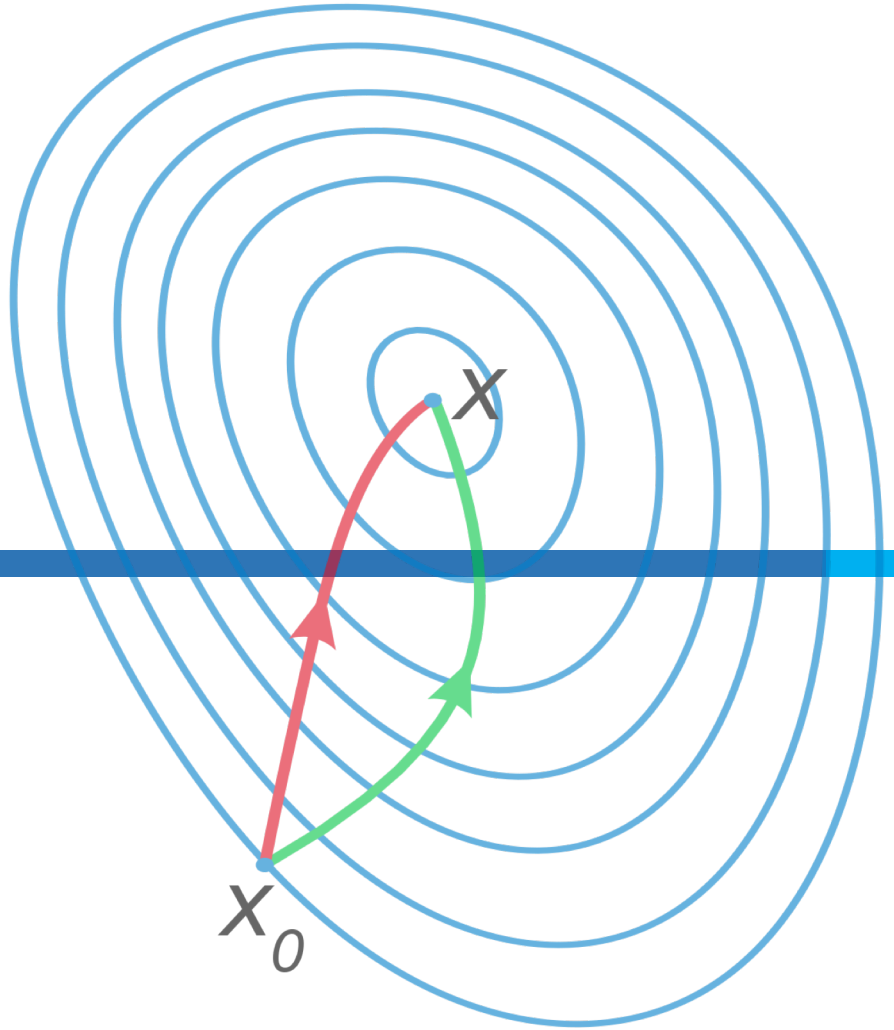
最优化方法



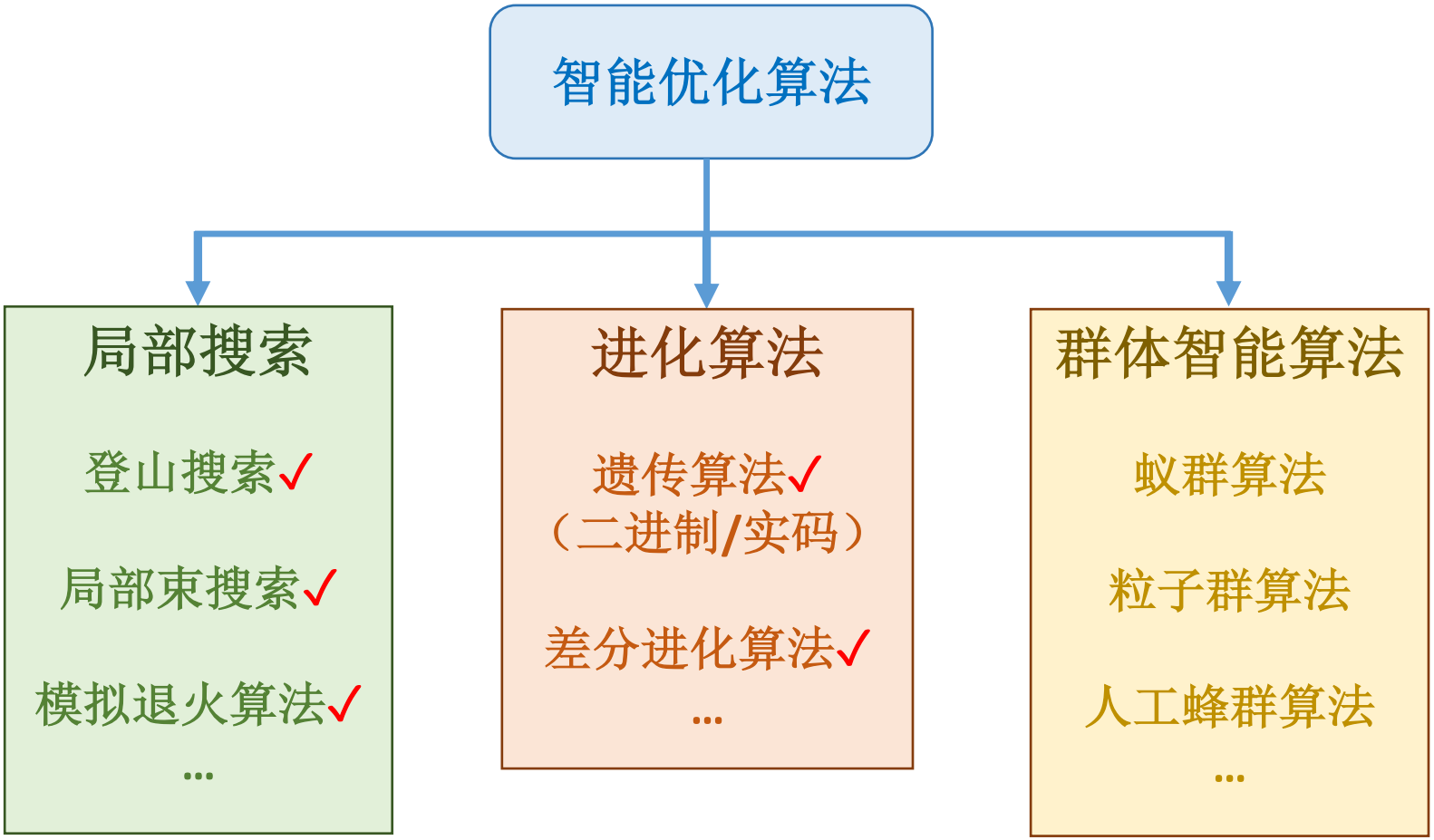
第七讲

计算机学院
余皓然

2023/4/23

群体智能之蚁群优化算法

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
局部搜索、模拟退火、遗传算法、差分进化算法、蚁群算法、粒子群优化算法、人工蜂群算法
Part4 针对较复杂的多目标最优化问题，如何找到较好解？


智能优化算法
局部搜索
登山搜索✓
局部束搜索✓
模拟退火算法✓
...
进化算法
遗传算法✓
（二进制/实码）
差分进化算法✓
...
群体智能算法
蚁群算法
粒子群算法
人工蜂群算法
...

群体智能算法

“群体智能”的两种不同含义

—— swarm intelligence（swarm: 指动物/昆虫的群体）

—— crowd intelligence（crowd: 指人群）

swarm intelligence

About 602,000 results (**0.05** sec)

Swarm intelligence
J Kennedy - Handbook of nature-inspired and innovative computing, 2006 - Springer
Swarm intelligence refers to a kind of problem-solving ability that emerges in the interactions of simple information-processing units. The concept of a swarm suggests multiplicity, stochasticity, randomness, and messiness, and the concept of intelligence suggests that the …
☆ 99 Cited by 10995 Related articles All 12 versions »

[BOOK] **Swarm intelligence**
RC Eberhart, Y Shi, J Kennedy - 2001 - books.google.com
Traditional methods for creating intelligent computational systems have privileged private" internal" cognitive and computational processes. In contrast, Swarm Intelligence argues that human intelligence derives from the interactions of individuals in a social world and further …
☆ 99 Cited by 1491 Related articles All 4 versions »

Swarm intelligence.
M Dorigo, M Birattari - Scholarpedia, 2007 - Springer
These proceedings contain the papers presented at ANTS 2016, the 10th International Conference on Swarm Intelligence, held at IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, during September 7–9, 2016. The ANTS series started in 1998 with the First …
☆ 99 Cited by 244 Related articles All 8 versions

crowd intelligence

About 714,000 results (**0.04** sec)

[HTML] **Crowd intelligence** in AI 2.0 era
W Li, W Wu, H Wang, X Cheng, H Chen, Z Zhou… - Frontiers of Information …, 2017 - Springer
The Internet based cyber-physical world has profoundly changed the information environment for the development of artificial **intelligence** (AI), bringing a new wave of AI research and promoting it into the new era of AI 2.0. As one of the most prominent …
☆ 99 Cited by 61 Related articles All 7 versions

Crowd intelligence enhances automated mobile testing
K Mao, M Harman, Y Jia - 2017 32nd IEEE/ACM International …, 2017 - ieeexplore.ieee.org
We show that information extracted from **crowd**-based testing can enhance automated mobile testing. We introduce Polariz, which generates replicable test scripts from **crowd**-based testing, extracting cross-appmotif'events: automatically-inferred reusable higher-level …
☆ 99 Cited by 41 Related articles All 5 versions

[HTML] **Crowd intelligence**: Analyzing online product reviews for preference measurement
S Xiao, CP Wei, M Dong - Information & Management, 2016 - Elsevier
The proliferation of product review websites produces a large, publicly accessible information resource for firms that seek to understand consumers' preferences. To facilitate product design or improvement, we propose a novel econometric preference measurement …
☆ 99 Cited by 65 Related articles All 4 versions

群体智能算法

“群体智能”的两种不同含义

—— swarm intelligence（swarm: 指动物/昆虫的群体）

通过模拟昆虫、兽群、鸟群、鱼群的群集行为搜索解（智能优化算法）

swarm intelligence

About 602,000 results (**0.05** sec)

Swarm intelligence

J Kennedy - Handbook of nature-inspired and innovative computing, 2006 - Springer

Swarm intelligence refers to a kind of problem-solving ability that emerges in the interactions of simple information-processing units. The concept of a swarm suggests multiplicity, stochasticity, randomness, and messiness, and the concept of intelligence suggests that the …

☆ 99 Cited by 10995 Related articles All 12 versions »

[BOOK] **Swarm intelligence**

RC Eberhart, Y Shi, J Kennedy - 2001 - books.google.com

Traditional methods for creating intelligent computational systems have privileged private" internal" cognitive and computational processes. In contrast, Swarm Intelligence argues that human intelligence derives from the interactions of individuals in a social world and further …

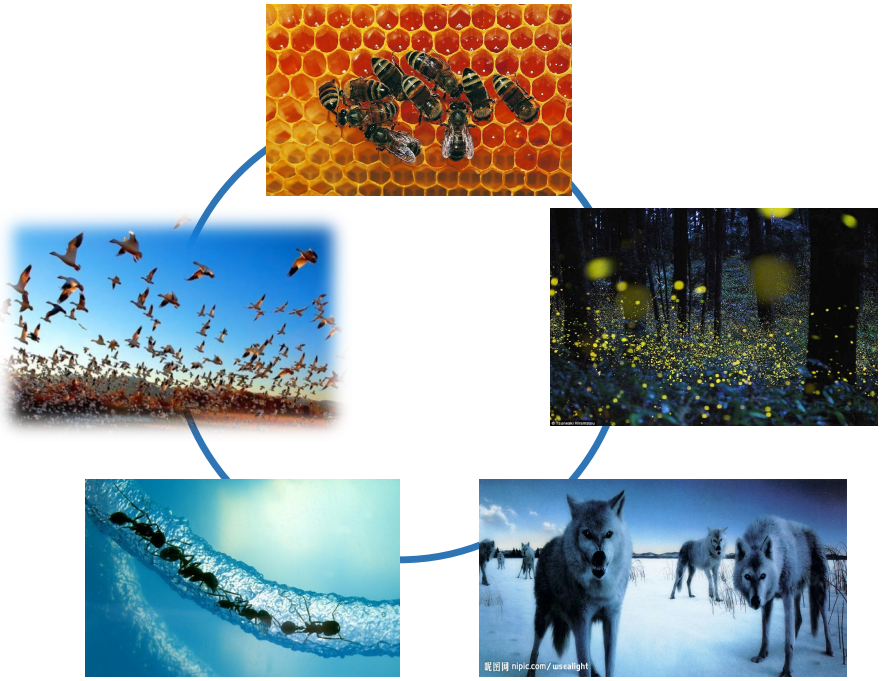
☆ 99 Cited by 1491 Related articles All 4 versions »

Swarm intelligence.

M Dorigo, M Birattari - Scholarpedia, 2007 - Springer

These proceedings contain the papers presented at ANTS 2016, the 10th International Conference on Swarm Intelligence, held at IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, during September 7–9, 2016. The ANTS series started in 1998 with the First …

☆ 99 Cited by 244 Related articles All 8 versions



群体智能算法

“群体智能”的两种不同含义

—— crowd intelligence（crowd: 指人群）

利用众人参与的形式完成特定任务（众包）

crowd intelligence

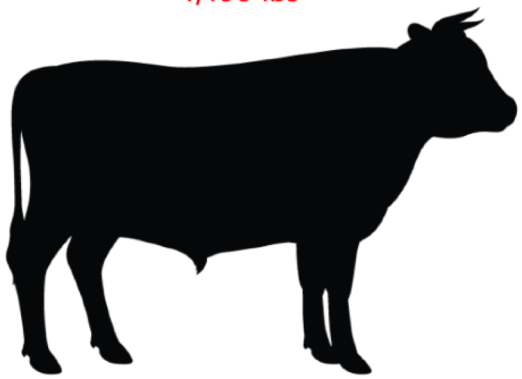
About 714,000 results (**0.04** sec)

[HTML] **Crowd intelligence** in AI 2.0 era
W Li, W Wu, H Wang, X Cheng, H Chen, Z Zhou… - Frontiers of Information …, 2017 - Springer
The Internet based cyber-physical world has profoundly changed the information environment for the development of artificial **intelligence** (AI), bringing a new wave of AI research and promoting it into the new era of AI 2.0. As one of the most prominent …
☆ 99 Cited by 61 Related articles All 7 versions

Crowd intelligence enhances automated mobile testing
K Mao, M Harman, Y Jia - 2017 32nd IEEE/ACM International …, 2017 - ieeexplore.ieee.org
We show that information extracted from **crowd**-based testing can enhance automated mobile testing. We introduce Polariz, which generates replicable test scripts from **crowd**-based testing, extracting cross-appmotif'events: automatically-inferred reusable higher-level …
☆ 99 Cited by 41 Related articles All 5 versions

[HTML] **Crowd intelligence**: Analyzing online product reviews for preference measurement
S Xiao, CP Wei, M Dong - Information & Management, 2016 - Elsevier
The proliferation of product review websites produces a large, publicly accessible information resource for firms that seek to understand consumers' preferences. To facilitate product design or improvement, we propose a novel econometric preference measurement …
☆ 99 Cited by 65 Related articles All 4 versions

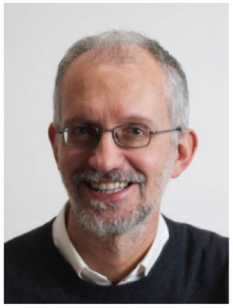
Actual weight of the ox on show:
1,198 lbs



Average guess of 787 people who took part:
1,207 lbs

蚁群优化算法（Ant colony optimization algorithm）

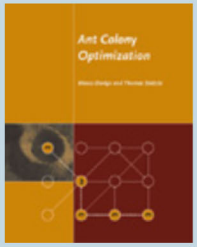
1992年Marco Dorigo在其博士学位论文中首次介绍



Agrégé de l'Enseignement Supérieur, Université Libre de Bruxelles, Belgium, 1995.
Ph.D. in System and Information Engineering, Politecnico di Milano, Italy, 1992.
He is the inventor of the Ant Colony Optimization metaheuristic for combinatorial optimization problems.
He is research director for the Belgian Fonds de la Recherche Scientifique.
He is co-director of the IRIDIA lab at the Université Libre de Bruxelles.

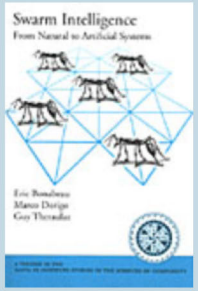
Curriculum Vitae
Last updated September 2018
Download PDF file (478 Kb)

Authored books

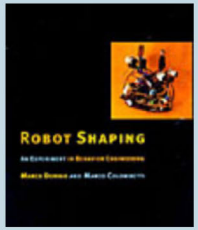


Ant Colony Optimization
MIT Press, 2004

Ant Colony Optimization (Chinese transalation)
China-Pub.com, 2006



Swarm intelligence
Oxford University Press, 1999



Robot Shaping
MIT Press, 1998

Marco Dorigo
的Google Scholar
总引约13万次

蚁群优化算法

蚂蚁寻食物

蚂蚁模拟 Colony simulation

https://www.bilibili.com/video/BV1JB4y1P7mk?from=search&seid=5718585914166882441

蚂蚁寻食机制

通过分别与环境交互的方式实现间接通信

蚂蚁视力不好，在爬过的地方留下信息素（pheromone）

通过信息素实现间接通信

某路径信息素浓度越高，后续蚂蚁选择该路径概率越大

双桥实验

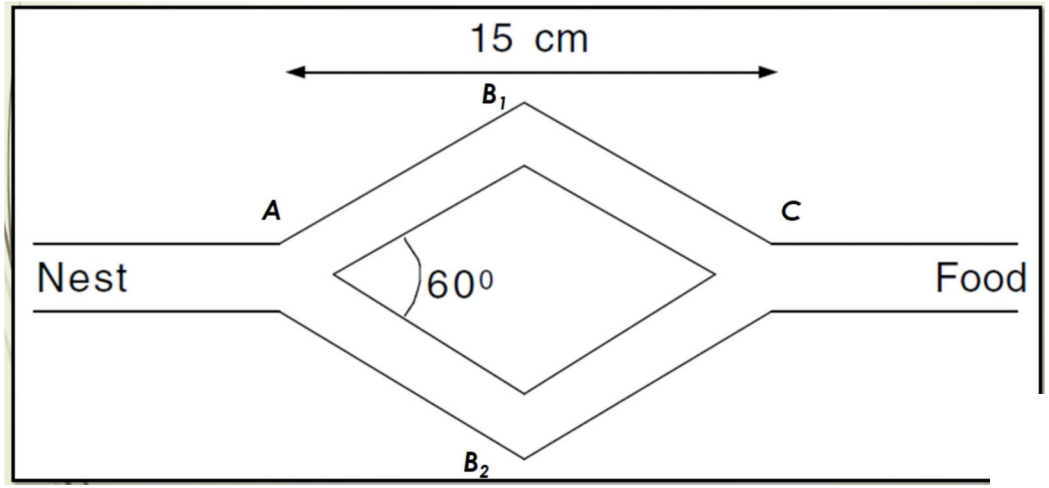
分析信息素对蚂蚁路径选择的影响

实验假设：

（1）所有蚂蚁爬行速度一样；（2）所有蚂蚁释放等量的信息素；

（3）蚂蚁在往返同一路径时释放的信息素没有差别

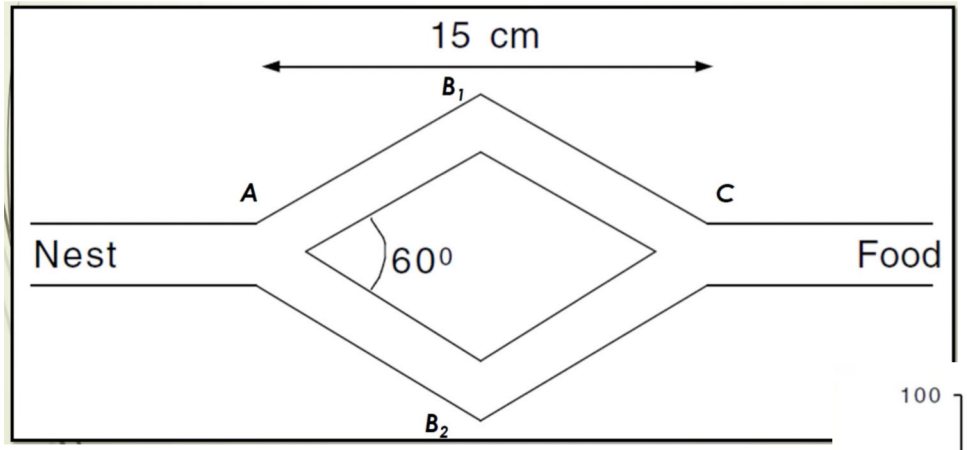
双桥实验



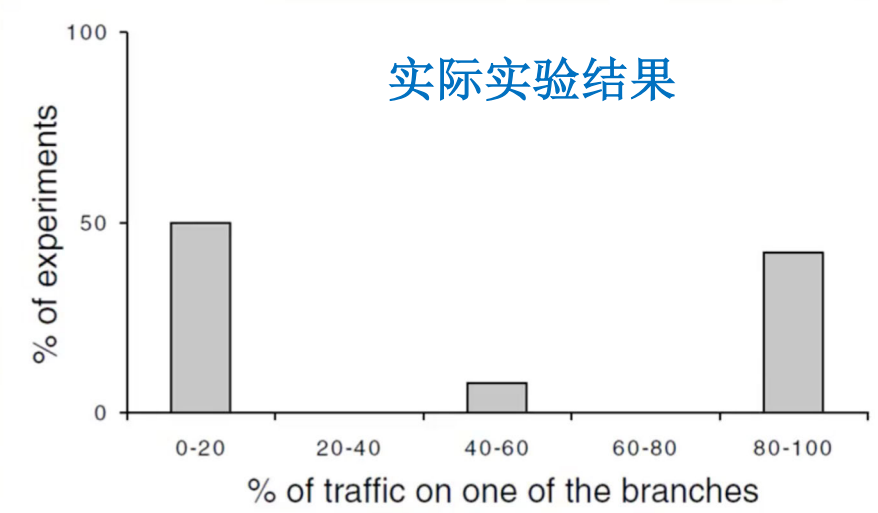
设置1

内容取自Jeff Lettman “An Introduction to Ant colony optimization”

