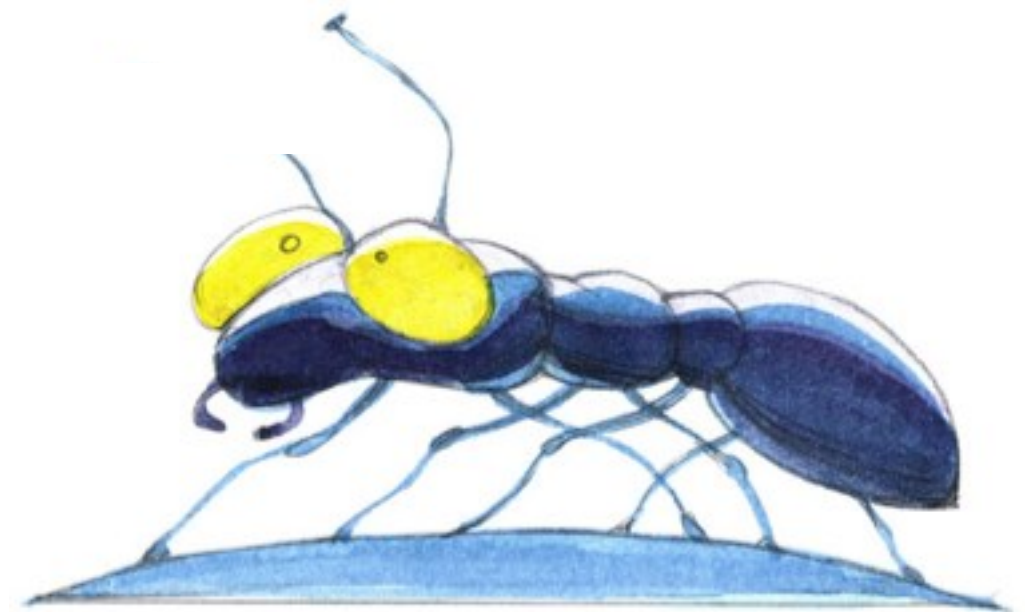
取值越大, 静态距离越重要

取值小意味着随机性更强 $\rightarrow$ 探索与利用策略

3. p (挥发率)

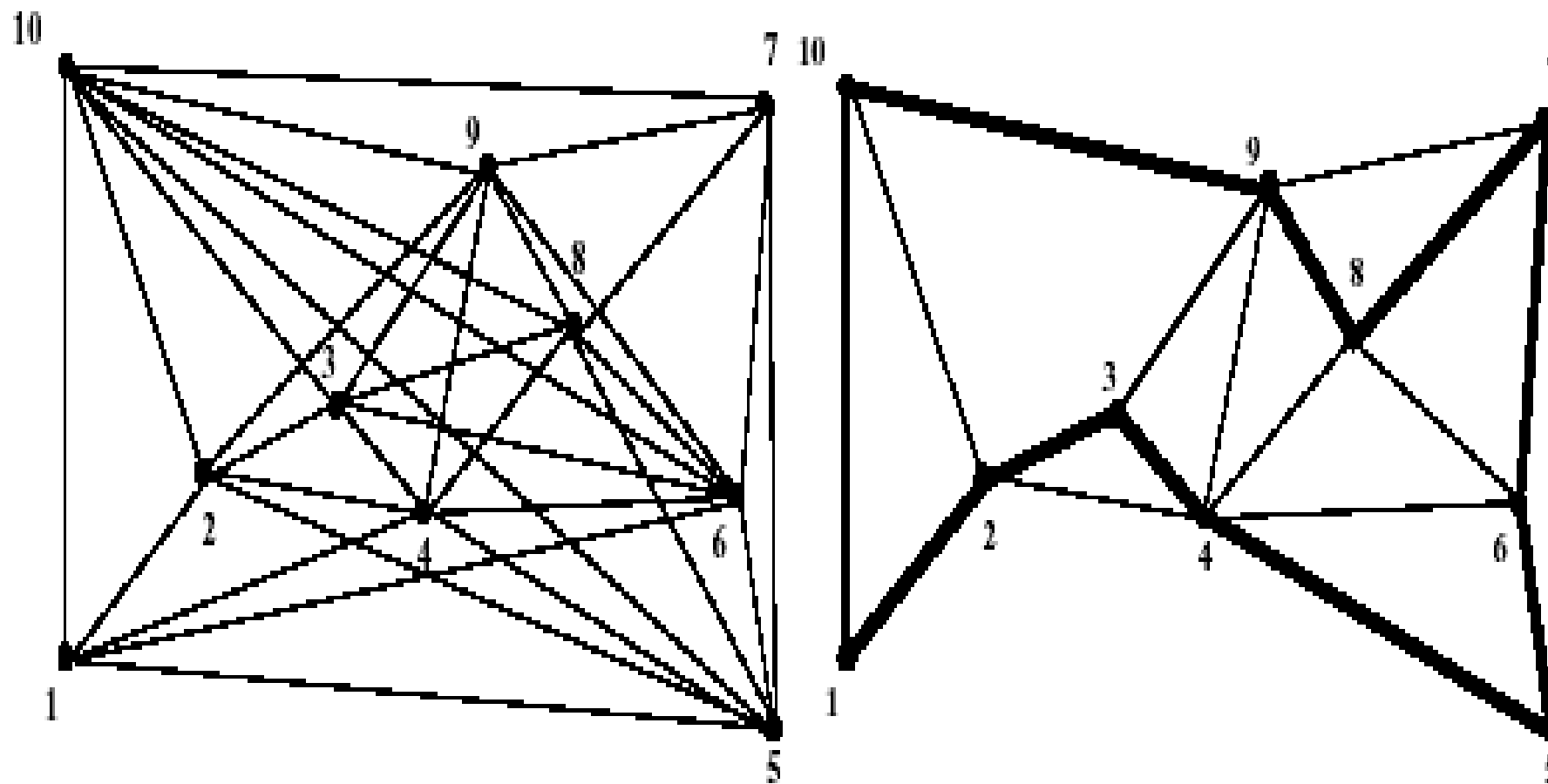
取值大意味着适应快, 但可能“过拟合”;