双桥实验

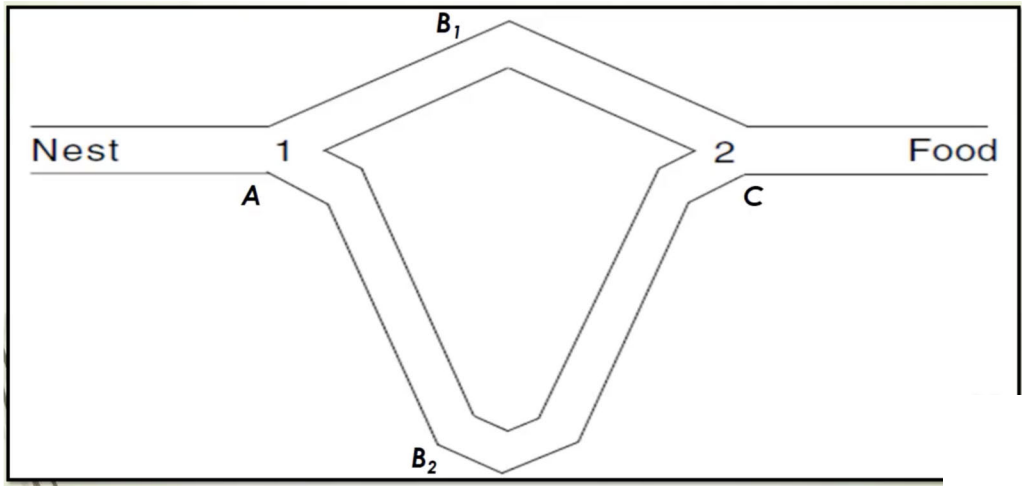


实际实验结果



内容取自Jeff Lettman “An Introduction to Ant colony optimization”

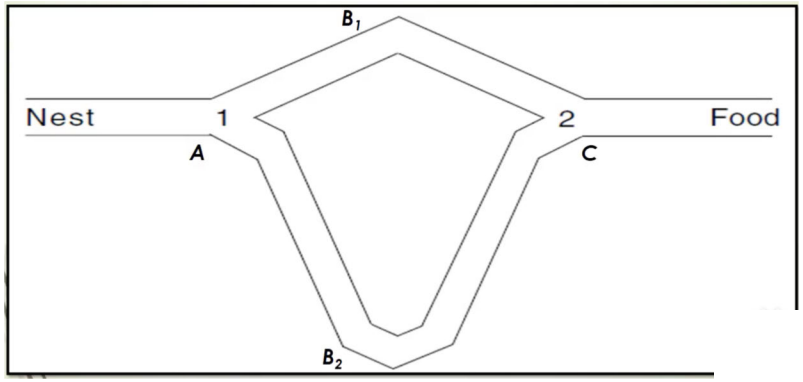
双桥实验

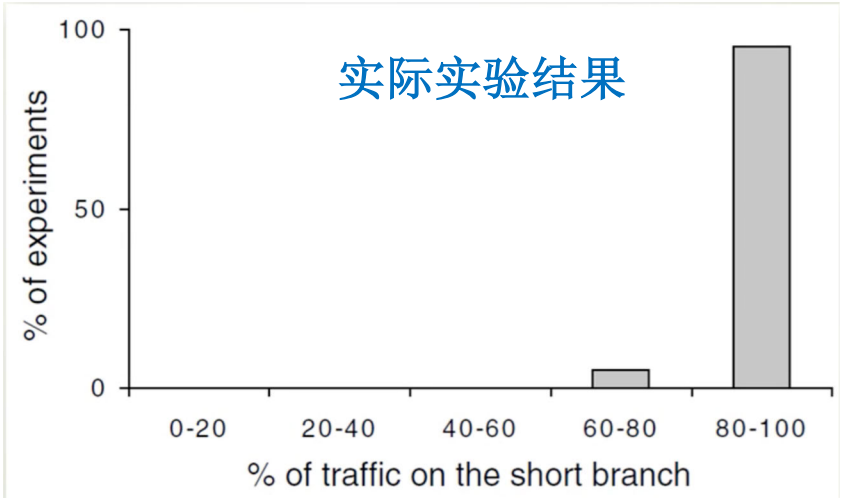


设置2

内容取自Jeff Lettman “An Introduction to Ant colony optimization”

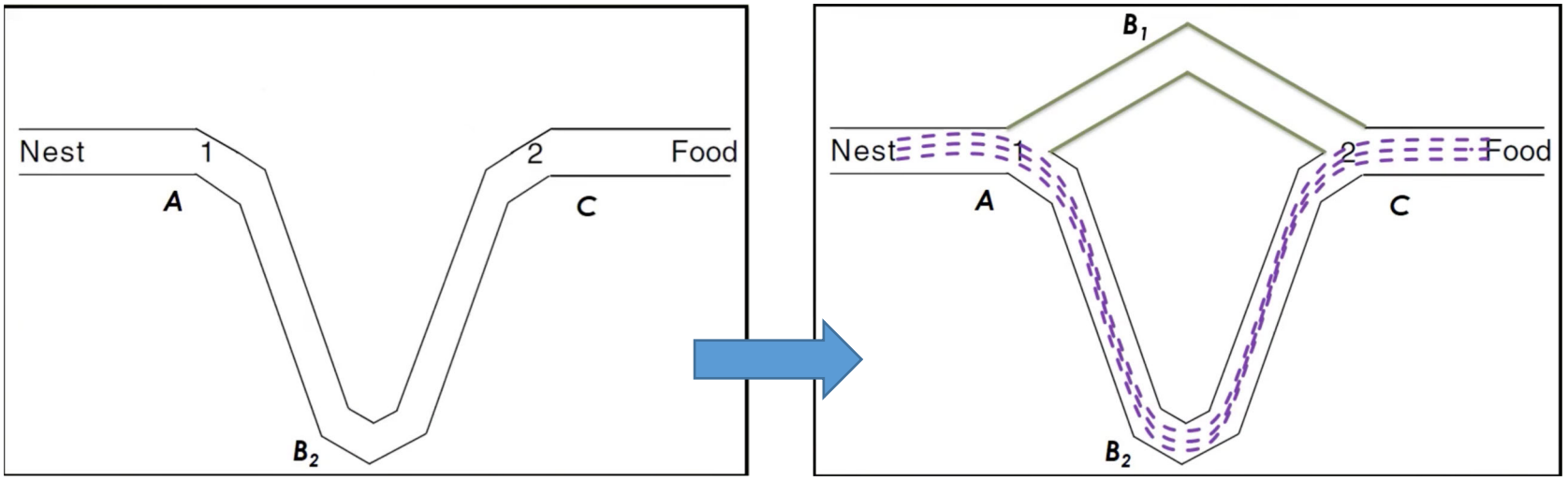
双桥实验





内容取自Jeff Lettman “An Introduction to Ant colony optimization”

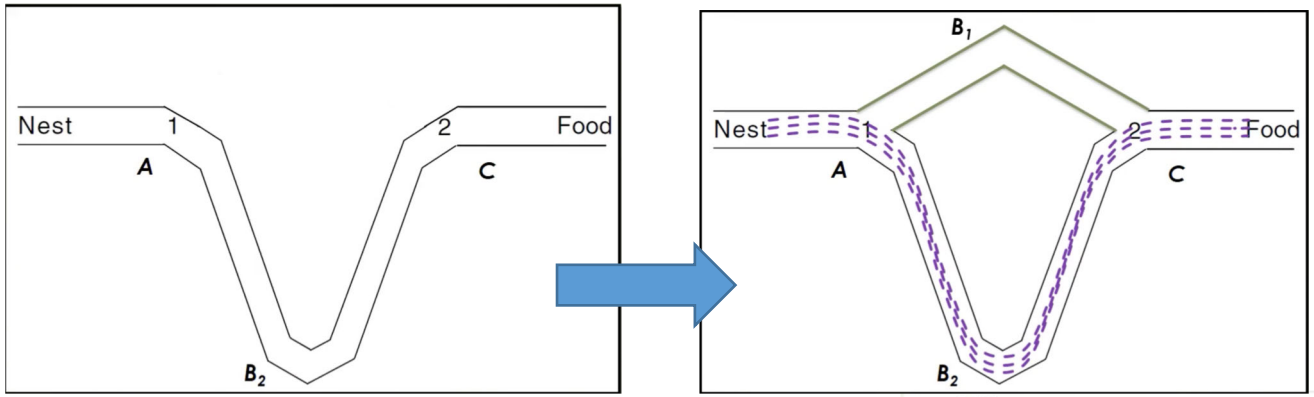
双桥实验



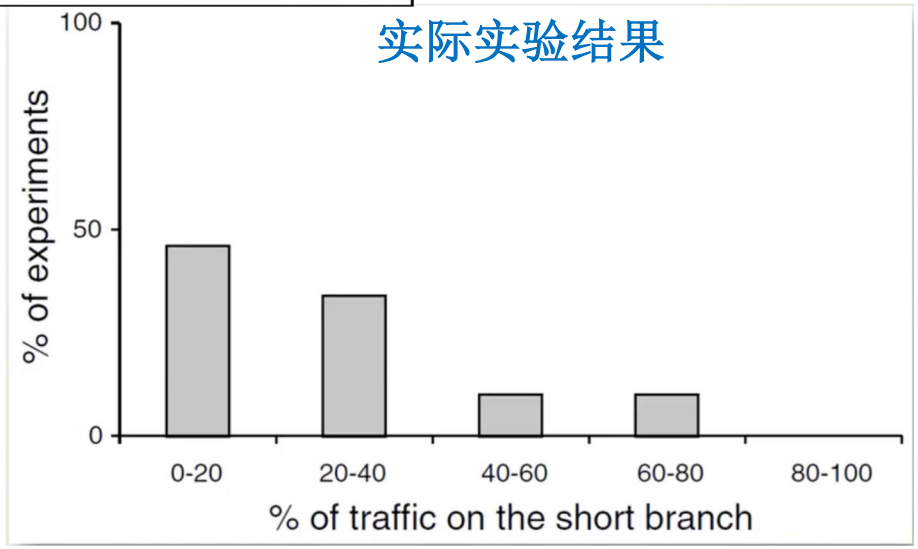
设置3

内容取自Jeff Lettman “An Introduction to Ant colony optimization”

双桥实验



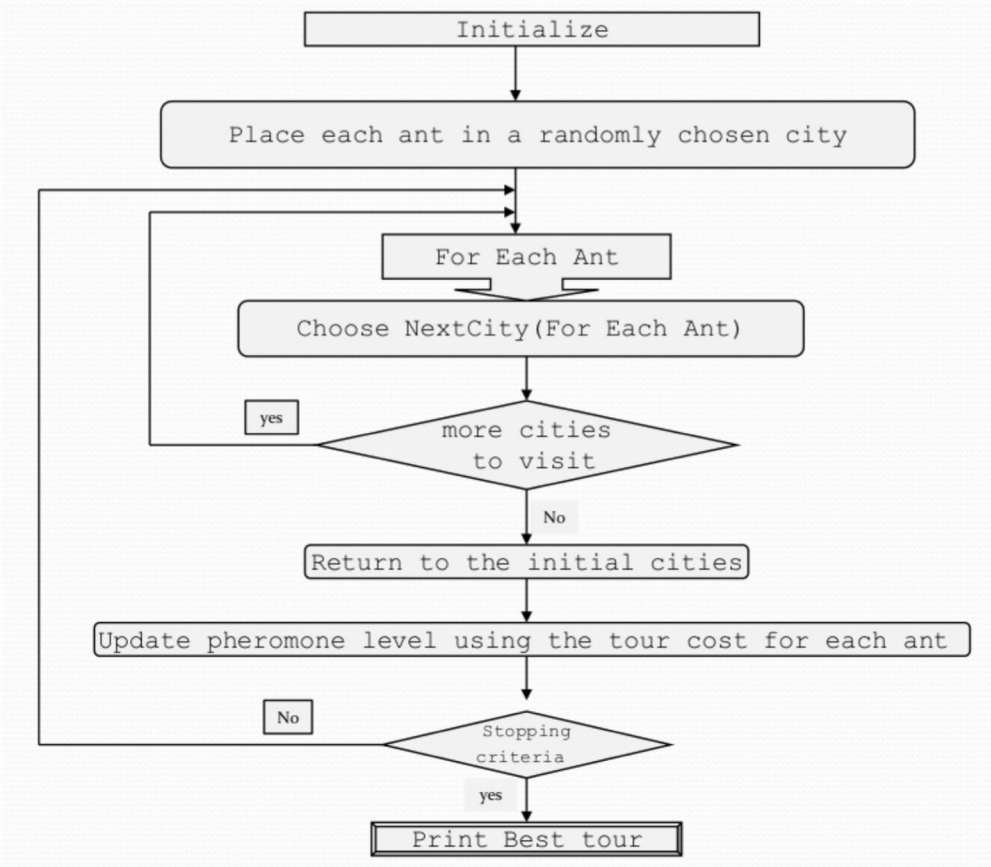
实际实验结果



内容取自Jeff Lettman “An Introduction to Ant colony optimization”

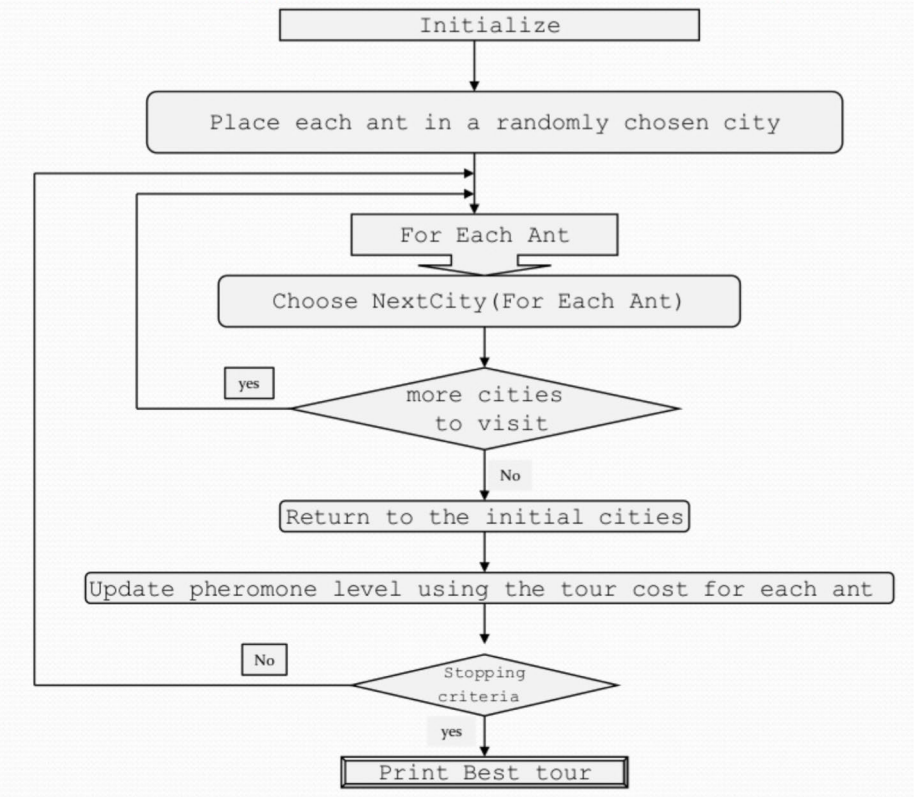
蚁群系统

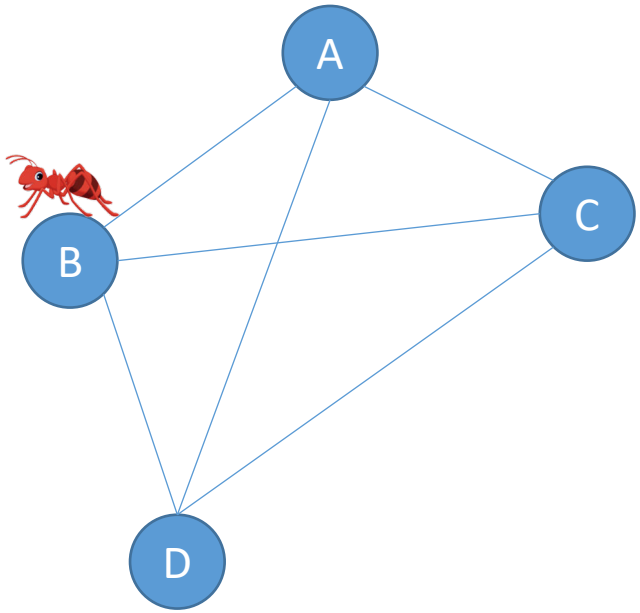
蚁群系统（ant system）是第一个蚁群优化算法，被用于解决旅行商问题



图片取自Joy Dutta课件“Ant colony optimization”

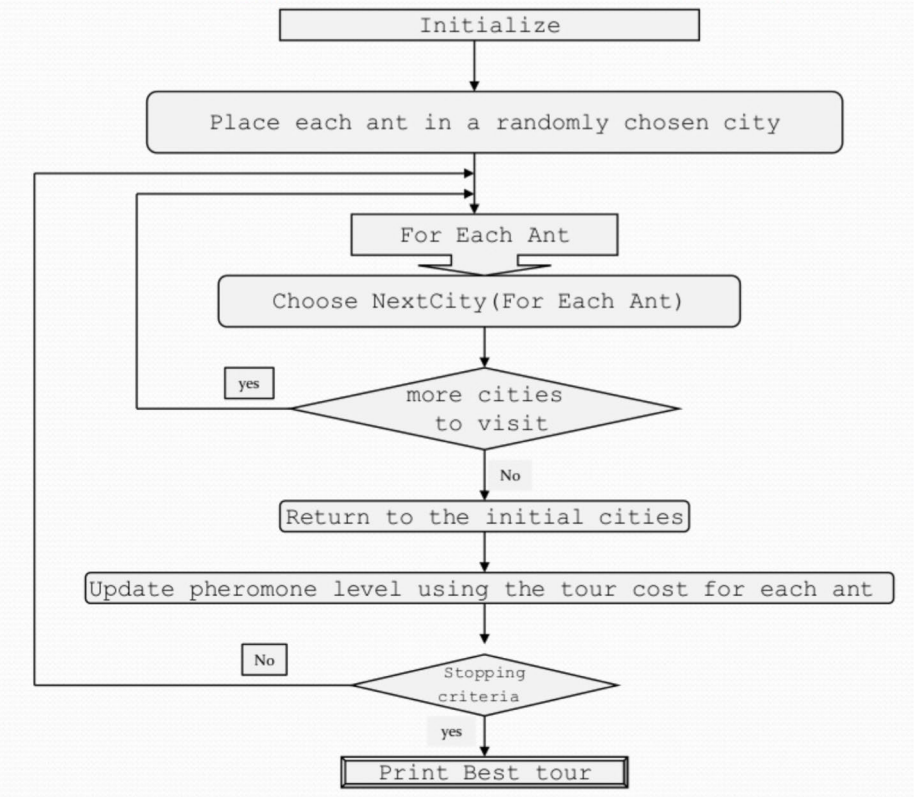
蚁群系统

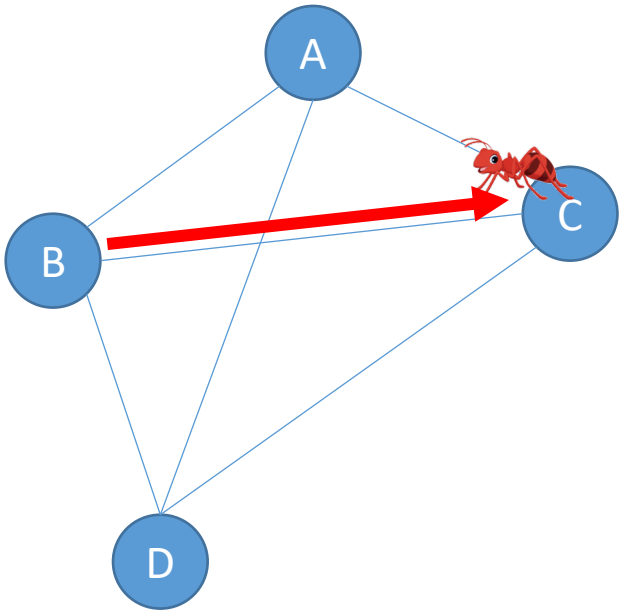




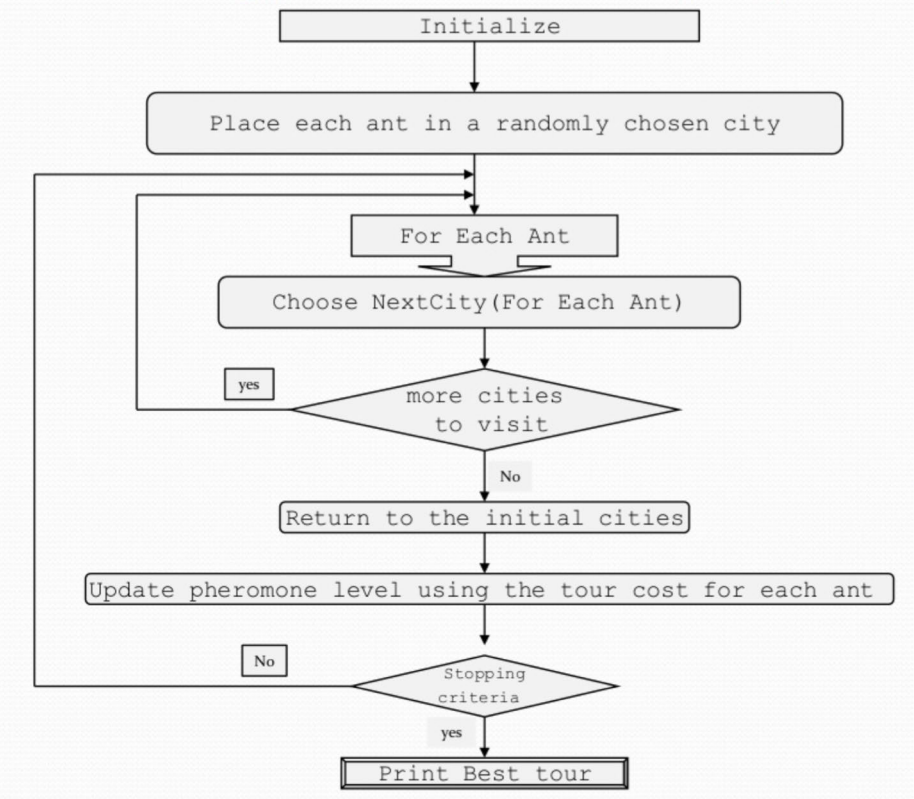
图片取自Joy Dutta课件“Ant colony optimization”

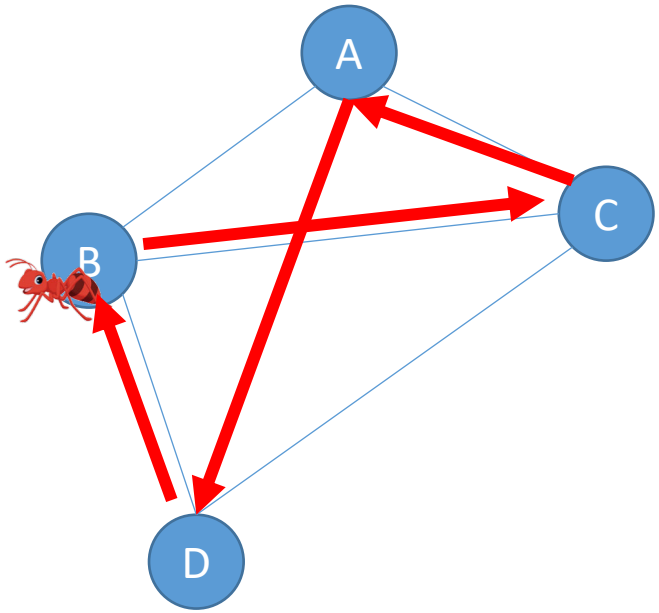
蚁群系统


Initialize
Place each ant in a randomly chosen city
For Each Ant
Choose NextCity(For Each Ant)
yes
more cities to visit
No
Return to the initial cities
Update pheromone level using the tour cost for each ant
No
Stopping criteria
yes
Print Best tour


A
B
C
D

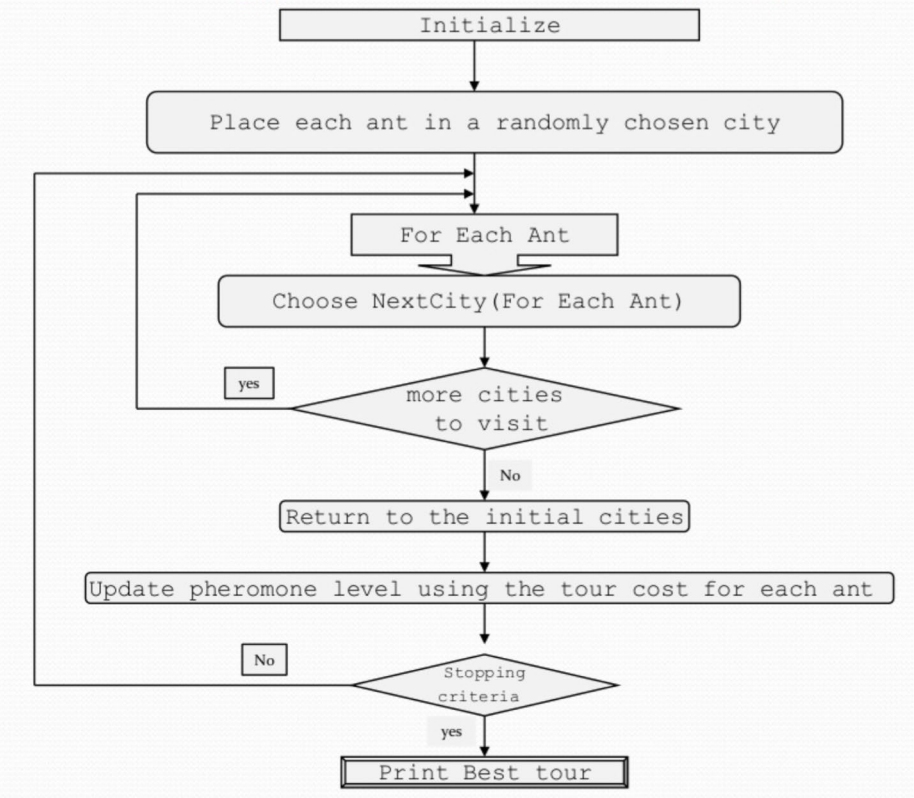
蚁群系统

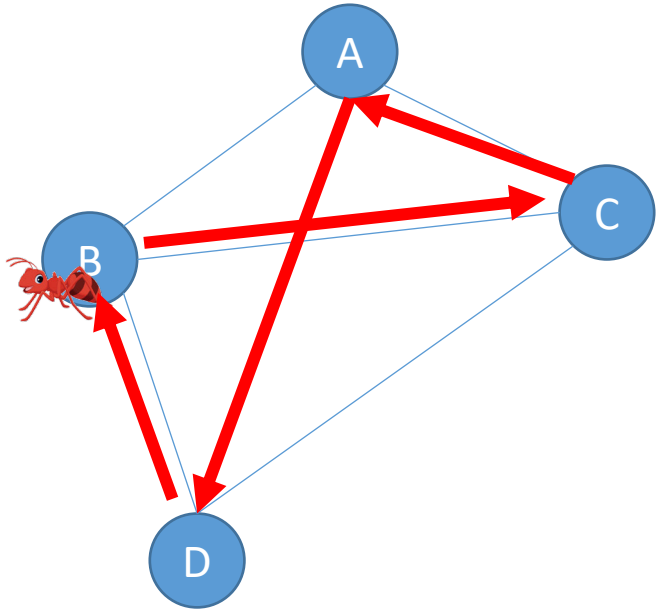




每只蚂蚁每次如何选择下一个城市？

蚁群系统





每只蚂蚁每次如何选择下一个城市？

$$p_k(r,s) = \frac{T(r,s)\cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c)\cdot H(r,c)^{\beta}}$$

k: 第k只蚂蚁

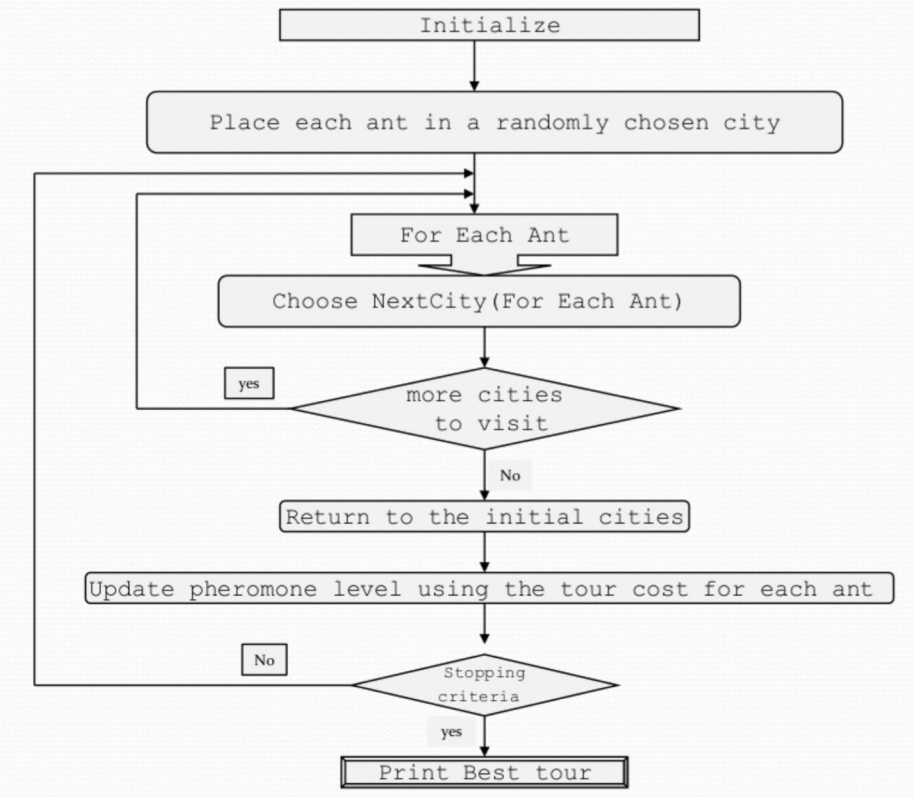
r: 现在城市； s和c: 可能选择的城市

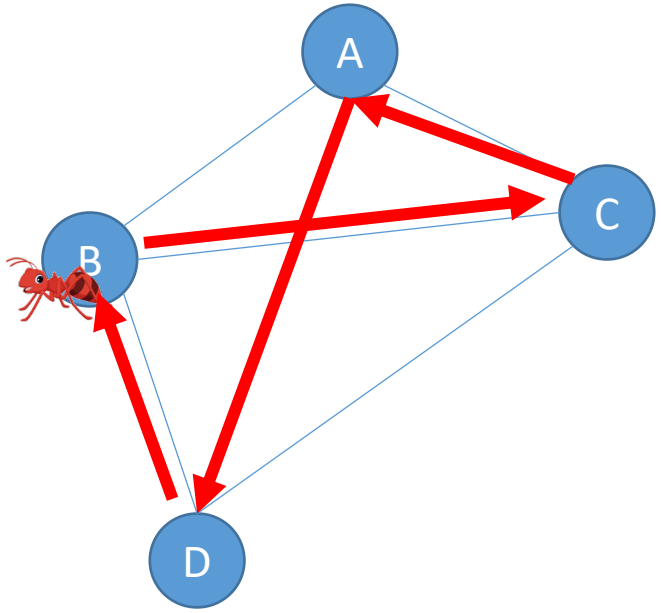
T(r, s): 信息素浓度

H(r, s): 距离倒数; β: 控制参数

有的版本会对T(r, s)的指数位置加控制参数

蚁群系统





信息素浓度如何更新？

$$T(r,s) \leftarrow \rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

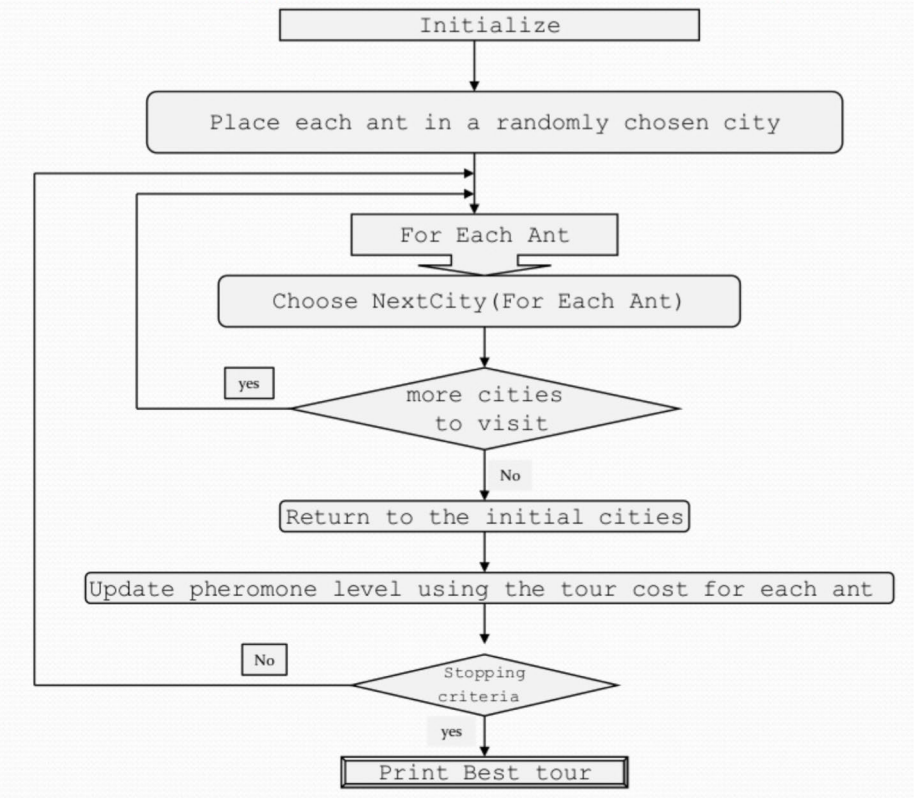
ρ: 衰减系数

m: 蚂蚁总数

$A_k(r, s)$: 如果蚂蚁k没走过rs，值为0

如果蚂蚁k走过rs，值为其总路程的倒数

蚁群系统



$$p_k(r,s) = \frac{T(r,s)\cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c)\cdot H(r,c)^{\beta}}$$

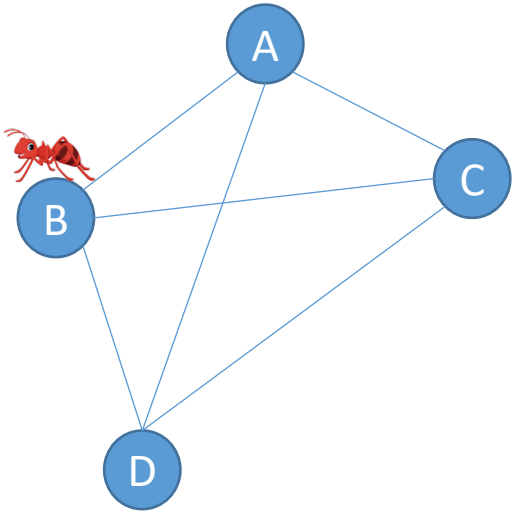
$$T(r,s) \leftarrow \rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

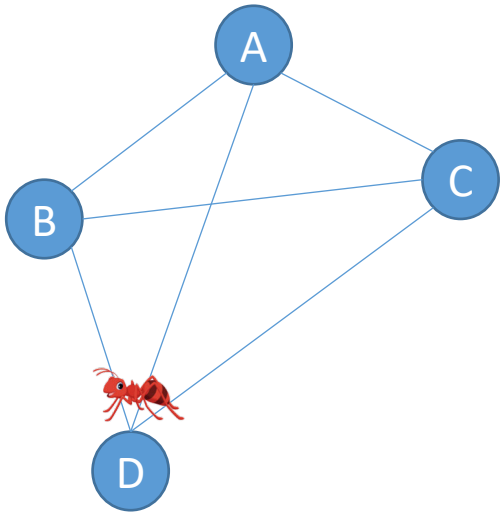
注意每次都是重新随机放置蚂蚁（相当于之前的蚂蚁死掉了）

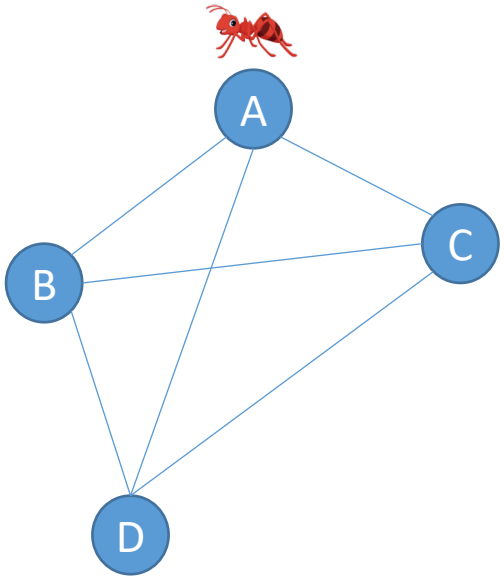
与遗传算法、差分进化算法不同

蚁群系统

每次迭代: 随机放置m只蚂蚁

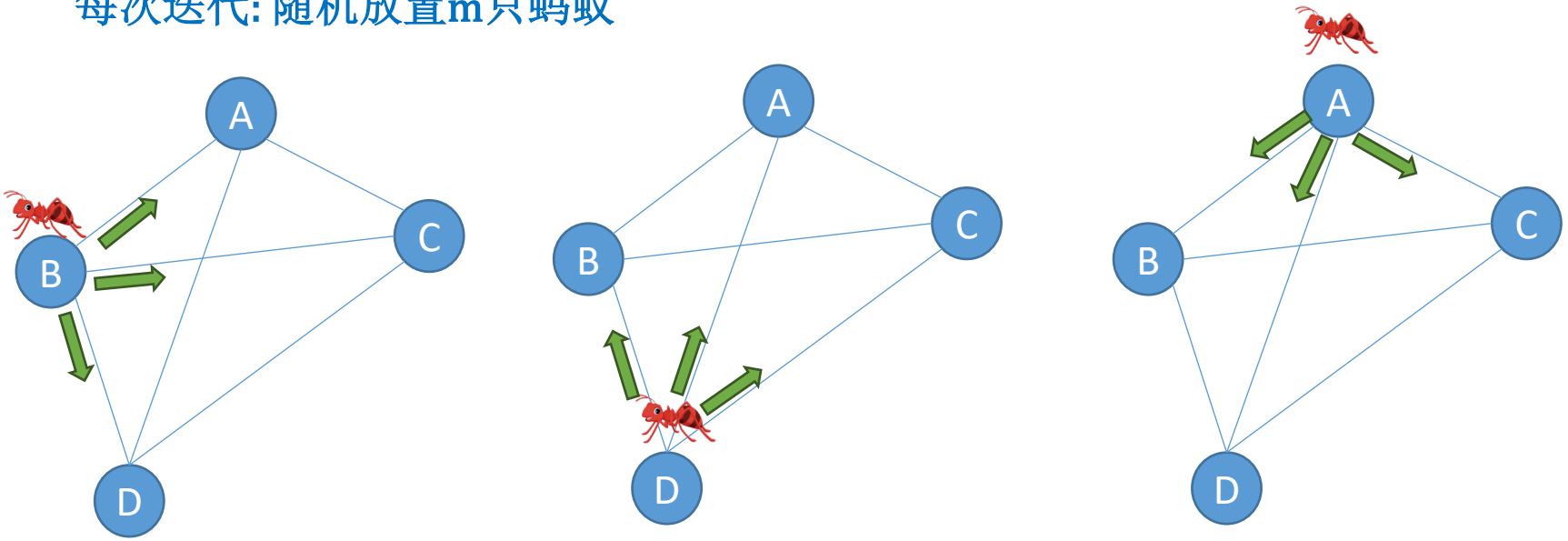






蚁群系统

每次迭代: 随机放置m只蚂蚁

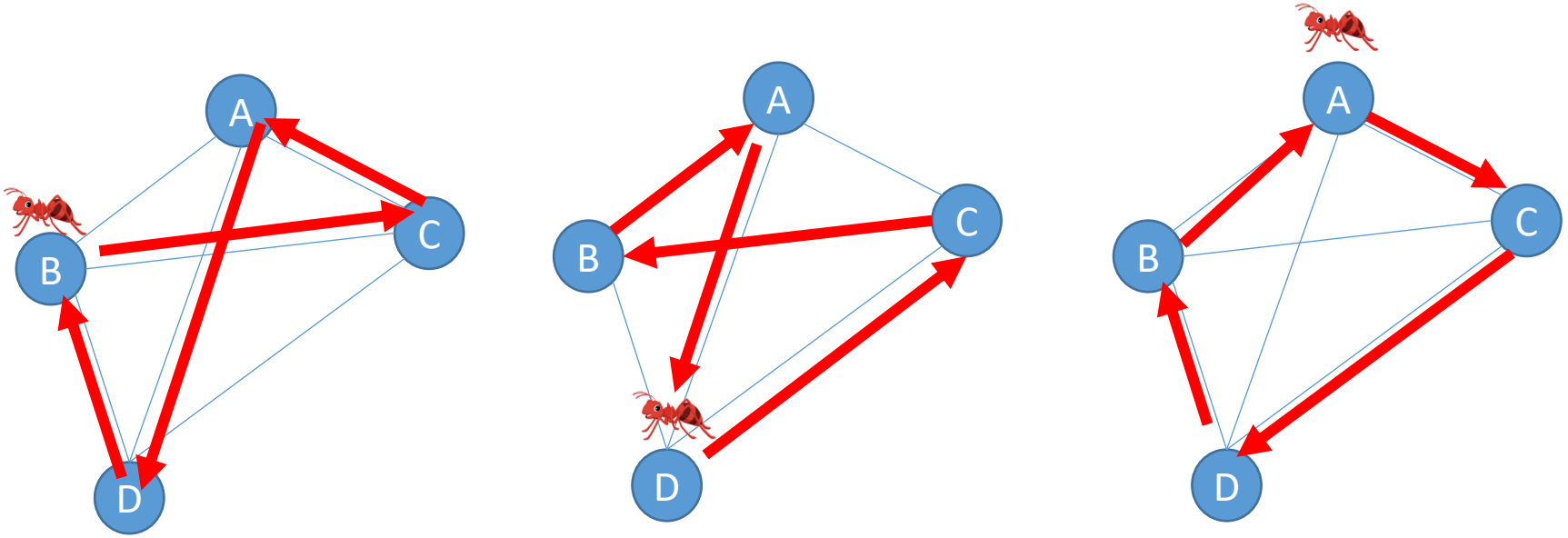


每只蚂蚁每次如何选择下一个城市？

$$p_k(r,s) = \frac{T(r,s)\cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c)\cdot H(r,c)^{\beta}}$$