取值小则适应慢



停止准则

- 停滞
- 最大迭代次数

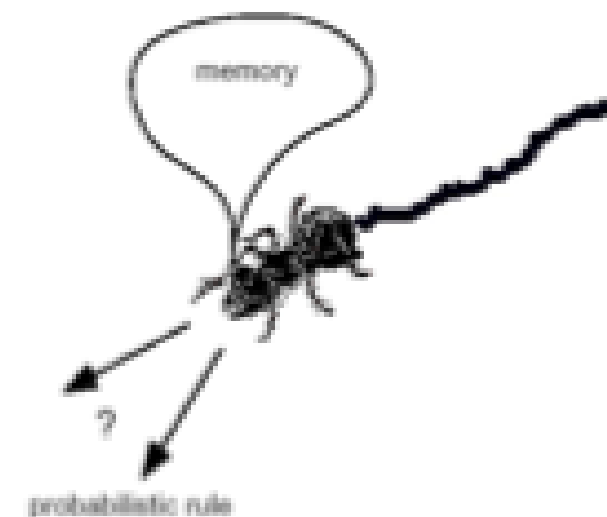


Ant System Algorithm for TSP



Pseudocode:

```
initialize all edges to (small) initial pheromone level  $\tau_0$ ;  
place each ant on a randomly chosen city;  
for each iteration do:  
  do while each ant has not completed its tour:  
    for each ant do:  
      move ant to next city by the probability function  
    end;  
  end;  
  for each ant with a complete tour do:  
    evaporate pheromones;  
    apply pheromone update;  
    if (ant k's tour is shorter than the global solution)  
      update global solution to ant k's tour  
    end;  
  end;  
end;
```



- 优点：
 - 分布式算法：蚂蚁并发且独立地工作
 - 反馈使好的解能够被快速发现
 - 概率性的（鲁棒）
- 缺点：
 - 解的质量没有保证；理论分析困难
 - 对全局解的利用效率不高
- 蚁群启发式算法有大量变体与专门会议

ACO 在其他问题上的应用

Problem name	Authors	Algorithm name	Year
Traveling salesman	Dorigo, Maniezzo & Colomi	AS	1991
	Gamberdella & Dorigo	Ant-Q	1995
	Dorigo & Gamberdella	ACS & ACS 3 opt	1996
	Stutzle & Hoos	MMAS	1997
	Bullnheimer, Hartl & Strauss	AS _{rank}	1997
	Cordon, et al.	BWAS	2000
Quadratic assignment	Maniezzo, Colomi & Dorigo	AS-QAP	1994
	Gamberdella, Taillard & Dorigo	HAS-QAP	1997
	Stutzle & Hoos	MMAS-QAP	1998
	Maniezzo	ANTS-QAP	1999
	Maniezzo & Colomi	AS-QAP	1994
Scheduling problems	Colomi, Dorigo & Maniezzo	AS-JSP	1997
	Stutzle	AS-SMTTP	1999
	Barker et al	ACS-SMTTP	1999
	den Besten, Stutzle & Dorigo	ACS-SMTWTP	2000
	Merkle, Middenderf & Schmeck	ACO-RCPS	1997
Vehicle routing	Bullnheimer, Hartl & Strauss	AS-VRP	1999
	Gamberdella, Taillard & Agazzi	HAS-VRP	1999

ACO 在其他问题上的应用

Problem name	Authors	Algorithm name	Year
Connection-oriented network routing	Schoonderwood et al.	ABC	1996
	White, Pagurek & Oppacher	ASGA	1998
	Di Caro & Dorigo	AntNet-FS	1998
	Bonsabeau et al.	ABC-smart ants	1998
Connection-less network routing	Di Caro & Dorigo	AntNet & AntNet-FA	1997
	Subramanian, Druschel & Chen	Regular ants	1997
	Heusse et al.	CAF	1998
	van der Put & Rethkrantz	ABC-backward	1998
Sequential ordering	Gamberdella & Dorigo	HAS-SOP	1997
Graph coloring	Costa & Hertz	ANTCOL	1997
Shortest common supersequence	Michel & Middendorf	AS_SCS	1998
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998
Generalized assignment	Ramalhinho Lourenco & Serra	MMAS-GAP	1998
Multiple knapsack	Leguizamón & Michalewicz	AS-MKP	1999
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999
Redundancy allocation	Liang & Smith	ACO-RAP	1999
Constraint satisfaction	Solnon	Ant-P-solver	2000

**遗憾的是，并非所有蚂蚁都用信息素来
指导行为.....**

自然中的算法

蚁群优化 2

- 优点：
 - 分布式算法：蚂蚁并发且独立地工作
 - 反馈使好的解能够被快速发现
 - 概率性的（鲁棒）
- 缺点：
 - 解的质量没有保证；理论分析很困难
 - 对全局解的利用有些低效