k: 第k只蚂蚁
r: 现在城市； s和c: 可能选择的城市
T(r, s): 信息素浓度
H(r, s): 距离倒数; β: 控制参数

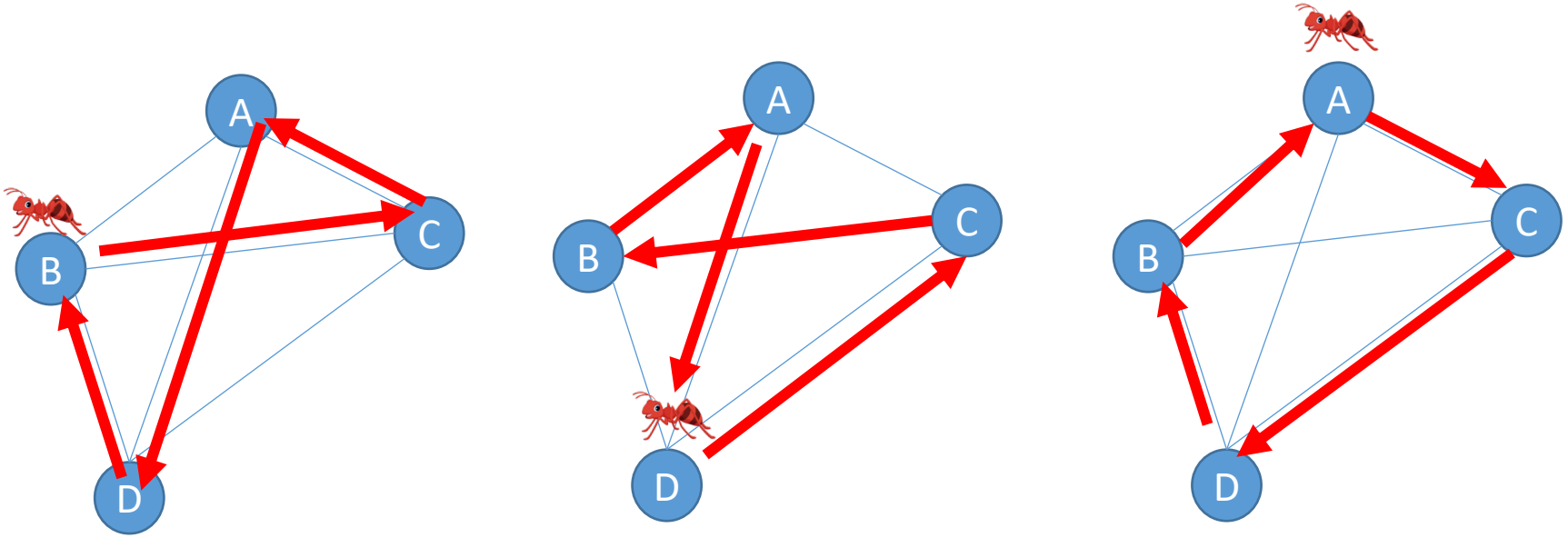
蚁群系统



根据各蚂蚁走过路径的总长更新相应各边的信息素浓度

思想: 路径总长越短，路径包含各边新增的信息素浓度越高

蚁群系统



$$T(r,s) \leftarrow \rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

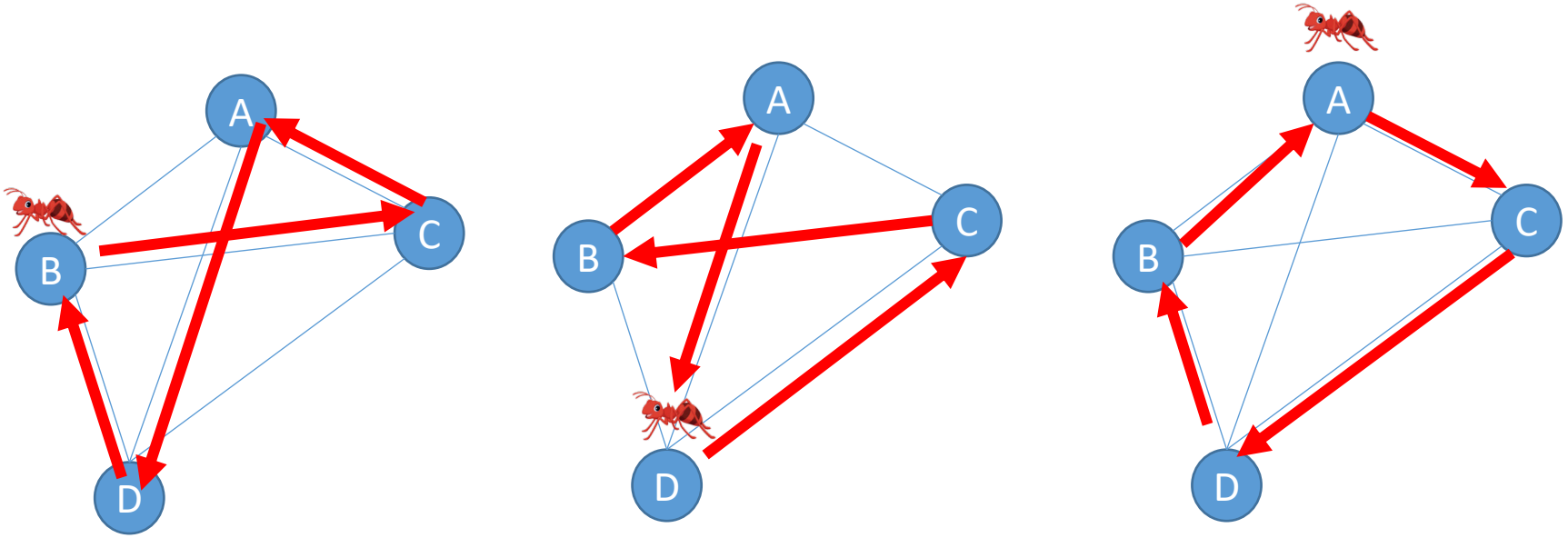
ρ: 衰减系数

m: 蚂蚁总数

$A_k(r, s)$: 如果蚂蚁k没走过rs，值为0

如果蚂蚁k走过rs，值为总路程倒数

蚁群系统



比如计算T(D, B):

$$T(r,s) \leftarrow \rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

$$T(D,B) \leftarrow \rho T(D,B) + \frac{1}{\text{lenth}_1} + \frac{1}{\text{lenth}_3}$$

蚁群优化

Ant system (AS) ✓

Ant colony system (ACS)

Elitist ant system

Max-min ant system (MMAS)

Rank-based ant system (ASrank) ✓

Continuous orthogonal ant colony (COAC)

Recursive ant colony optimization

基于排序的蚁群系统

蚁群系统（ant system）

所有m只蚂蚁都对此项贡献

$$p_k(r,s)=\frac{T(r,s)\cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c)\cdot H(r,c)^{\beta}}$$

随机生成新解

$$T(r,s)\leftarrow \rho\cdot T(r,s)+\sum_{k=1}^{m} A_k(r,s)$$

更新信息素浓度

基于排序的蚁群系统（rank-based ant system）

思想: 改进更新信息素浓度方法

对蚂蚁排序，加强排序在前的蚂蚁对信息素浓度的影响

对所有蚂蚁根据路径总长由短到长排序，只允许前面若干蚂蚁对此项贡献

基于排序的蚁群系统

所有m只蚂蚁都对此项贡献

蚁群系统（ant system）

$$T(r,s) \leftarrow \rho \cdot T(r,s) + \sum_{k=1}^{m} A_k(r,s)$$

基于排序的蚁群系统（rank-based ant system）

$$T(r,s) \leftarrow \rho T(r,s) + \sum_{R=1}^{w-1} (w-R) A^R(r,s) + w A^{\text{best}}(r,s)$$

w只蚂蚁: (1) w-1只当前迭代回合所有m只蚂蚁中排前面的w-1只

(2) 历史上所有尝试过的蚂蚁中最佳的蚂蚁

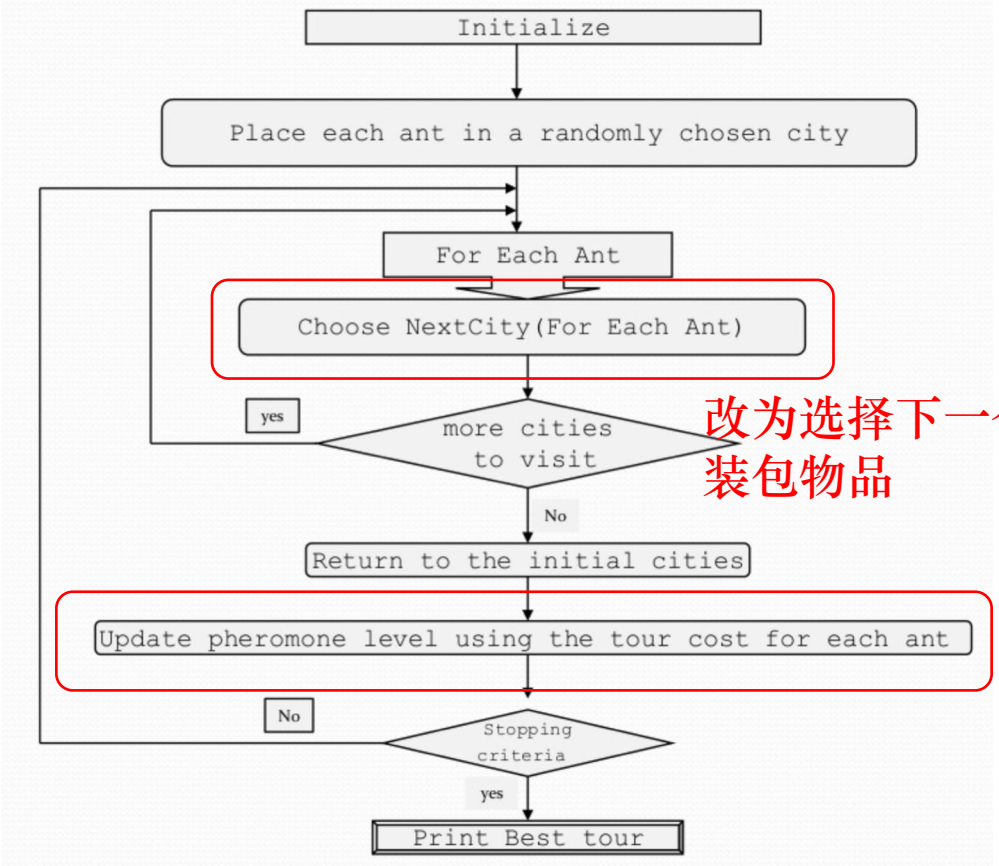
0-1背包问题

$$\max \sum_{i=1}^{n} z_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

如何用类似思路求解0-1背包问题?



0-1背包问题

$$\max \sum_{i=1}^{n} z_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

V_c:背包剩余空间　　N_i:剩余可放入背包物品集合

begin
  while (a cycle exists) **do**
    while (an ant k, which has not yet worked, exists) **do**
      while ($V_c \ge 0$) **do**
        select a next object o_j from N_i with probability $p_j = \begin{cases} \frac{\tau_j^\alpha \mu_j^\beta}{\sum_{j \in N_i} \tau_j^\alpha \mu_j^\beta}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases}$
        add a selected object to a partial solution $S = S + \{o_j\}$
        update the current knapsack load capacity $V_C = V_C - w_j$
        update the profit $Z = Z + z_j$
        update the neighbourhood of the current state $N_i = \{o_i : w_i \le V_C\}$
      end
      remember the best solution if a better solution has been found
    end
    remember a global best solution if a better solution has been found
    use an evaporation mechanism $\tau = \rho\tau$
    update pheromone trails $\tau = \tau + \Delta\tau$
  end
end.

伪码取自Krzysztof Schiff, “Ant colony optimization algorithm for the 0-1 knapsack problem”

0-1背包问题

$$\max \sum_{i=1}^{n} z_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

V_c:背包剩余空间　　N_i:剩余可放入背包物品集合

```
begin
    while (a cycle exists) do
        while (an ant k, which has not yet worked, exists) do
            while (V_c ≥ 0) do
                select a next object o_j from N_i with probability p_j = { τ_j^α μ_j^β / Σ_{j∈N_i} τ_j^α μ_j^β , for j ∈ N_i
                                                                         { 0,                                  for j ∉ N_i
                add a selected object to a partial solution S = S + {o_j}
                update the current knapsack load capacity V_C = V_C − w_j
                update the profit Z = Z + z_j
                update the neighbourhood of the current state N_i = {o_i : w_i ≤ V_C}
            end
            remember the best solution if a better solution has been found
        end
        remember a global best solution if a better solution has been found
        use an evaporation mechanism τ = ρτ
        update pheromone trails τ = τ + Δτ
    end
end.
```

信息素浓度　　物品本身特征

$$p_j = \begin{cases} \dfrac{\tau_j^{\alpha}\mu_j^{\beta}}{\sum_{j\in N_i}\tau_j^{\alpha}\mu_j^{\beta}}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases}$$

如何更新信息素浓度？如何定义物品特征μ_j？

0-1背包问题

$$\max \sum_{i=1}^{n} z_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

V_c:背包剩余空间　　N_i:剩余可放入背包物品集合

```
begin
    while (a cycle exists) do
        while (an ant k, which has not yet worked, exists) do
            while (V_c ≥ 0) do
                select a next object o_j from N_i with probability p_j = { τ_j^α μ_j^β / Σ_{j∈N_i} τ_j^α μ_j^β , for j ∈ N_i
                                                                         { 0,                                  for j ∉ N_i
                add a selected object to a partial solution S = S + {o_j}
                update the current knapsack load capacity V_C = V_C − w_j
                update the profit Z = Z + z_j
                update the neighbourhood of the current state N_i = {o_i : w_i ≤ V_C}
            end
            remember the best solution if a better solution has been found
        end
        remember a global best solution if a better solution has been found
        use an evaporation mechanism τ = ρτ
        update pheromone trails τ = τ + Δτ
    end
end.
```

信息素浓度　　物品本身特征

$$p_j = \begin{cases} \dfrac{\tau_j^{\alpha}\mu_j^{\beta}}{\sum_{j\in N_i}\tau_j^{\alpha}\mu_j^{\beta}}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases}$$

特征μ_j可以取$\frac{z_j}{w_j}$

0-1背包问题

$$\max \sum_{i=1}^{n} z_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

V_c:背包剩余空间　　N_i:剩余可放入背包物品集合

```
begin
    while (a cycle exists) do
        while (an ant k, which has not yet worked, exists) do
            while (V_c ≥ 0) do
                select a next object o_j from N_i with probability p_j = { τ_j^α μ_j^β / Σ_{j∈N_i} τ_j^α μ_j^β , for j ∈ N_i
                                                                         { 0,                                  for j ∉ N_i
                add a selected object to a partial solution S = S + {o_j}
                update the current knapsack load capacity V_C = V_C − w_j
                update the profit Z = Z + z_j
                update the neighbourhood of the current state N_i = {o_i : w_i ≤ V_C}
            end
            remember the best solution if a better solution has been found
        end
        remember a global best solution if a better solution has been found
        use an evaporation mechanism τ = ρτ
        update pheromone trails τ = τ + Δτ
    end
end.
```

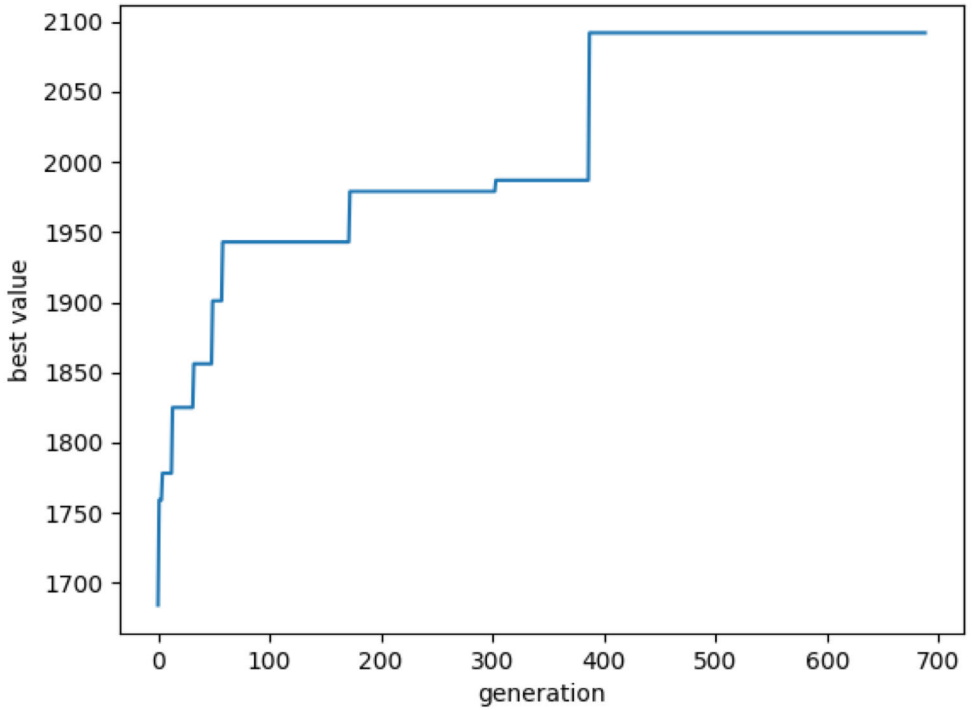
信息素浓度　　物品本身特征

$$p_j = \begin{cases} \dfrac{\tau_j^{\alpha}\mu_j^{\beta}}{\sum_{j\in N_i}\tau_j^{\alpha}\mu_j^{\beta}}, & \text{for } j \in N_i \\ 0, & \text{for } j \notin N_i \end{cases}$$

$\Delta\tau_j$可以取 $\begin{cases} 0, & \text{若当前迭代回合最佳方案不含物品j,} \\ \dfrac{1}{1+\frac{z_{best}-z}{z_{best}}}, & \text{若当前迭代回合最佳方案含物品j.} \end{cases}$

0-1背包问题

n=100, C=1000 代码结果示例：



群体智能之粒子群优化

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
局部搜索、模拟退火、遗传算法、差分进化算法、蚁群算法、粒子群优化算法、人工蜂群算法
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

粒子群优化

粒子群优化（Particle Swarm Optimization）

受鸟群、鱼群的群体活动启发而设计

粒子群优化

International Conference on Neural Networks

Particle swarm optimization

J Kennedy, R Eberhart - Proceedings of ICNN'95-international …, 1995 - ieeexplore.ieee.org

A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and …

☆ 99 Cited by 65809 Related articles All 34 versions

思想：将个体经验和群体经验结合

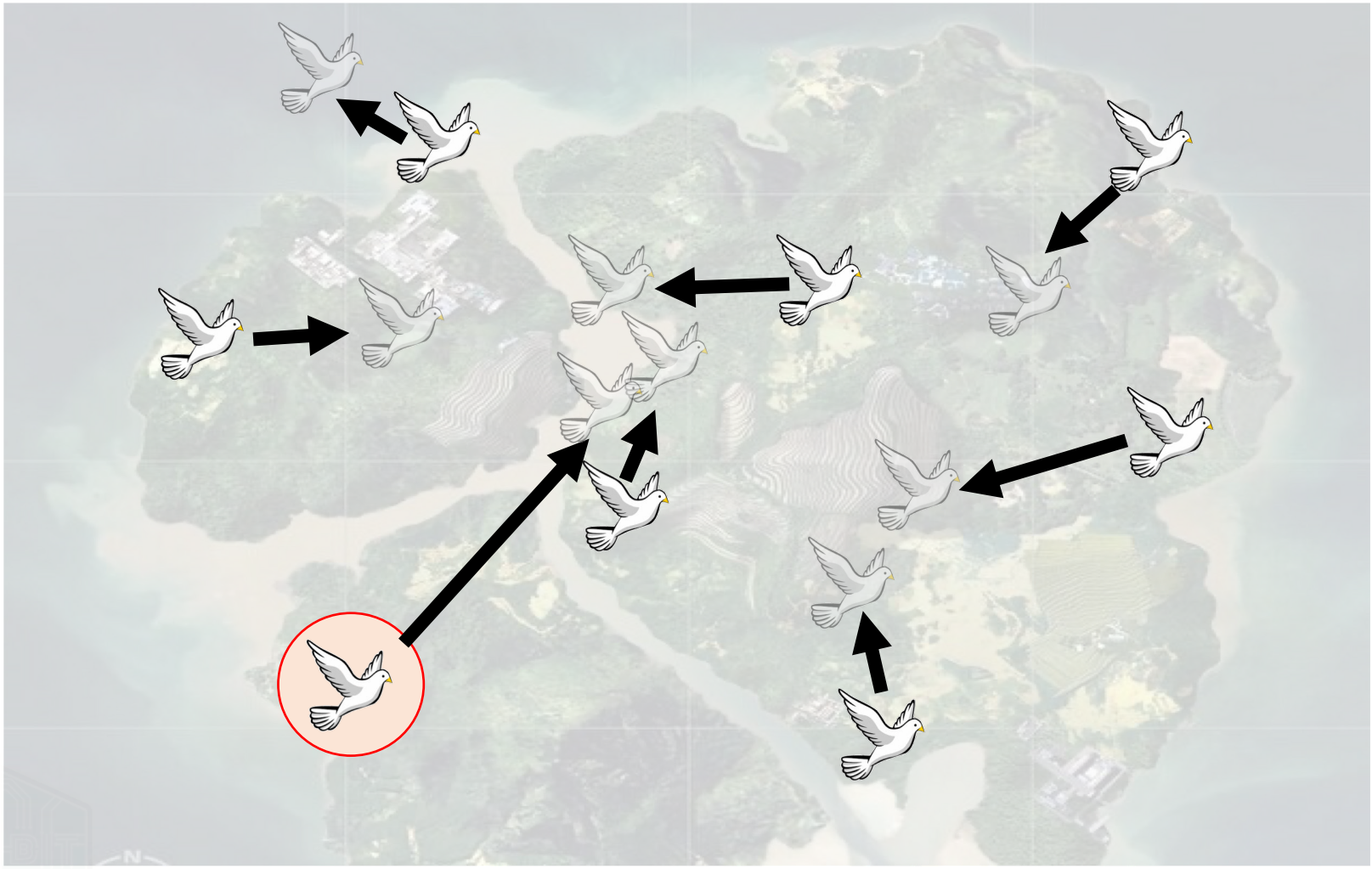
主要思想

二维连续空间找到“食物最多”的地点

主要思想

如何确定每只鸟当前的“速度”？

即下一时刻位置与当前时刻位置的差值（即一个向量）



主要思想

考虑三方面的因素：

（1）上一时刻“速度”

主要思想

考虑三方面的因素：

（1）上一时刻“速度”

（2）该只鸟在历史中找到的最佳取食点的方向

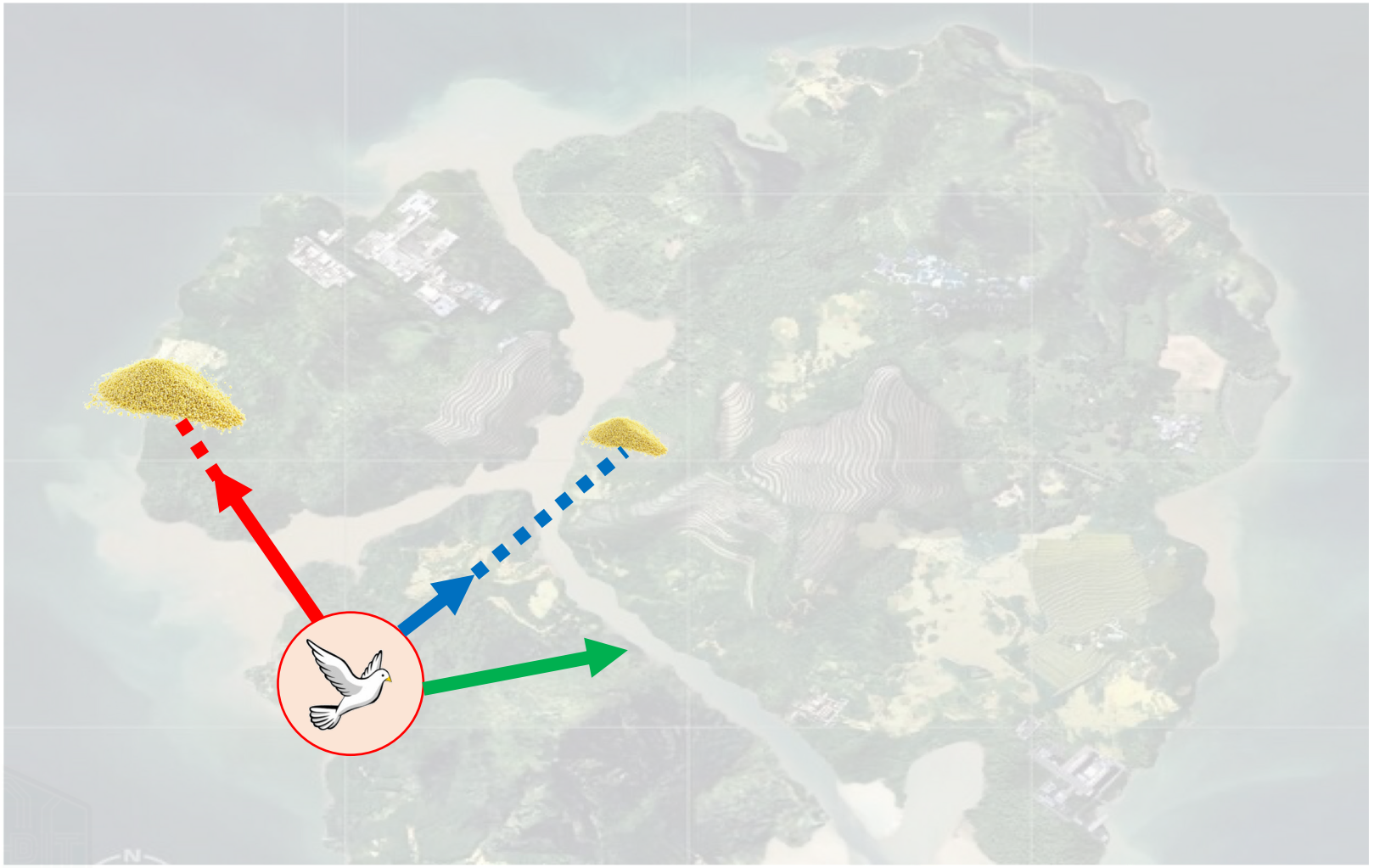
主要思想

考虑三方面的因素：

（1）上一时刻“速度”

（2）该只鸟在历史中找到的最佳取食点的方向

（3）所有鸟在历史中找到的最佳取食点的方向



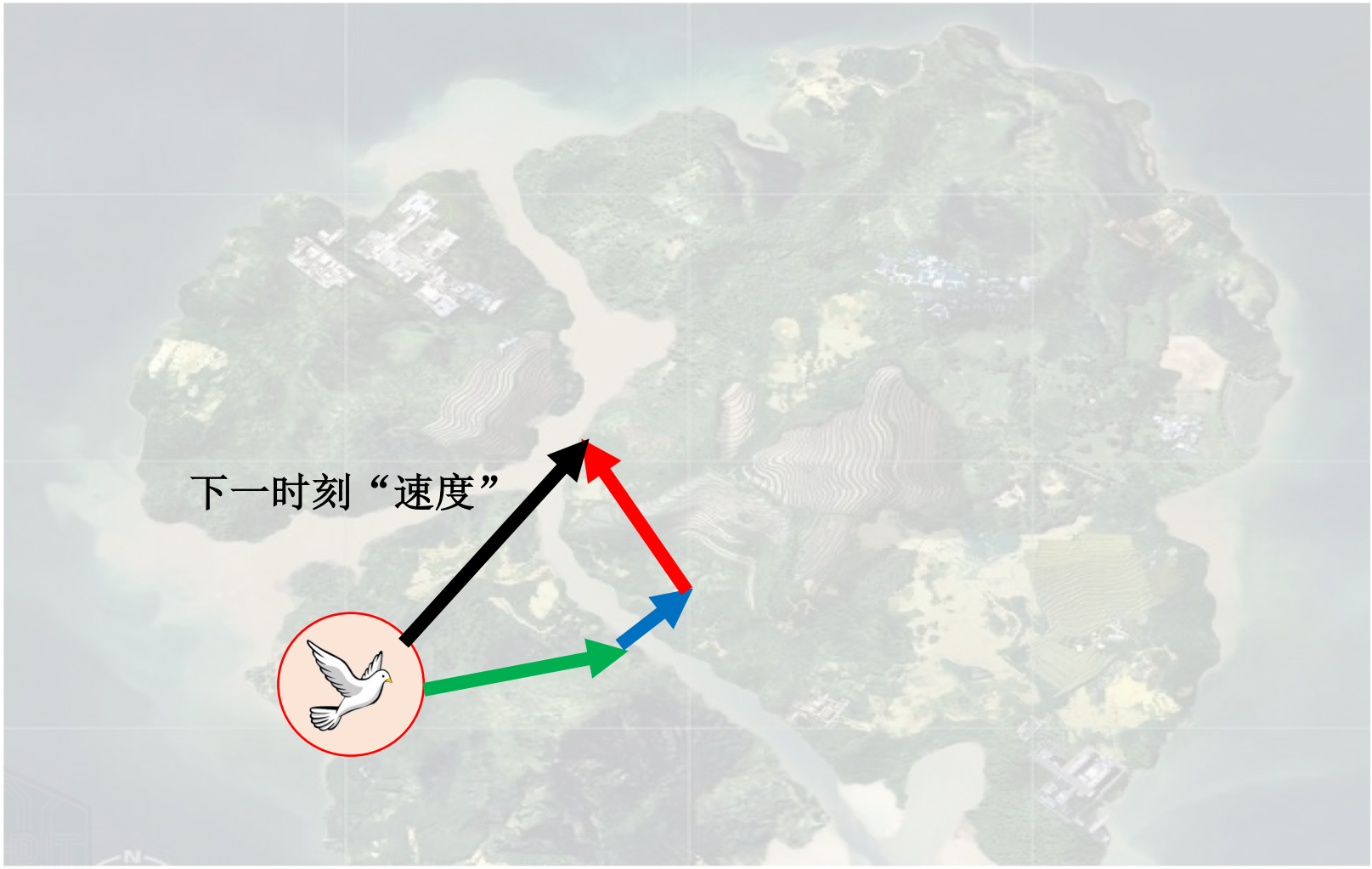
主要思想

考虑三方面的因素：

（1）上一时刻“速度”

（2）该只鸟在历史中找到的最佳取食点的方向

（3）所有鸟在历史中找到的最佳取食点的方向



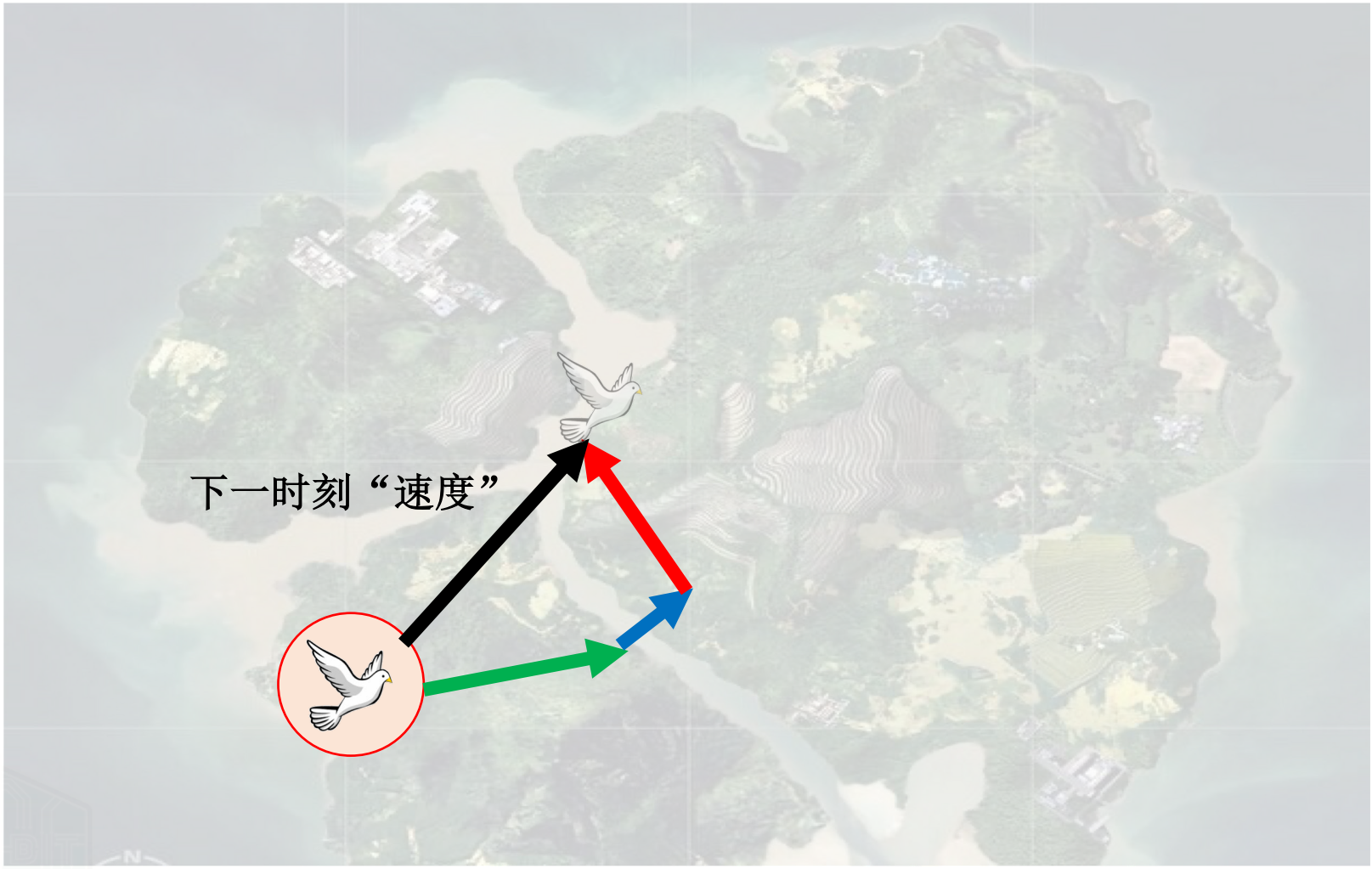
主要思想

考虑三方面的因素：

（1）上一时刻“速度”