ACO 在其他问题上的应用

Problem name	Authors	Algorithm name	Year
Traveling salesman	Dorigo, Maniezzo & Colomi	AS	1991
	Gamberdella & Dorigo	Ant-Q	1995
	Dorigo & Gamberdella	ACS & ACS 3 opt	1996
	Stutzle & Hoos	MMAS	1997
	Bullnheimer, Hartl & Strauss	AS _{rank}	1997
	Cordon, et al.	BWAS	2000
Quadratic assignment	Maniezzo, Colomi & Dorigo	AS-QAP	1994
	Gamberdella, Taillard & Dorigo	HAS-QAP	1997
	Stutzle & Hoos	MMAS-QAP	1998
	Maniezzo	ANTS-QAP	1999
	Maniezzo & Colomi	AS-QAP	1994
Scheduling problems	Colomi, Dorigo & Maniezzo	AS-JSP	1997
	Stutzle	AS-SMTTP	1999
	Barker et al	ACS-SMTTP	1999
	den Besten, Stutzle & Dorigo	ACS-SMTWTP	2000
	Merkle, Middenderf & Schmeck	ACO-RCPS	1997
Vehicle routing	Bullnheimer, Hartl & Strauss	AS-VRP	1999
	Gamberdella, Taillard & Agazzi	HAS-VRP	1999

ACO 在其他问题上的应用

Problem name	Authors	Algorithm name	Year
Connection-oriented network routing	Schoonderwood et al.	ABC	1996
	White, Pagurek & Oppacher	ASGA	1998
	Di Caro & Dorigo	AntNet-FS	1998
	Bonsabeau et al.	ABC-smart ants	1998
Connection-less network routing	Di Caro & Dorigo	AntNet & AntNet-FA	1997
	Subramanian, Druschel & Chen	Regular ants	1997
	Heusse et al.	CAF	1998
	van der Put & Rethkrantz	ABC-backward	1998
Sequential ordering	Gamberdella & Dorigo	HAS-SOP	1997
Graph coloring	Costa & Hertz	ANTCOL	1997
Shortest common supersequence	Michel & Middendorf	AS_SCS	1998
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998
Generalized assignment	Ramalhinho Lourenco & Serra	MMAS-GAP	1998
Multiple knapsack	Leguizamón & Michalewicz	AS-MKP	1999
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999
Redundancy allocation	Liang & Smith	ACO-RAP	1999
Constraint satisfaction	Solnon	Ant-P-solver	2000

匿名文件共享 (MUTE)

<http://mute-net.sourceforge.net/>

- 问题：在用户之间实现 P2P 文件共享，同时保护用户隐私
- 通常，消息在互联网上发送时，消息的源 IP 与目的 IP 都是可见的。
- 这里要确保没有任何节点知道发送方与接收方使用的是哪台计算机。