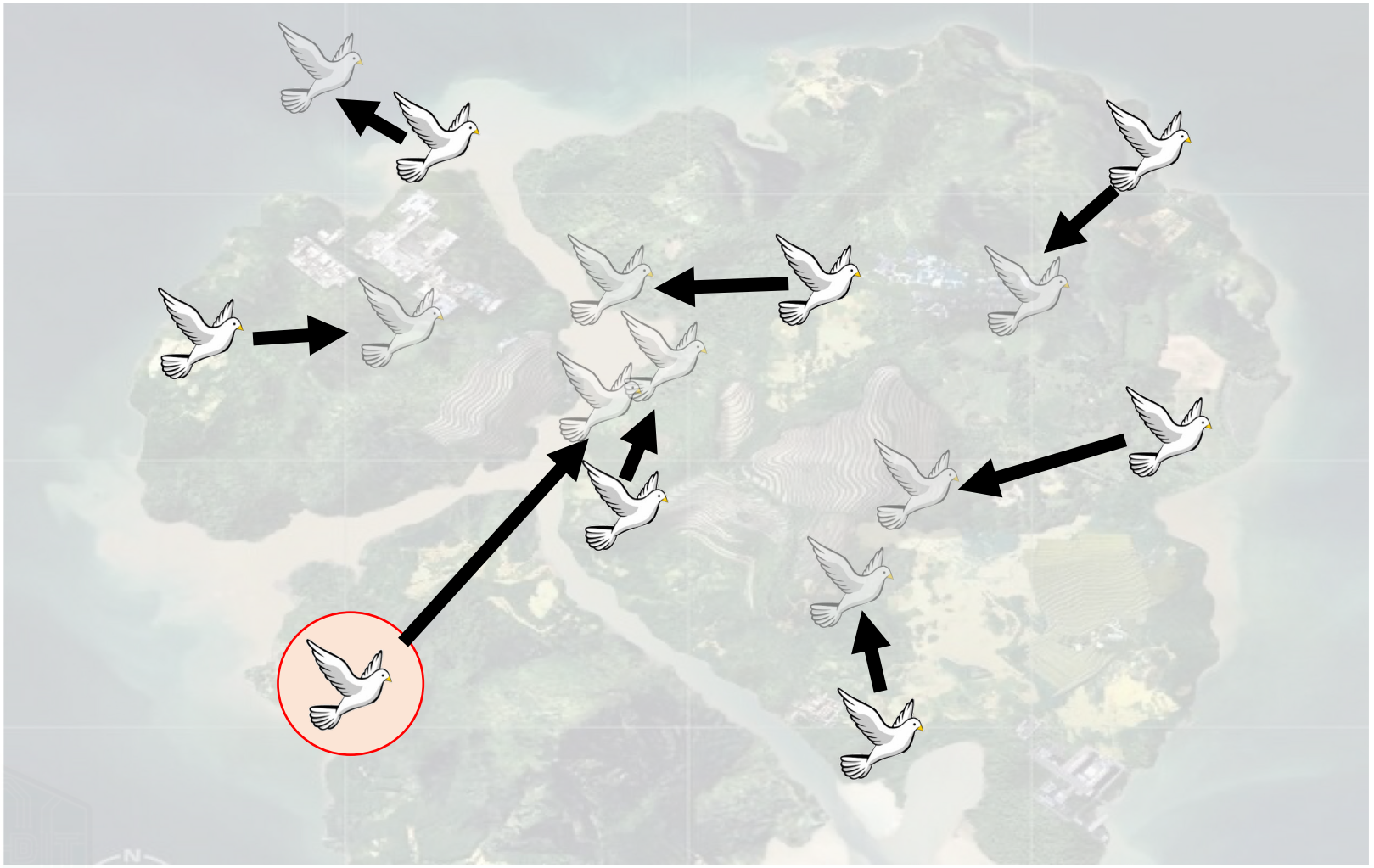
（2）该只鸟在历史中找到的最佳取食点的方向

（3）所有鸟在历史中找到的最佳取食点的方向

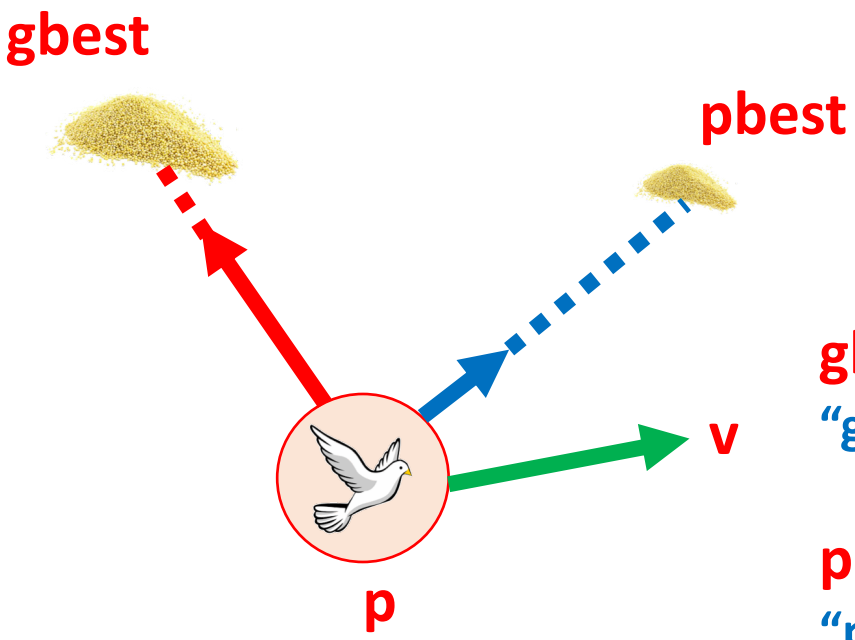


主要思想

所有鸟分别根据三因素计算当前“速度”
根据“速度”移动到下一位置



主要思想



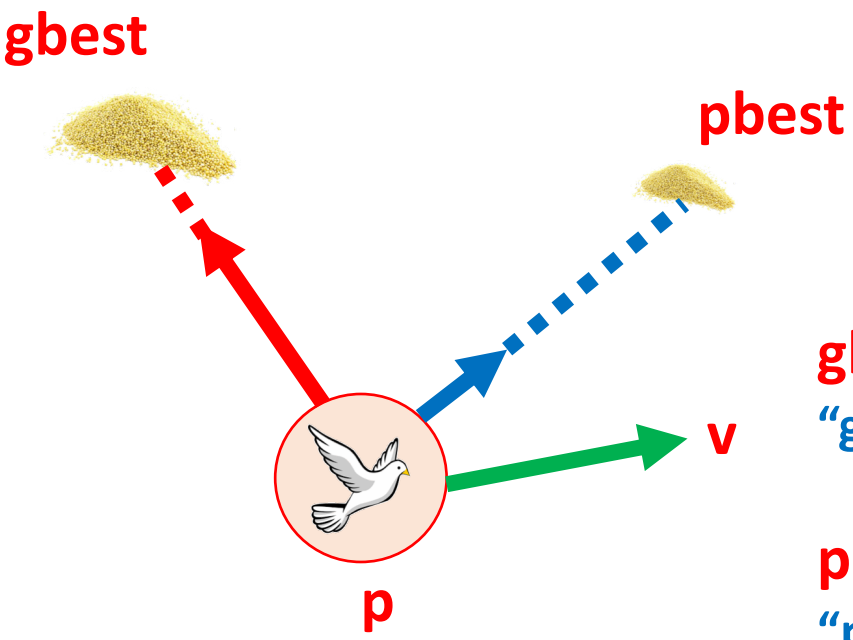
p: 该只鸟当前位置

v: 该只鸟上一时刻速度

gbest: 所有鸟在历史中找到的最佳取食点
“g” – group

pbest: 该只鸟在历史中找到的最佳取食点
“p” – particle (粒子)

主要思想



p: 该只鸟当前位置

v: 该只鸟上一时刻速度

gbest: 所有鸟在历史中找到的最佳取食点
“g” – group

pbest: 该只鸟在历史中找到的最佳取食点
“p” – particle (粒子)

下一时刻速度计算方法：

$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$

粒子群优化

```
[x*] = PSO()
P = Particle_Initialization();
For i=1 to it_max
    For each particle p in P do
        fp = f(p);
        If fp is better than f(pBest)
            pBest = p;
        end
    end
    gBest = best pBest in P;
    For each particle p in P do
        v = w*v + c1*rand*(pBest – p) + c2*rand*(gBest – p);
        p = p + v;
    end
end
```

确定迭代次数

对每只鸟更新**pBest**

根据已更新的**pBest**，更新**gBest**

对每只鸟，更新速度

根据速度，更新位置

最后输出是*gBest*

粒子群优化

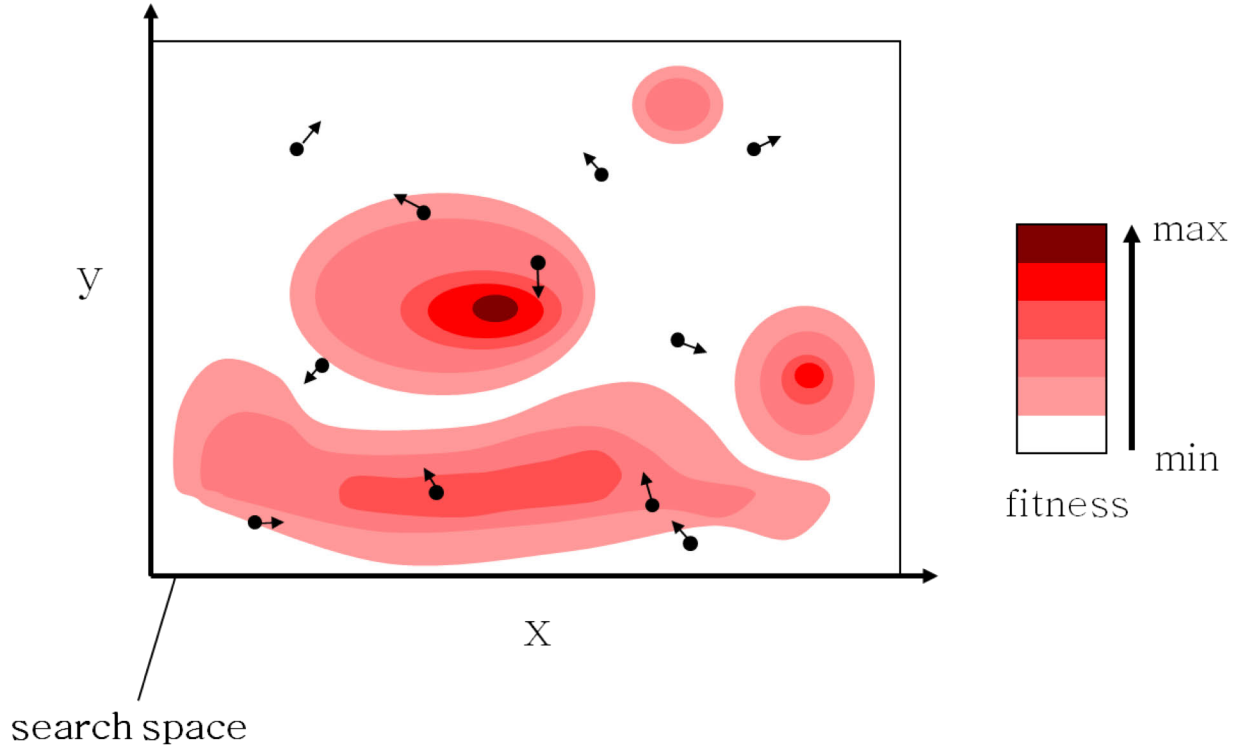
下一时刻速度计算方法：

$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$

1. Inertia
2. Personal Influence
3. Social Influence

思考: 三部分中哪些对应exploration、哪些对应exploitation？

例子



粒子群算法可视化

https://www.bilibili.com/video/BV1Lf4y1v7pt?from=search&seid=12959864526843471690

粒子群优化

- 优势
  - 易于实施
  - 易于并行运行
  - 算法参数少
  - 全局搜索性能好
- 劣势
  - 易于提早收敛到次优解
  - 在局部精确搜索能力差 （可以将PSO与local search结合）

投资组合优化

如何对不同资产进行投资，从而最大化收益的期望、最小化风险（方差）

投资策略记为$\boldsymbol{w} = (w_1, w_2, \ldots, w_N)^T$

The idea of **Markowitz's mean-variance portfolio (MVP)** (Markowitz 1952) is to find a trade-off between the expected return $\mathbf{w}^T\boldsymbol{\mu}$ and the risk of the portfolio measured by the variance $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$:

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \mathbf{w}^T\boldsymbol{\mu} - \lambda\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} \\ \text{subject to} \quad & \mathbf{1}^T\mathbf{w} = 1 \qquad \boxed{\mathbf{w} \geq 0.} \end{aligned}$$

where $\mathbf{w}^T\mathbf{1} = 1$ is the capital budget constraint and λ is a parameter that controls how risk-averse the investor is.

该问题是有约束的凸优化问题

投资组合优化

如何对不同资产进行投资，从而最大化收益的期望、最小化风险（方差）

投资策略记为$\boldsymbol{w} = (w_1, w_2, \ldots, w_N)^T$

给定参数

存在其它建模形式

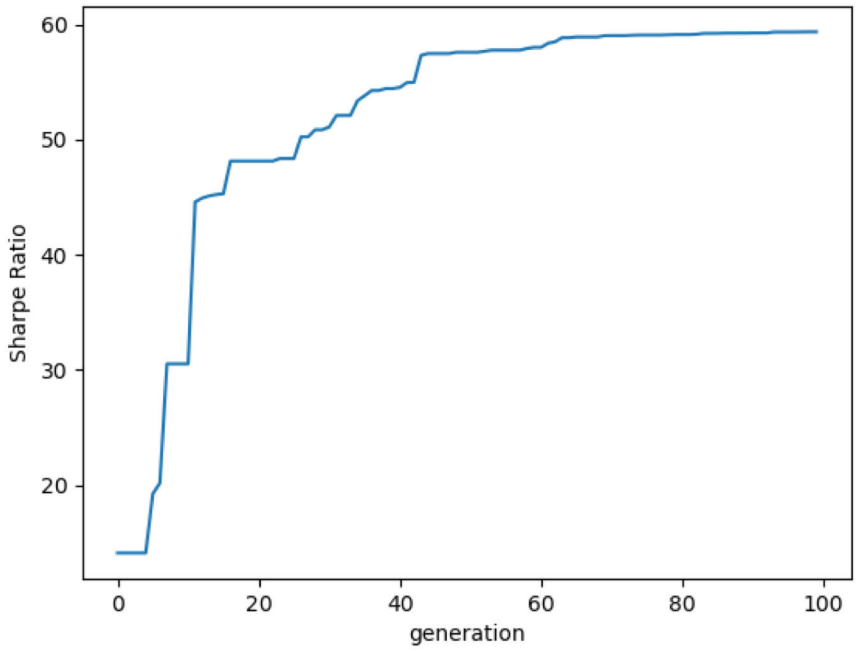
$$\text{maximize } \frac{\boldsymbol{w}^T\boldsymbol{\mu} - R_f}{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}$$

$$\text{subject to } \quad \mathbf{1}^T\mathbf{w} = 1 \quad \mathbf{w} \geq 0.$$

投资组合优化

8组资产，PSO算法代码结果示例：

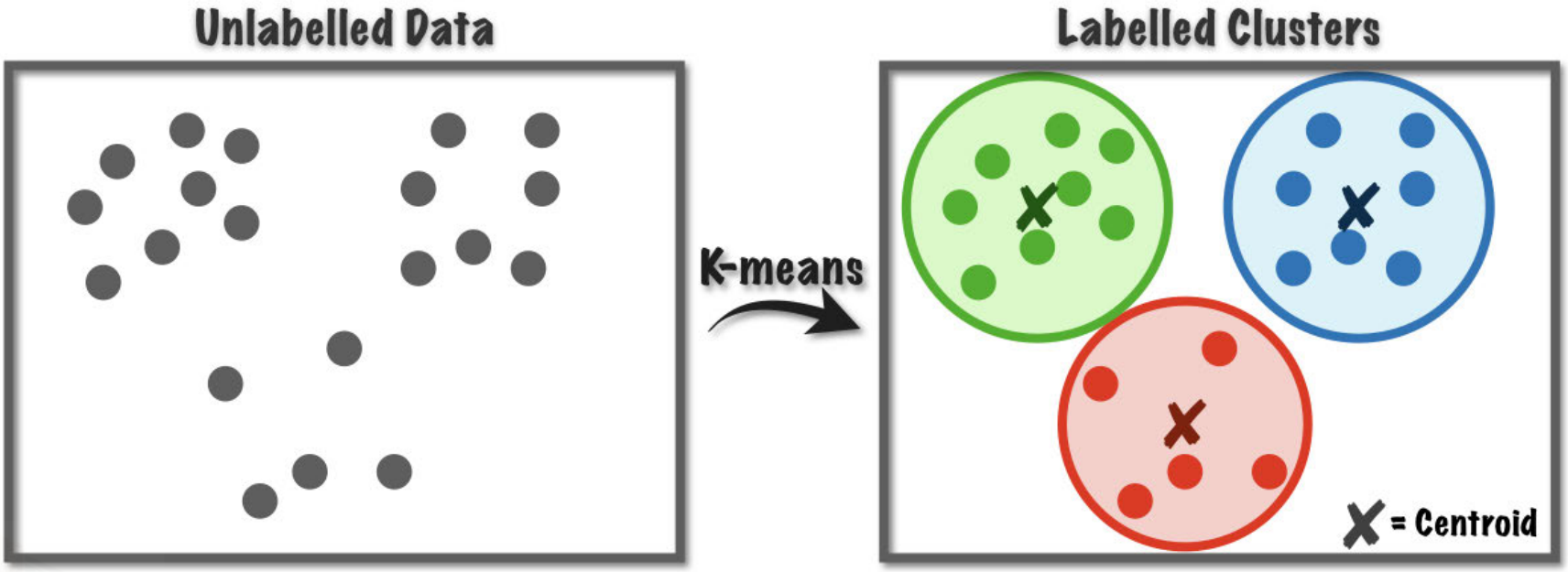
```
[x*] = PSO()
P = Particle_Initialization();
For i=1 to it_max
    For each particle p in P do
        fp = f(p);
        If fp is better than f(pBest)
            pBest = p;
        end
    end
    gBest = best pBest in P;
    For each particle p in P do
        v = w*v + c1*rand*(pBest – p)
                + c2*rand*(gBest – p);
        p = p + v;
    end
end
```



聚类问题

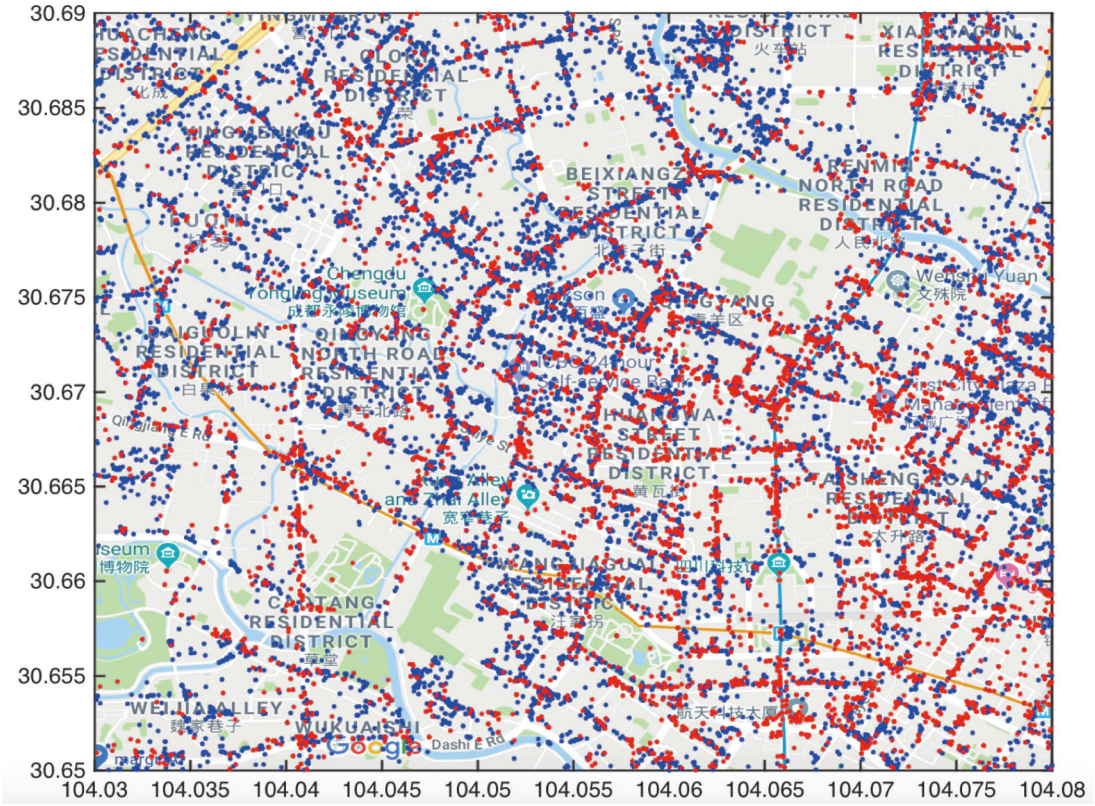
聚类：将给定数据集分成若干子集，确保每个子集中的数据彼此相似

应用场景包括商品分类、顾客分类等



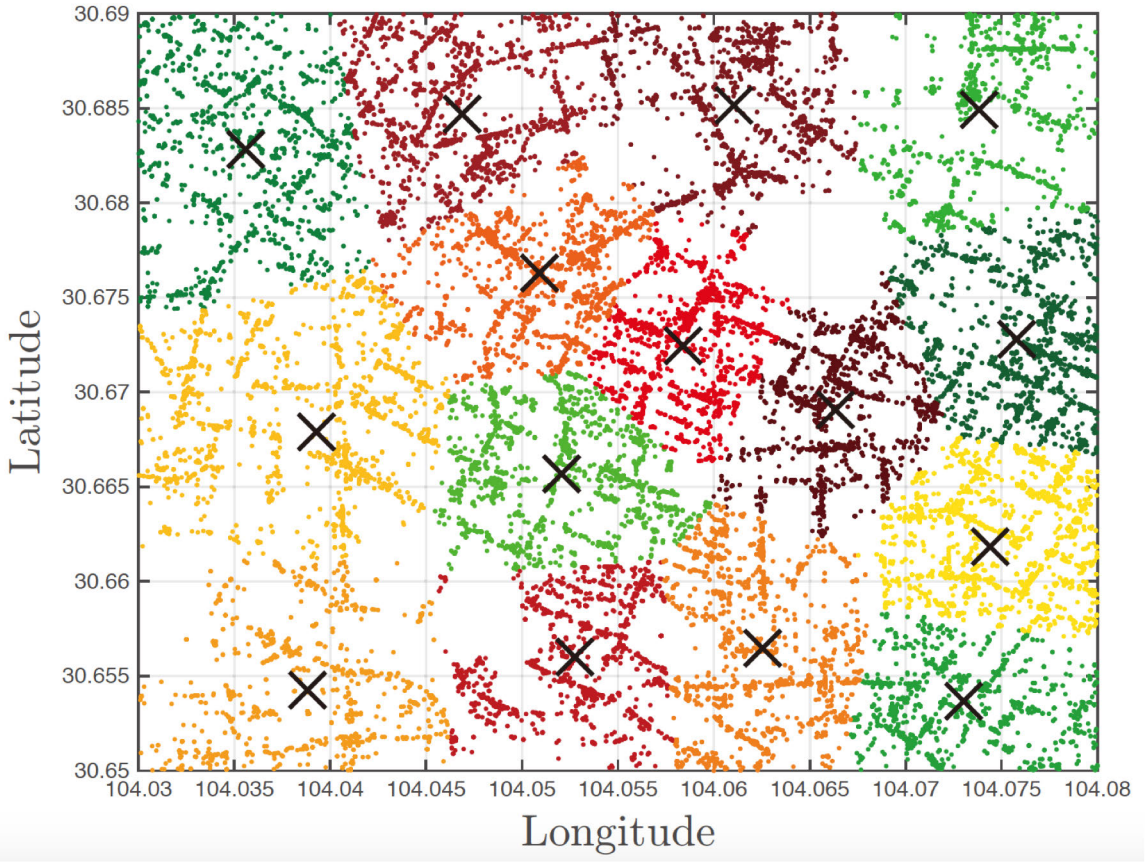
聚类问题

聚类：将给定数据集分成若干子集，确保每个子集中的数据彼此相似



聚类问题

聚类：将给定数据集分成若干子集，确保每个子集中的数据彼此相似



K-均值聚类

聚类：将给定数据集分成若干子集，确保每个子集中的数据彼此相似

K-均值聚类问题（NP-hard）：

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

K是类的数目； x_i 是第j类中的数据点； μ_j是第j类的中心（即类中所有点的平均值）

为什么选择中心作为标准?

K-均值聚类

聚类：将给定数据集分成若干子集，确保每个子集中的数据彼此相似

K-均值聚类问题（NP-hard）：

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

K是类的数目； x_i 是第j类中的数据点； μ_j是第j类的中心（即类中所有点的平均值）

决策变量：是否将x_i分配到集合C_j（第j类）中

K-均值聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

K是类的数目； x_i 是第j类中的数据点； μ_j是第j类的中心（即类中所有点的平均值）

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-均值聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

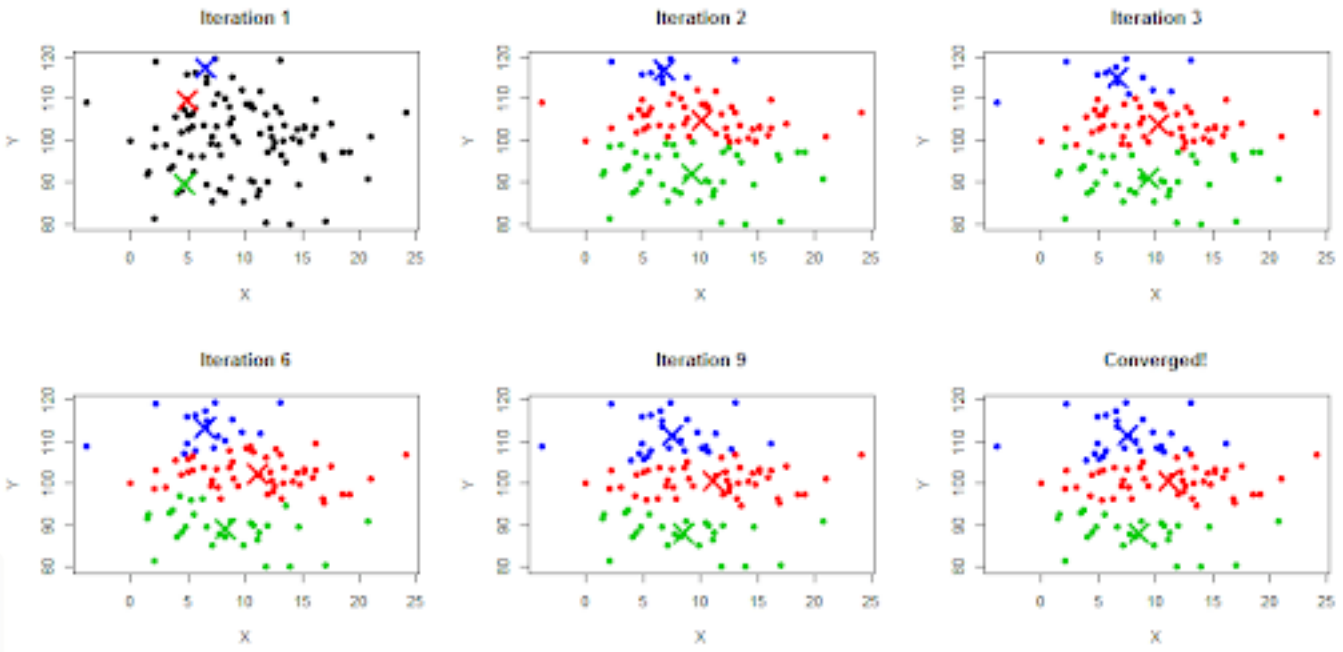
K是类的数目； x_i 是第j类中的数据点； μ_j是第j类的中心（即类中所有点的平均值）

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

在数学上，可将算法理解为：在每次迭代中，先固定$\boldsymbol{\mu}_j$，优化$\boldsymbol{x}_i$，再“滞后”更新$\boldsymbol{\mu}_j$

K-均值聚类

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change



K-均值聚类

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

缺点: 容易陷入局部最优；可能产生不包含任何数据的空类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

可以用PSO求解该问题

PSO聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

```
[x*] = PSO()
P = Particle_Initialization();
For i=1 to it_max
    For each particle p in P do
        fp = f(p);
        If fp is better than f(pBest)
            pBest = p;
        end
    end
    gBest = best pBest in P;
    For each particle p in P do
        v = w*v + c1*rand*(pBest – p) + c2*rand*(gBest – p);
        p = p + v;
    end
end
```

确定迭代次数

对每只鸟更新**pBest**

根据已更新的**pBest**，更新**gBest**

对每只鸟，更新速度

根据速度，更新位置

PSO聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

The PSO Clustering Algorithm pseudo-code as follow:

Initialize each particle with ***K*** random cluster centers. 一个粒子对应？
for iteration count = 1 to maximum iterations **do**
for all particle *i* **do**
for all pattern **X***p* in the dataset **do**
calculate Euclidean distance of **X***p* with all cluster center
assign **X***p* to the cluster that have nearest center to **X***p*
end for
calculate the fitness function *f*.
end for
Find the personal best and global best position of each particle.
Update cluster center according to velocity and coordinate updating formula of PSO.
end for

取自Nasser M. 课件 <Swarm Intelligence>

PSO聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \left\| x_i - \mu_j \right\|^2$$

The PSO Clustering Algorithm pseudo-code as follow:

Initialize each particle with ***K*** random cluster centers.
for iteration count = 1 to maximum iterations **do**
for all particle *i* **do**
for all pattern **X***p* in the dataset **do**
calculate Euclidean distance of **X***p* with all cluster center
assign **X***p* to the cluster that have nearest center to **X***p*
end for
calculate the fitness function *f*.
计算粒子当前找到的解的质量
end for
Find the personal best and global best position of each particle.
Update cluster center according to velocity and coordinate updating formula of PSO.
end for

PSO聚类

$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \left\| x_i - \mu_j \right\|^2$$

The PSO Clustering Algorithm pseudo-code as follow:

Initialize each particle with ***K*** random cluster centers.
for iteration count = 1 to maximum iterations **do**
for all particle *i* **do**
for all pattern **X***p* in the dataset **do**
calculate Euclidean distance of **X***p* with all cluster center
assign **X***p* to the cluster that have nearest center to **X***p*
end for
calculate the fitness function *f*.
计算粒子当前找到的解的质量
end for
Find the personal best and global best position of each particle.
Update cluster center according to velocity and coordinate updating formula of PSO.
初始速度是随机生成
end for

PSO聚类

利用PSO做数据聚类的具体细节可参见以下论文

Data clustering using particle swarm optimization

DW Van der Merwe… - The 2003 Congress on …, 2003 - ieeexplore.ieee.org

… Abstract- This paper proposes two new approaches to **using** PSO to **cluster data**. It is shown … of **clusters**. The **algorithm** is then extended to **use** K-means **clustering** to seed the initial …

☆ Save 99 Cite Cited by 1040 Related articles All 6 versions »

聚类效果对比

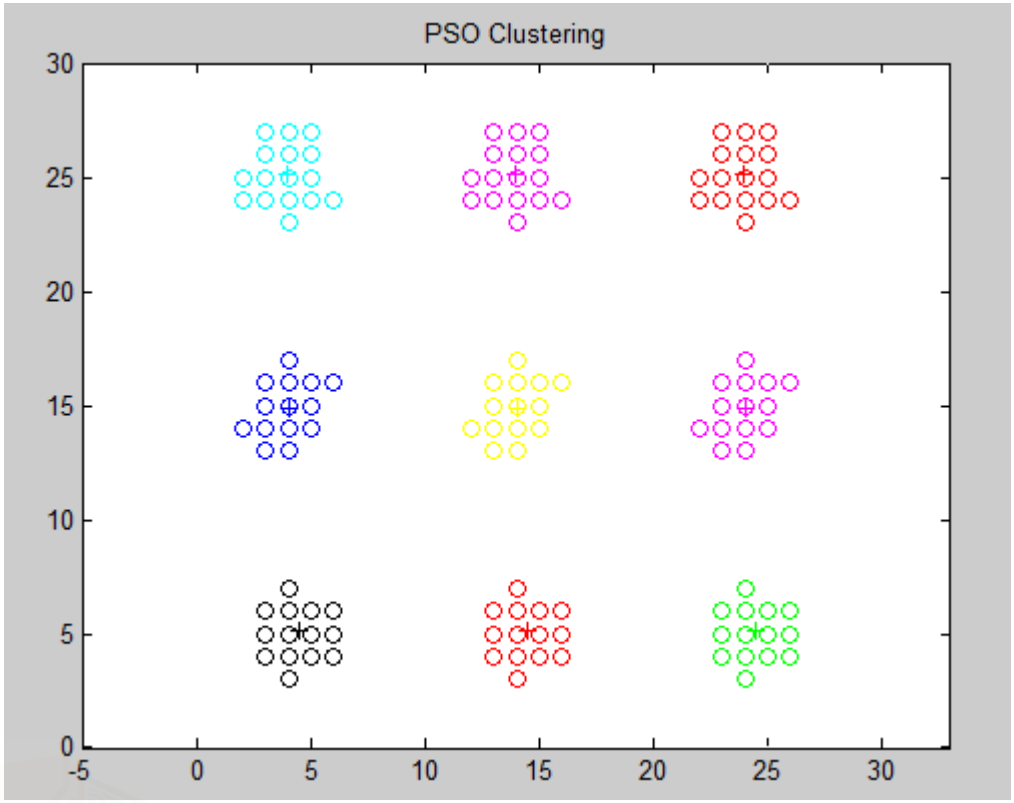
考虑两种数据集，对比基于PSO的聚类算法与K-均值聚类算法

—— 大数据集

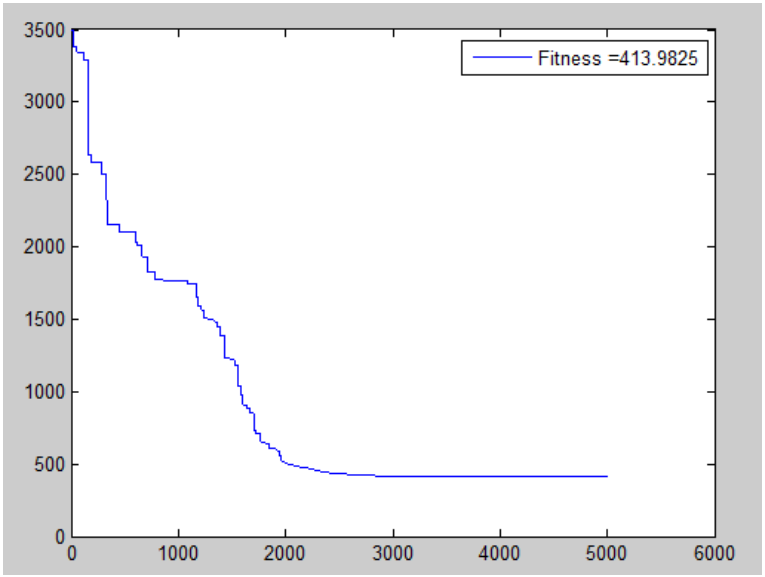
—— 小数据集

聚类效果对比

基于PSO聚类算法



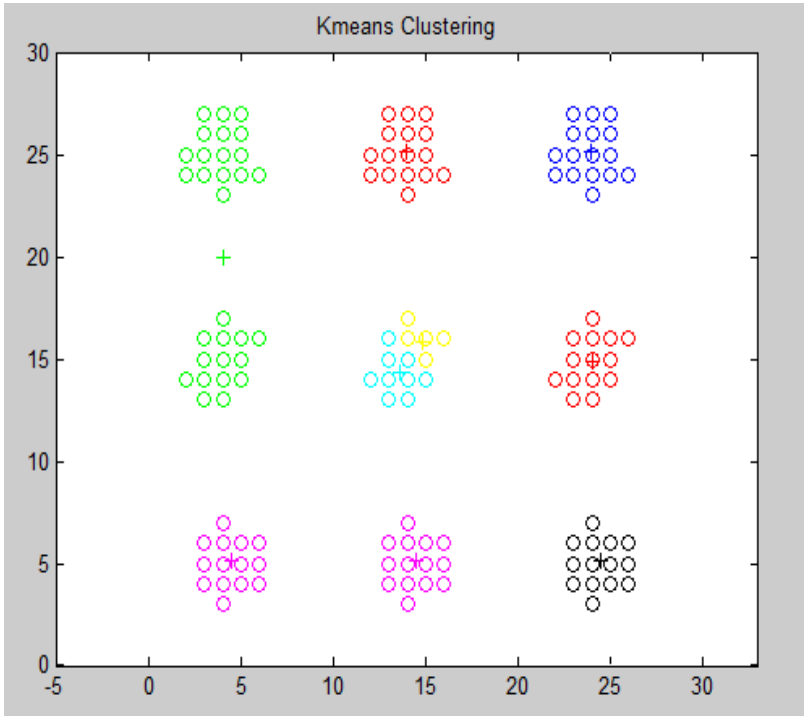
$$\min \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$



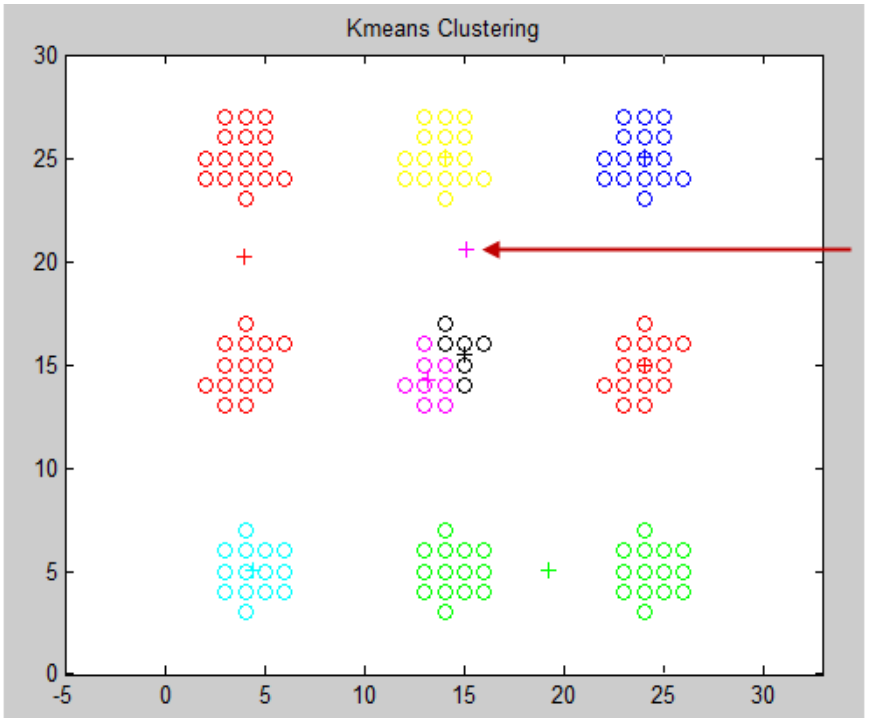
取自Nasser M. 课件 <Swarm Intelligence>

聚类效果对比

K-均值算法



算法收敛于局部最优

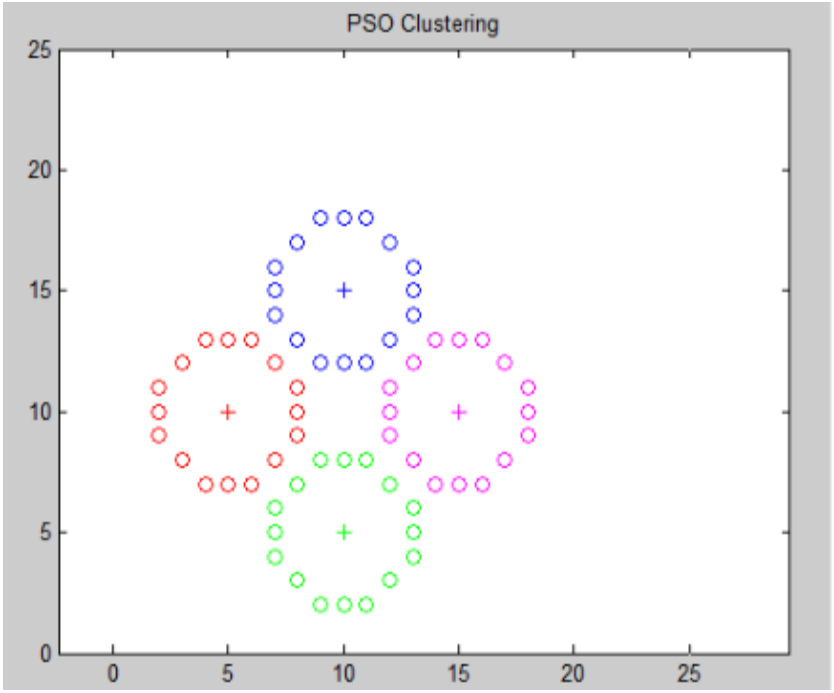


产生不包含数据点的类

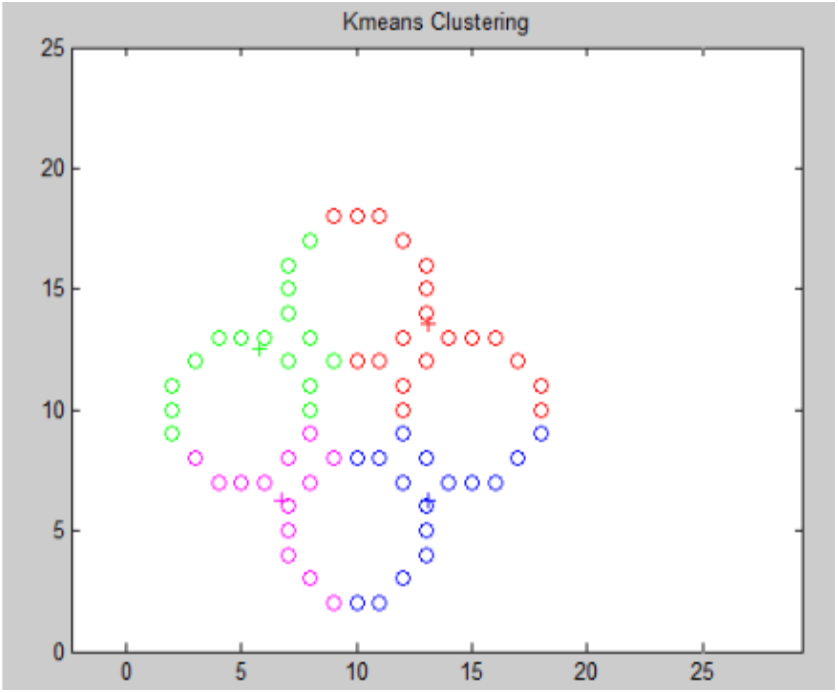
取自Nasser M. 课件 <Swarm Intelligence>

聚类效果对比

基于PSO聚类算法



K-均值算法



取自Nasser M. 课件 <Swarm Intelligence>

聚类效果对比

将基于PSO的聚类算法与K-均值聚类算法合并：

—— K-均值聚类算法有收敛速度快的优点

—— 可以将K-均值聚类算法的结果用于初始化基于PSO的聚类算法

即令一个粒子的初始状态为K-均值聚类算法的输出结果

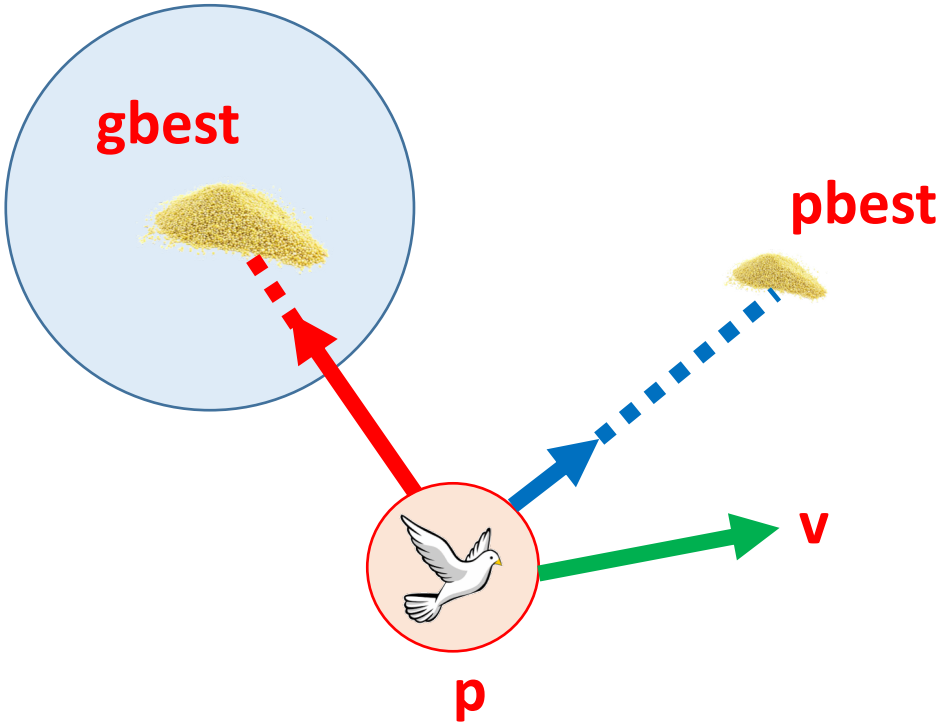
变式之一：由gBest改成lBest

```
[x*] = PSO()
P = Particle_Initialization();
For i=1 to it_max
    For each particle p in P do
        fp = f(p);
        If fp is better than f(pBest)
            pBest = p;
        end
    end
    gBest = best pBest in P;
    For each particle p in P do
        v = w*v + c1*rand*(pBest – p) + c2*rand*(gBest – p);
        p = p + v;
    end
end
```