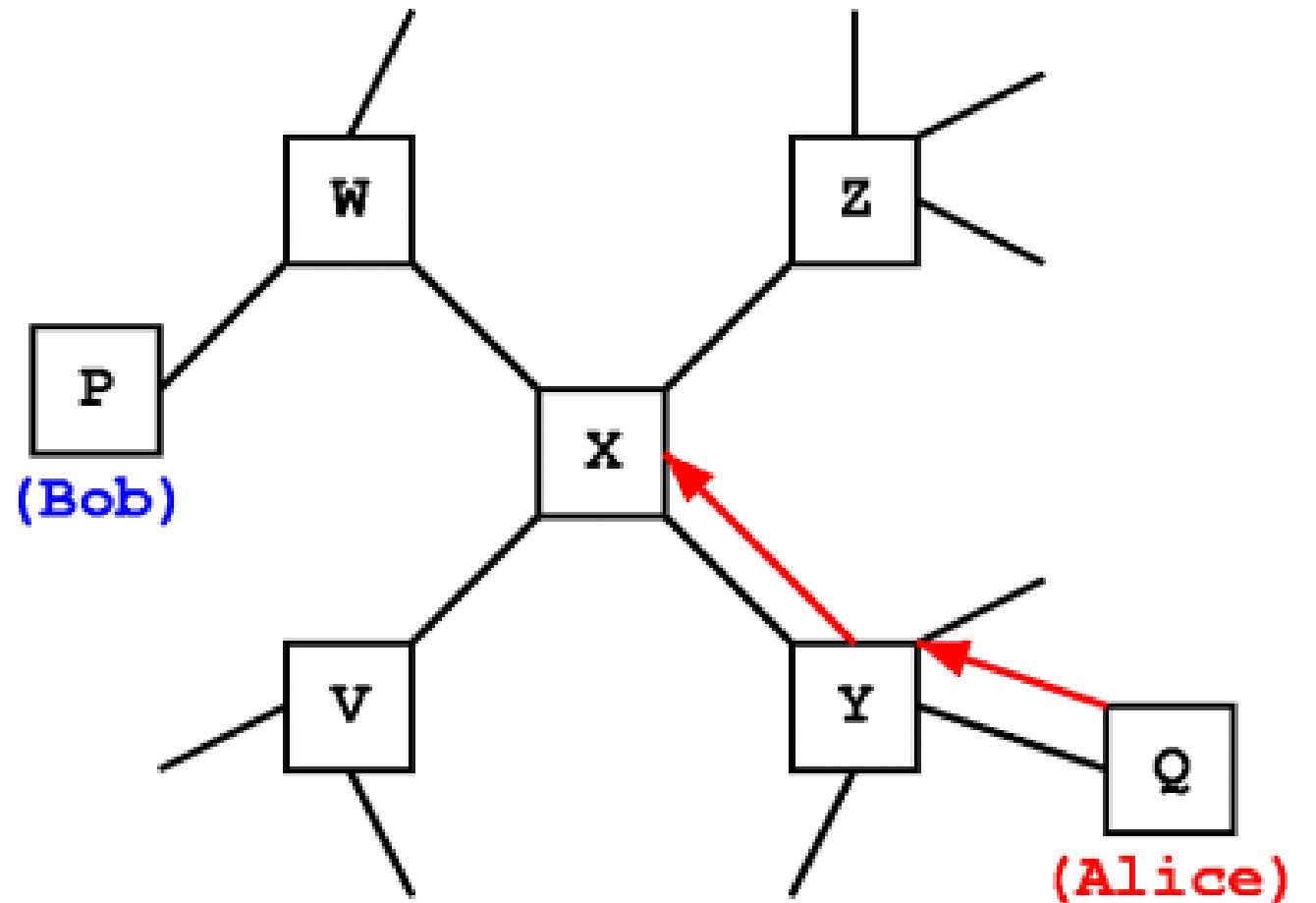
MUTE

假设 X 收到一条
从 Alice 发给 Bob 的消息，它经由
邻居节点 Y 转发。

X 可能并不知道 Bob 在
网络中的位置。

但是，在收到该
消息之后，X 就得到了关于
Alice 的线索：即来自
Alice 的消息是经由 Y 传来的。

今后，如果 X 需要发送
消息给 Alice，它就知道应当
经由 Y 发送。

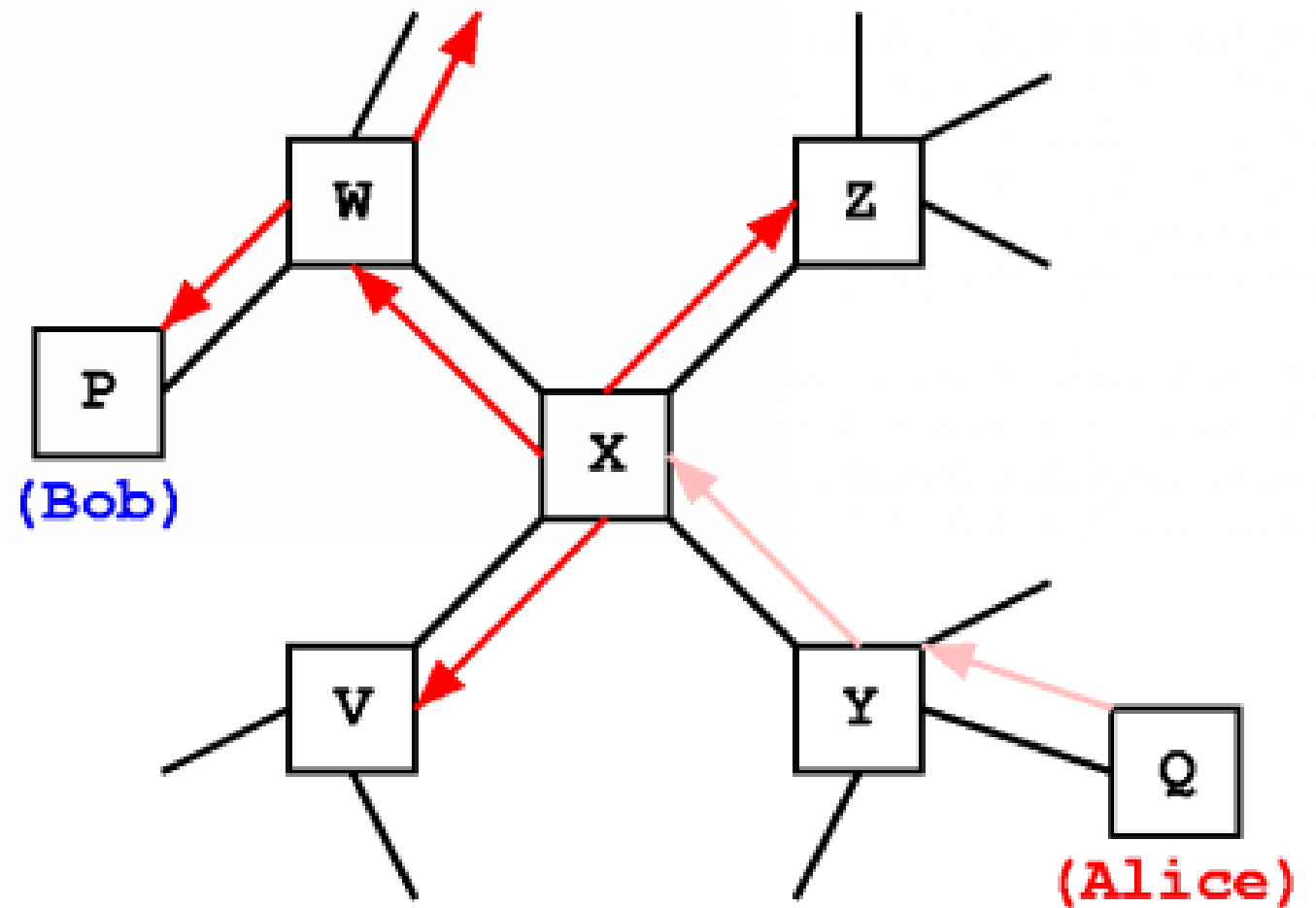


MUTE

X 对 Bob 一无所知，那么它会怎么做？它把这条消息广播给 X 的所有邻居。

其中某个邻居可能知道 Bob 所在方向的信息。（如果也不知道，它们同样广播）

如果 Bob 存在，消息最终会到达他那里。

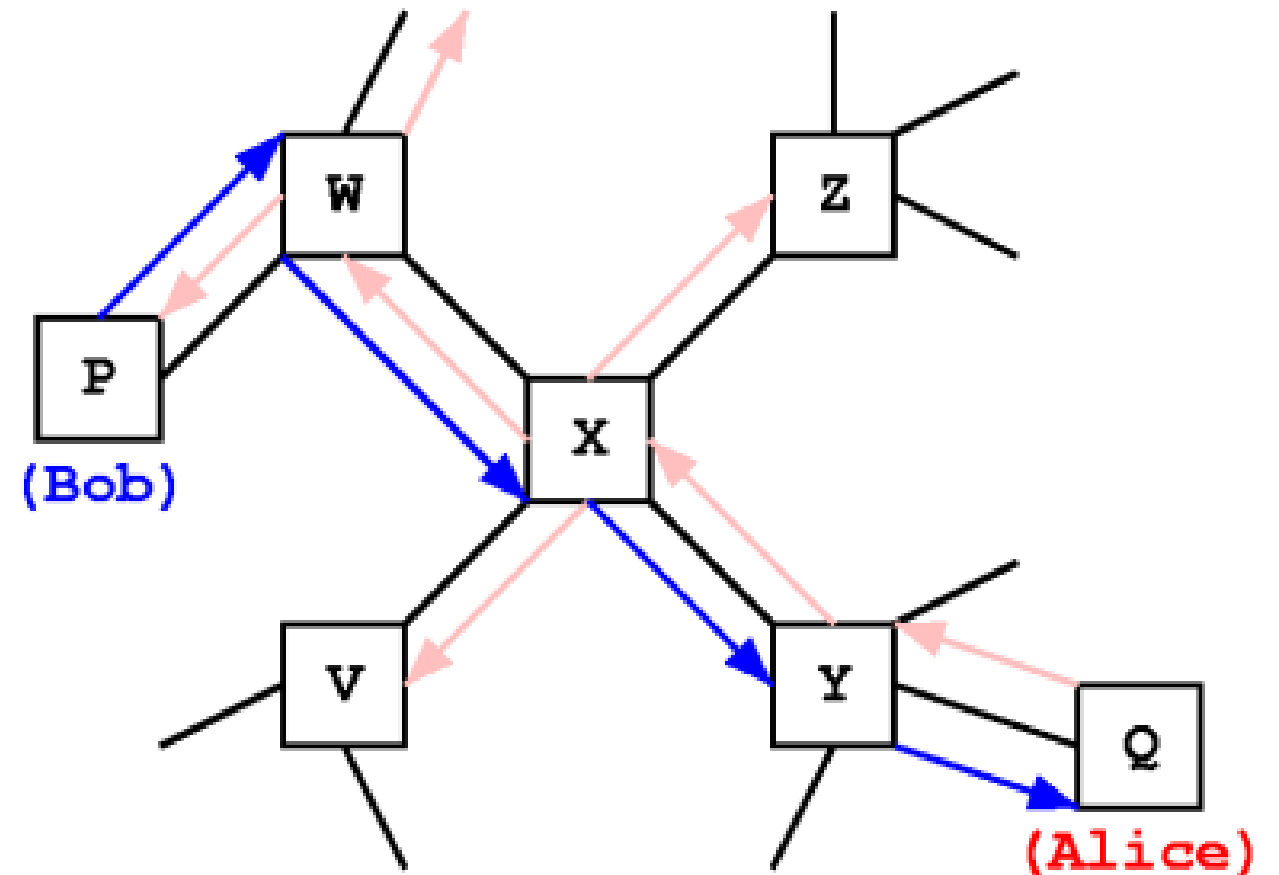


MUTE

在寻找 Bob 的过程中，
这条消息留下了一串关于
Alice 的线索。

如果消息到达了 Bob，并且
Bob 需要回复 Alice，那么
回复可以沿同一条路径
返回 Alice。

这又留下了关于如何
找到 Bob 的线索（而且始终不
泄露 Bob 所用计算机
的身份）。



任何节点都不能说：“我的邻居是 Alice”。
它只能说：“我的邻居比我知道
更多关于 Alice 的事”。

**遗憾的是，并非所有蚂蚁都用信息素来
指导行为.....**

The Regulation of Ant Colony Foraging Activity without Spatial Information

Balaji Prabhakar¹, Katherine N. Dektar², Deborah M. Gordon^{3*}

¹ Departments of Computer Science and Electrical Engineering, Stanford University, Stanford, California, United States of America, ² Biomedical Computation, School of Engineering, Stanford University, Stanford, California, United States of America, ³ Department of Biology, Stanford University, Stanford, California, United States of America