把gBest替换成每个particle独有的lbest

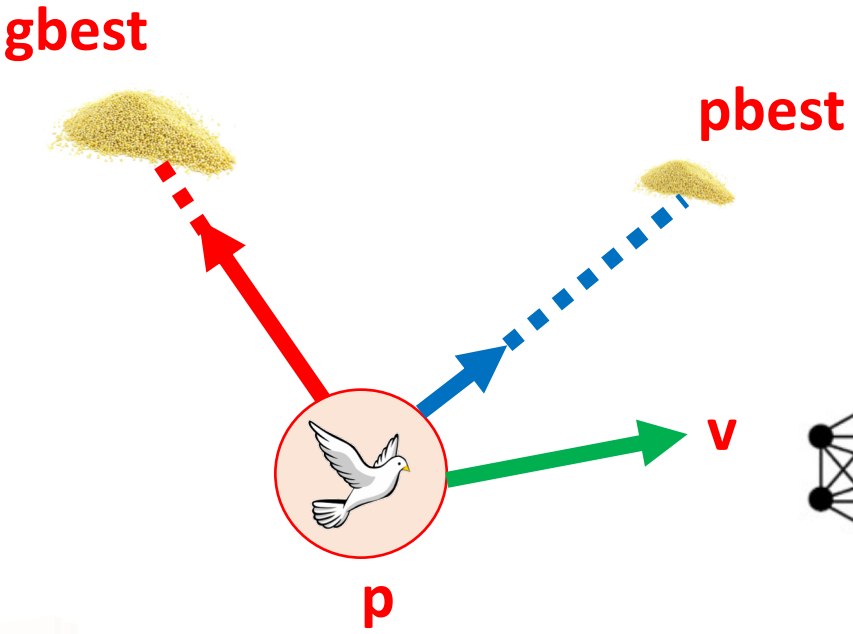
变式之一： 由gBest改成lBest

把gBest替换成每个particle独有的lbest

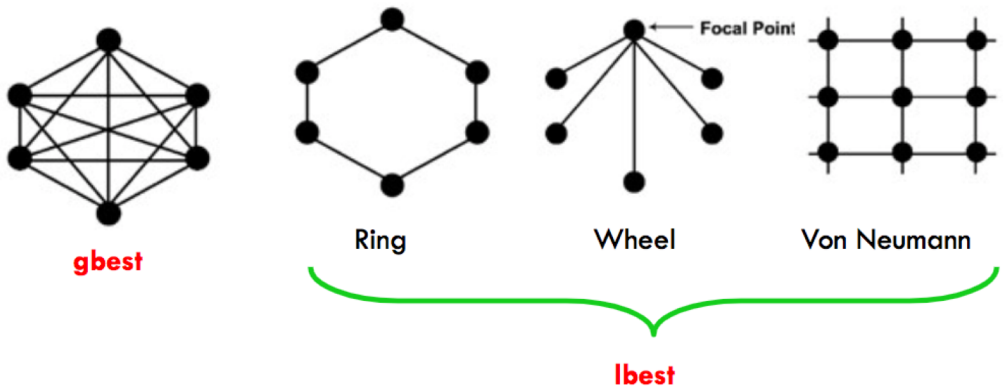


变式之一： 由gBest改成lBest

把gBest替换成每个particle独有的lbest



neighbour的定义：根据解空间远近或者人为定义联系



变式之二：离散变量

```
[x*] = PSO()
P = Particle_Initialization();
For i=1 to it_max
    For each particle p in P do
        fp = f(p);
        If fp is better than f(pBest)
            pBest = p;
        end
    end
    gBest = best pBest in P;
    For each particle p in P do
        v = w*v + c1*rand*(pBest – p) + c2*rand*(gBest – p);
        p = p + v;
    end
end
```

比如求解旅行商问题，

如何定义粒子？

哪些步骤需要调整？

变式之二：离散变量

$$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$$

需要重新定义 +、*、 – 等运算

Discrete particle swarm optimization, illustrated by the traveling salesman problem

M Clerc - New optimization techniques in engineering, 2004 - Springer

Abstract The classical Particle Swarm Optimization is a powerful method to find the minimum of a numerical function, on a continuous definition domain. As some binary versions have already successfully been used, it seems quite natural to try to define a framework for a discrete PSO. In order to better understand both the power and the limits of this approach, we examine in detail how it can be used to solve the well known Traveling Salesman Problem, which is in principle very “bad” for this kind of optimization heuristic. Results show …

☆ 99 Cited by 631 Related articles All 6 versions

变式之二：离散变量

$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$

需要重新定义 +、*、 – 等运算

重新定义的加法运算: 粒子位置 加 速度

Example

$$\begin{cases} p = (1,2,3,4,5,1) \\ v = ((1,2),(2,3)) \end{cases}$$

Applying v to x, we obtain successively

$(2,1,3,4,5,2)$

$(3,1,2,4,5,3)$

变式之二：离散变量

$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$

需要重新定义 +、*、 – 等运算

重新定义的加法运算: 速度 加 速度

Let v_1 and v_2 be two velocities. In order to compute $v_1 \oplus v_2$ we consider the list of transpositions which contains first the ones of v_1, followed by the ones of v_2.

变式之二：离散变量

$$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$$

需要重新定义 +、*、 – 等运算

重新定义的减法运算: 位置 减 位置

Let x_1 and x_2 be two positions. The difference $x_2 - x_1$ is defined as the velocity v, found by a given algorithm, so that applying v to x_1 gives x_2.

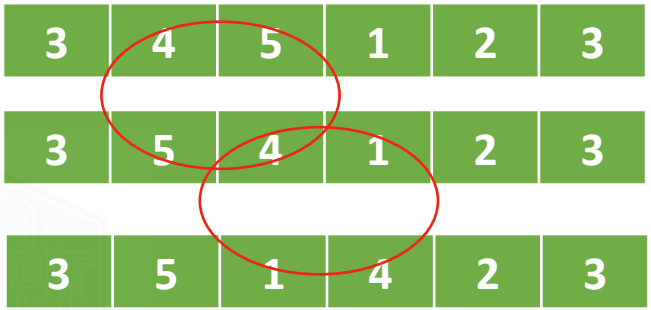
变式之二：离散变量

$$v \leftarrow w*v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$$

需要重新定义 +、*、 – 等运算

重新定义的减法运算: 位置 减 位置

p　　pBest



故pBest-p为((4, 5), (4, 1))

群体智能之人工蜂群算法

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
局部搜索、模拟退火、遗传算法、差分进化算法、蚁群算法、粒子群优化算法、人工蜂群算法
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

人工蜂群算法

人工蜂群（Artificial Bee Colony，ABC）

On the performance of **artificial bee colony** (ABC) algorithm
D Karaboga, B Basturk - Applied soft computing, 2008 - Elsevier
Artificial bee colony (ABC) algorithm is an optimization algorithm based on a particular intelligent behaviour of honeybee swarms. This work compares the performance of ABC algorithm with that of differential evolution (DE), particle swarm optimization (PSO) and …
☆ 99 Cited by 3731 Related articles All 7 versions

A comparative study of **artificial bee colony** algorithm
D Karaboga, B Akay - Applied mathematics and computation, 2009 - Elsevier
Abstract **Artificial Bee Colony** (ABC) algorithm is one of the most recently introduced swarm-based algorithms. ABC simulates the intelligent foraging behaviour of a honeybee swarm. In this work, ABC is used for optimizing a large set of numerical test functions and the results …
☆ 99 Cited by 3231 Related articles All 14 versions

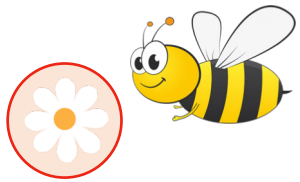
A modified **artificial bee colony** algorithm
W Gao, S Liu - Computers & Operations Research, 2012 - Elsevier
Artificial bee colony algorithm (ABC) is a relatively new optimization technique which has been shown to be competitive to other population-based algorithms. However, there is still an insufficiency in ABC regarding its solution search equation, which is good at exploration …
☆ 99 Cited by 616 Related articles All 6 versions

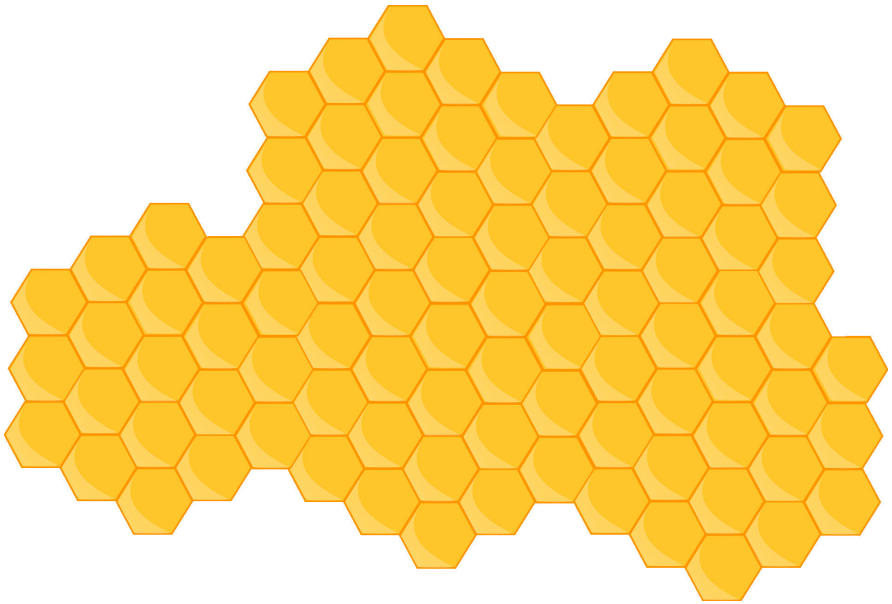
A comprehensive survey: **artificial bee colony** (ABC) algorithm and applications
D Karaboga, B Gorkemli, C Ozturk… - **Artificial** Intelligence …, 2014 - Springer
Swarm intelligence (SI) is briefly defined as the collective behaviour of decentralized and self-organized swarms. The well known examples for these swarms are bird flocks, fish schools and the **colony** of social insects such as termites, ants and bees. In 1990s …
☆ 99 Cited by 1504 Related articles All 11 versions

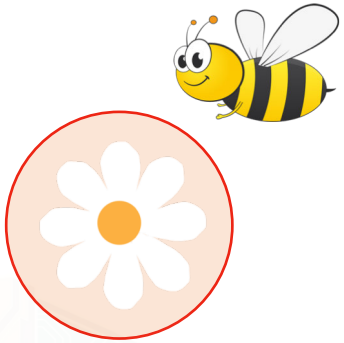
人工蜂群算法

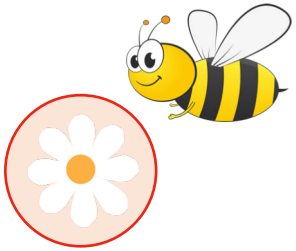
第一种蜂: 雇佣蜂 (employed bee)

模拟蜜蜂采蜜过程





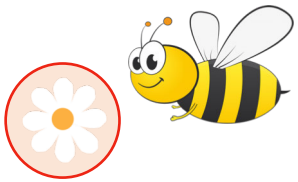


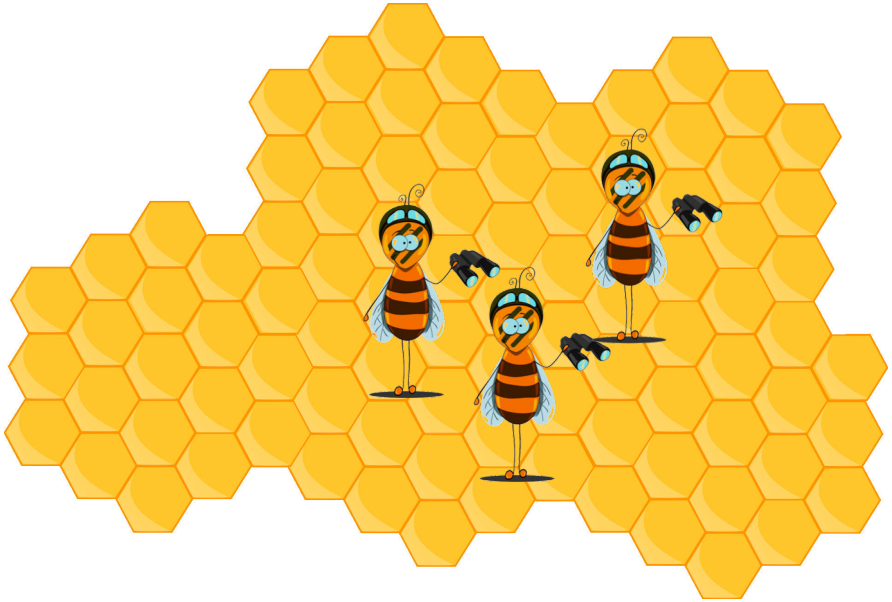


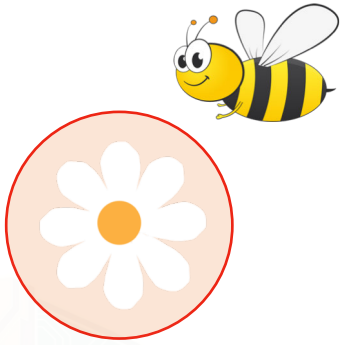
人工蜂群算法

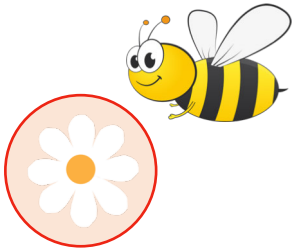
第二种蜂: 观望蜂 (onlooker bee)

模拟蜜蜂采蜜过程





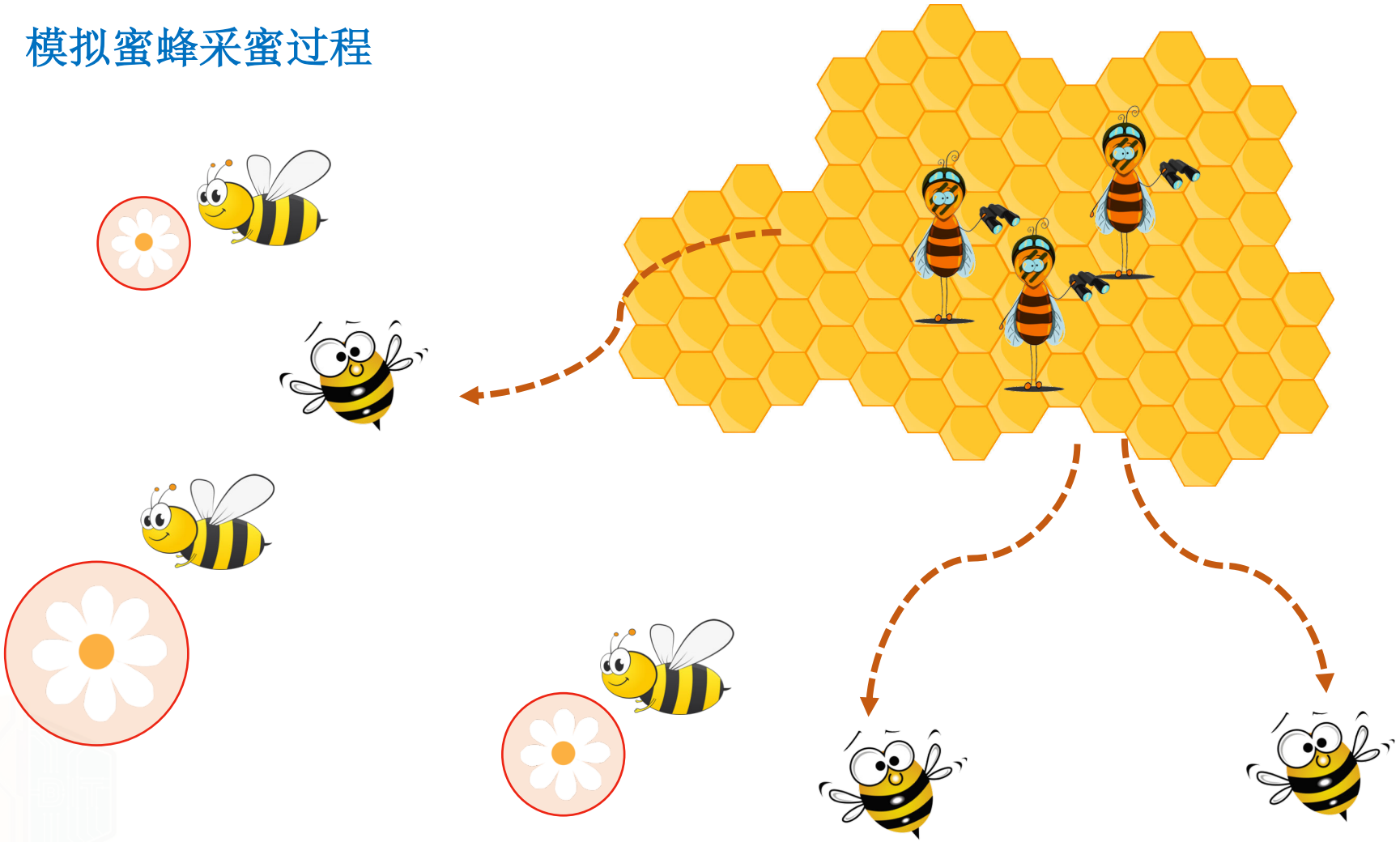




人工蜂群算法

第三种蜂: 侦察蜂 (scout bee)

模拟蜜蜂采蜜过程



人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

(6) Calculate the probability values P_i for the solution (x_i) by (3)

雇佣蜂传递信息

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}
(6) Calculate the probability values P_i for the solution (x_i) by (3)
(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}
(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)
(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

雇佣蜂

人工蜂群算法

(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

$$v_{ij} = \begin{cases} x_{ij} + \text{rand} \cdot \left(x_{ij} - x_{kj} \right), & j = j_{\text{rand}}, \\ x_{ij}, & j \neq j_{\text{rand}}. \end{cases}$$

人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

(6) Calculate the probability values P_i for the solution (x_i) by (3)

雇佣蜂传递信息

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

人工蜂群算法

(6) Calculate the probability values P_i for the solution (x_i) by (3) 雇佣蜂传递信息

$$P_i = \frac{f_i}{\sum_{k=1}^{SN} f_k}$$

人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

(6) Calculate the probability values P_i for the solution (x_i) by (3)

雇佣蜂传递信息

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

人工蜂群算法

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

$$P_i = \frac{f_i}{\sum_{k=1}^{SN} f_k}$$

$$v_{ij} = \begin{cases} x_{ij} + \text{rand} \cdot \left(x_{ij} - x_{kj} \right), & j = j_{\text{rand}}, \\ x_{ij}, & j \neq j_{\text{rand}}. \end{cases}$$

对于最大化问题，直接根据每个解的目标函数值，按比例计算概率

对于最小化问题，需要先根据目标函数值进行变换，然后再计算概率（见后面例子）

人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

(6) Calculate the probability values P_i for the solution (x_i) by (3)

雇佣蜂传递信息

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

人工蜂群算法

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

如果某一个解经过多轮一直没有改进，则随机产生新解将其代替

该过程可能把质量高的解替换掉

人工蜂群算法

(1) Generate the initial population x_i $(i = 1, 2, \ldots, SN)$
(2) Evaluate the fitness (fit(x_i)) of the population
(3) Set cycle to 1
(4) Repeat
(5) For each employed bee {
Produce new solution v_i by using (2)
Calculate its fitness value fit(v_i)
Apply greedy selection process}

雇佣蜂

(6) Calculate the probability values P_i for the solution (x_i) by (3)

雇佣蜂传递信息

(7) For each onlooker bee {
Select a solution x_i depending on P_i
Produce new solution v_j
Calculate its fitness value fit(v_j)
Apply greedy selection process}

观望蜂

(8) **If** there is an abandoned solution for the scout,
then replace it with a new solution which will be randomly produced by (4)

侦察蜂

(9) Memorize the best solution so far
(10) Cycle = cycle +1
(11) **Until** cycle = MEN

本讲小结

- 群体智能之蚁群优化算法
- 群体智能之粒子群优化算法
- 群体智能之人工蜂群算法

计算机学院
余皓然

主要参考资料

B站UP主 FancyZerOkami <蚂蚁模拟 Colony simulation> Video

Jeff Lettman <An Introduction to Ant colony optimization> Video

Joy Dutta <Ant colony optimization> Slides

C. Blum <Ant colony optimization: Introduction and recent trends> Paper

Krzysztof Schiff <Ant colony optimization algorithm for the 0-1 knapsack problem> Paper

Prakash Kotecha (IIT Guwahati) <Computer Aided Applied Single Objective Optimization> Slides

B站UP主 幼鹰me <粒子群算法可视化> Video

Nasser M. <Swarm Intelligence> Slides

计算机学院
余皓然

2023/4/23

谢谢！

计算机学院
余皓然
2023/4/23