收获蚁

- 生活在沙漠中，风把种子随机地吹散。
- 蚂蚁代谢种子中的脂肪获取水分与养分
- 一只蚂蚁就能取回种子并带回群体——与之前不同，“招募”其他蚂蚁没有好处。



觅食的挑战

- 蚂蚁觅食时会失水，因此需要根据当前条件来调节觅食：
 - 天气
 - 一天中的时段
 - 当前的食物可得性
 - 与邻近群体可能的竞争
- 必须在没有集中控制的情况下决定何时出去觅食



觅食是如何发生的？

一只蚂蚁经由一条 5–10 厘米的通道回巢，把种子丢在通道里。种子会被另一只蚂蚁捡起并带入巢内。回巢的蚂蚁重新排队。

调节依赖于来自回巢觅食蚁的反馈，它们（通过触角接触）刺激更多的觅食蚁离开巢穴

觅食蚁的回巢率与食物可得性相对应：食物越多 → 回巢率越高 → 出巢率越高



本文：提出一个关于回巢率如何影响出巢率的模型。

方法

- 野外研究在美国新墨西哥州 Rodeo 进行：
 - 2009 年：8 天内、9 个群体，共 33 次试验
 - 2010 年：5 天内、8 个群体，共 29 次试验
- 对觅食速率进行操控：
 - 记录离巢觅食蚁的数量
 - 记录回巢觅食蚁的数量
 - 由一人移除成功的觅食蚁（携带种子的个体）
- 每次试验记录 20 分钟：
 - 0–240 秒：操控前
 - 240–430 秒：操控中
 - 500–1100 秒：操控后

视频录制与追踪, 使用 www.antrack.org

The logo for AnTracks features the word "AnTracks" in a white, sans-serif font. The letters "AnTrac" are enclosed within a green rectangular border. The letter "K" is stylized, with its left vertical stroke in red and its right vertical stroke in green. The letter "s" is white and follows the "K".

AnTracks

Demonstration Video:
Tracking Ants on the Soil

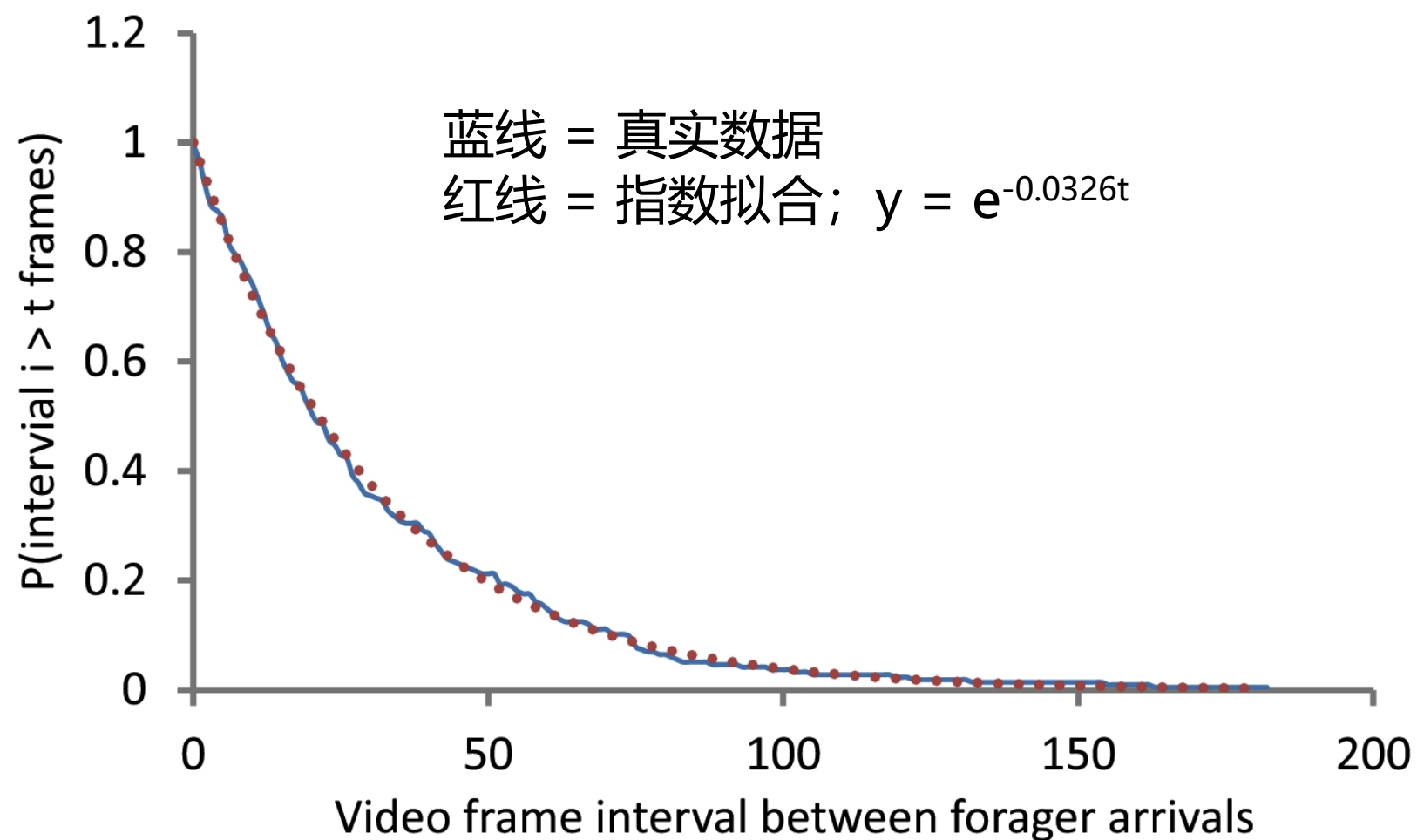
© Martin Stumpe 2010

<http://antracks.martinstumpe.com>

觅食蚁的回巢率服从泊松分布

计算两只相继回巢的觅食蚁之间的时间间隔。
单次试验，由操控前的时间窗计算得到。

两只相继回巢的
觅食蚁之间的间隔
大于 t 帧
的概率



间隔长度
(1 帧 = 1/30 秒)

觅食调节的第一个模型

- 线性模型——出巢觅食蚁的速率 $x(t)$ 取决于回巢觅食蚁的速率加上恒定的基础速率：

$$(1) \quad \lambda(t) = \lambda_{ac} f(t, \tau) \quad \text{[回巢觅食蚁的速率]}$$

$$(2) \quad x(t) = \text{Poisson}(\lambda_b) + \text{Poisson}(\lambda(t)) \quad \text{[出巢觅食蚁的速率]}$$

where λ_b is a baseline rate of outgoing foragers, independent of the rate at which other foragers return; λ_{ac} sets the number of outgoing foragers per returning forager; and $f(t, \tau)$ is the number of returning foragers between times $t-\tau$ and t .

- 这还不够，因为觅食速率并非在所有时间区间上都一致：会因天气、时段、群体等而不同。

改进的模型

- 在离散时间下运行:

$$(3) \quad \alpha_n = \max(\alpha_{n-1} - qD_{n-1} + cA_n - d, \alpha), \quad \alpha_0 = 0$$

出巢觅食蚁在时刻 n 的速率

α_{n-1} 先前的速率

D_{n-1} D_n 是在时刻 n 离巢的觅食蚁数量;
 α 减小, 因为队列中的觅食蚁变少;
 q 是一个参数

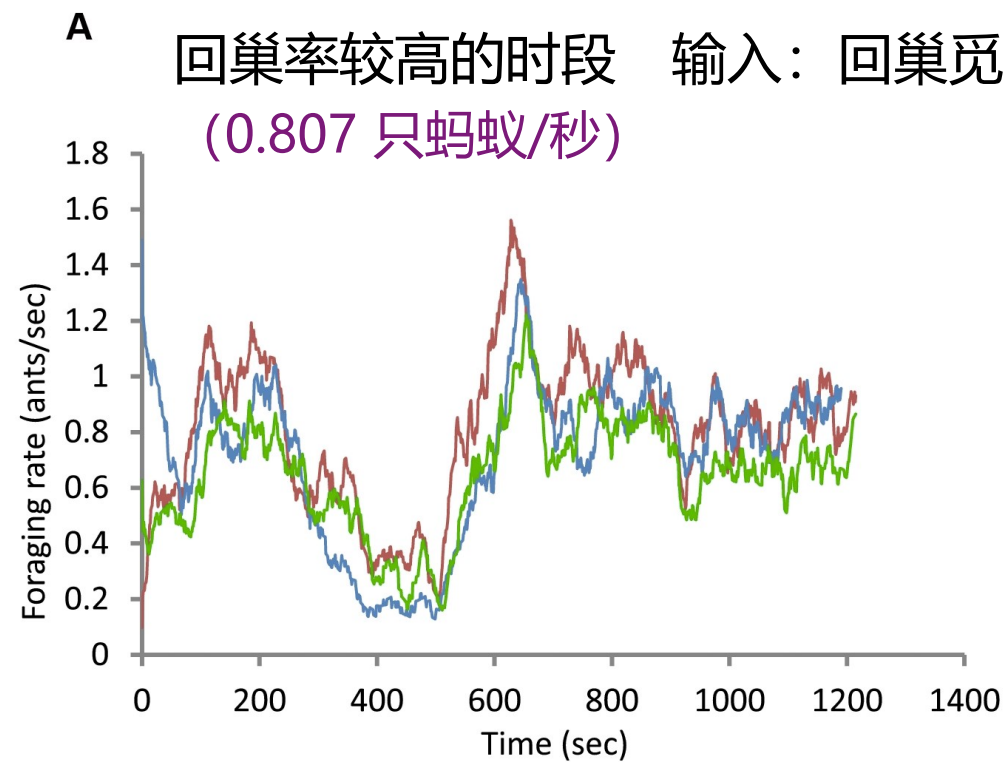
A_n 是携带食物回巢的觅食蚁数量;
 α 增大, 因为队列中的蚂蚁更多; c 是一个参数

常数衰减

基准值:
有些觅食蚁也会离巢, 即使一段时间内没有觅食蚁回巢。

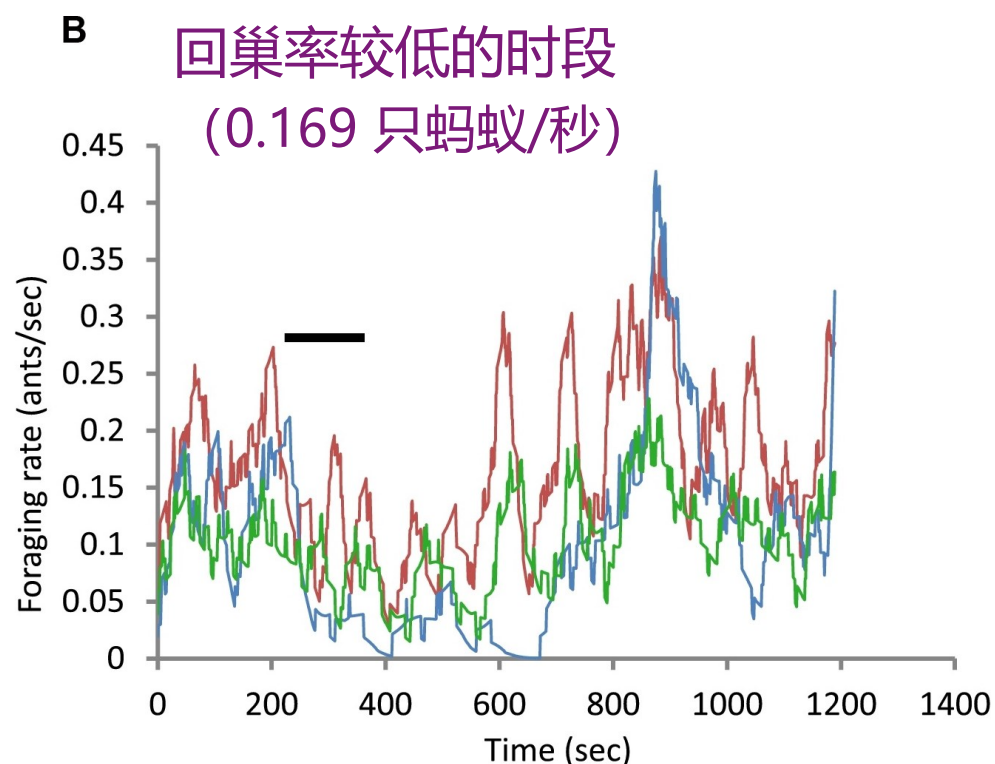
(4) $D_n \sim \text{Poisson}(\alpha_n)$ 来自图 1

模型与数据的比较



输出：预测的出巢觅食率 (α)
只改变 c (其余参数固定)，并报告最佳匹配结果。

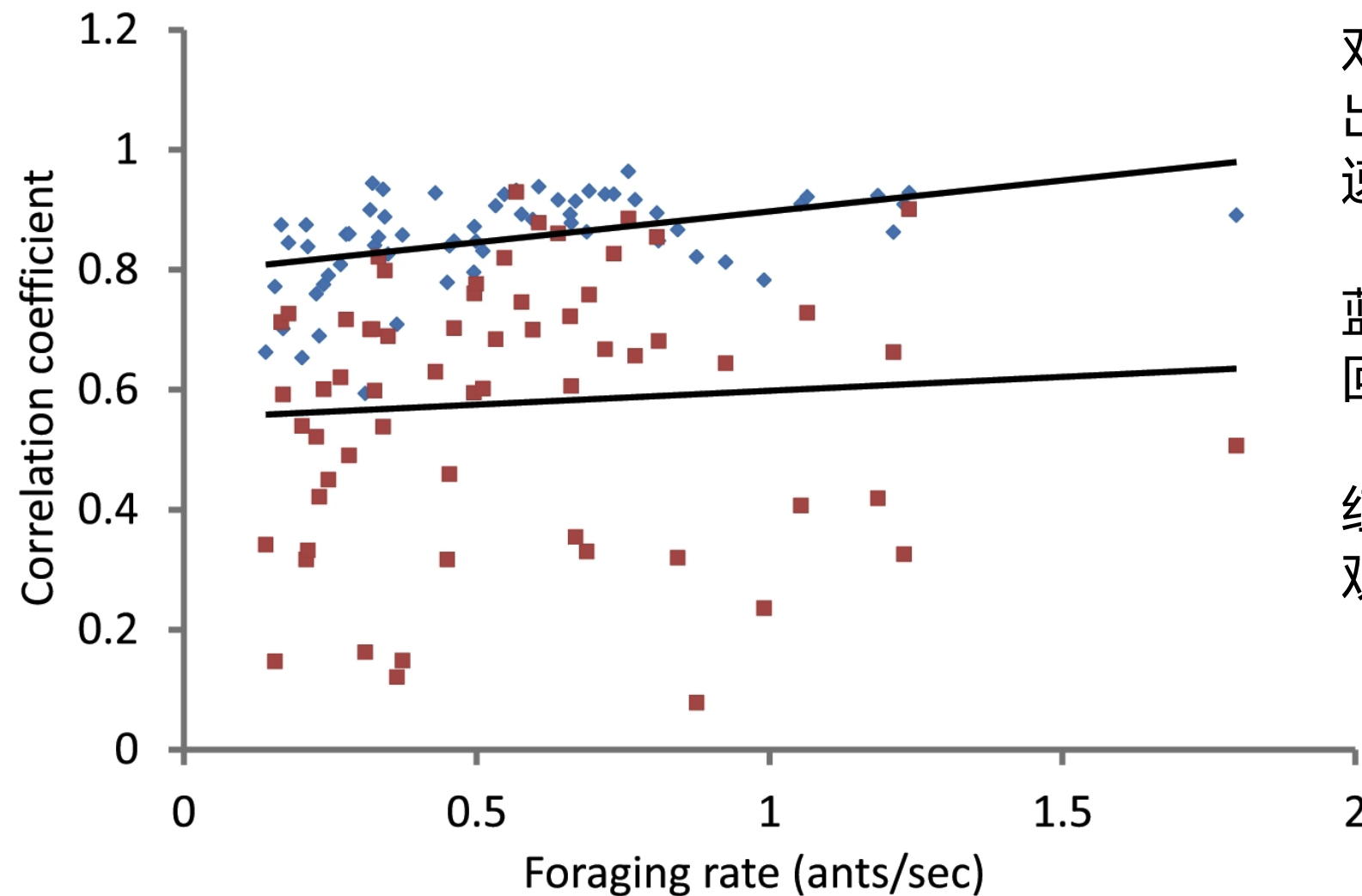
红线 = 回巢觅食蚁的速率
蓝线 = 实际的 α
绿线 = 预测的 α



结果：
* 移除成功的觅食蚁后，出巢率急剧下降（黑线）：当速率高时（A）下降明显，速率低时（B）则不那么明显。

出巢率与

活动水平



对每个点（试验），计算
出巢与回巢觅食
速率的相关性。

蓝色菱形 = $\text{corr}(\text{观测的}$
回巢率, 预测的出巢率)

红色方块 = $\text{corr}(\text{观测的回巢率},$
观测的出巢率)

试验期间回巢觅食蚁的平均速率

这意味着还有其他因素支配着出巢率，例如巢穴
结构与天气，而模型并未把它们考虑在内。

“蚂蚁互联网”

- 传输控制协议 (TCP) : 管理拥塞, 使互联网具有可扩展性
- 如何确定发送文件时的可用带宽:
 - 当源 A 向目的 B 发送数据时, 它把消息拆成数据包
 - 收到一个数据包后, B 向 A 发送 ACK
 - 若 ACK 返回慢 (返回率低) → 可用带宽少 → 源 A 降低发送速率 (发送速率变慢)
 - 若 ACK 返回快 (返回率高) → 可用带宽多 → 源 B 提高传输速率 (发送速率变快)

“蚂蚁互联网”

- 还有两点联系：
 - 慢启动：起初以可变的速率发送，以探测可靠性/带宽，然后才调整发送速率
 - 超时：当觅食蚁被阻止返回巢穴达 20 分钟时，几乎不再有蚂蚁离巢。
- 展望：蚂蚁有 11000 个物种，每一种都针对略有不同的目的、条件与环境而优化。还有许多其他算法有待发现.....