上节回顾

第七章 遗传算法

- 生物理论基础
- 算法基本框架
- 编码/解码
- 适应度函数
- 遗传算子（选择、交叉、变异）
- 停止条件

第八章　群智能

三个臭皮匠顶个诸葛亮。

众人拾柴火焰高。

群智能概述

- **群智能（ Swarm Intelligence, SI ）**

  群（swarm）：某种交互作用的组织或agent的结构集合。

  对于群居昆虫，如蚂蚁、蜜蜂、鱼群、鸟群等，个体在结构上是很简单的，而它们的集体行为却可能变得相当复杂。

  人们把群居昆虫的集体行为称作"群智能"，即低智能的主体通过合作表现出高智能行为的特性。

群智能算法是一种基于生物群体行为规律的计算技术。

- **特点**

1. 个体的行为很简单，但当它们一起协同工作时却能够突现出非常复杂（智能）的行为特征。

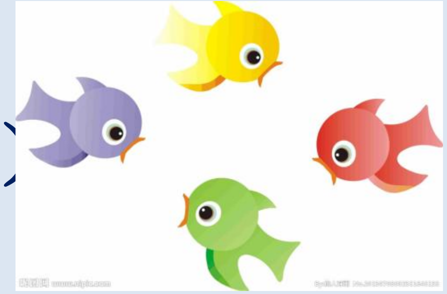
2. 群智能优化在没有集中控制且不提供全局模型的前提下，为寻找复杂的分布式问题求解方案提供了基础。

- **优点**

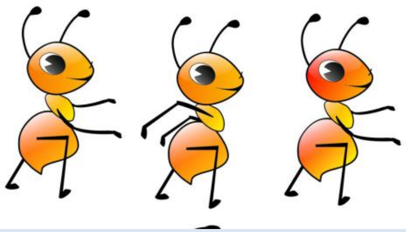
1. 灵活性：群体可以适应随时变化的环境；
2. 稳健性：个体失败，群体仍能完成任务；
3. 自组织：活动既不受中央控制，也不受局部监管。

- **典型算法**

粒子群优化算法（鸟群捕食）



蚁群算法（蚂蚁觅食）



粒子群优化算法

- **粒子群优化算法简述**

粒子群优化算法（Particle Swarm Optimization，PSO），也称为粒子群算法，是近几年来发展起来的一种新的群体搜索算法。

和遗传算法相似，它也是从随机的解出发，通过迭代寻找最优解，通过适应度来评价解的品质。

比遗传算法规则更为简单，它没有遗传算法的“交叉”(Crossover) 和“变异”(Mutation) 操作，而是追随当前搜索到的最优值来寻找全局最优。

- ## 粒子群算法的思想和起源

- 由James Kenney（社会心理学博士）和Russ Eberhart（电子工程学博士），于1995年提出。

- 该算法源于对鸟群捕食行为的研究，是受到飞鸟集群活动的规律性启发，利用群体智能建立的一个简化模型。

• 粒子群算法的原理描述（1）

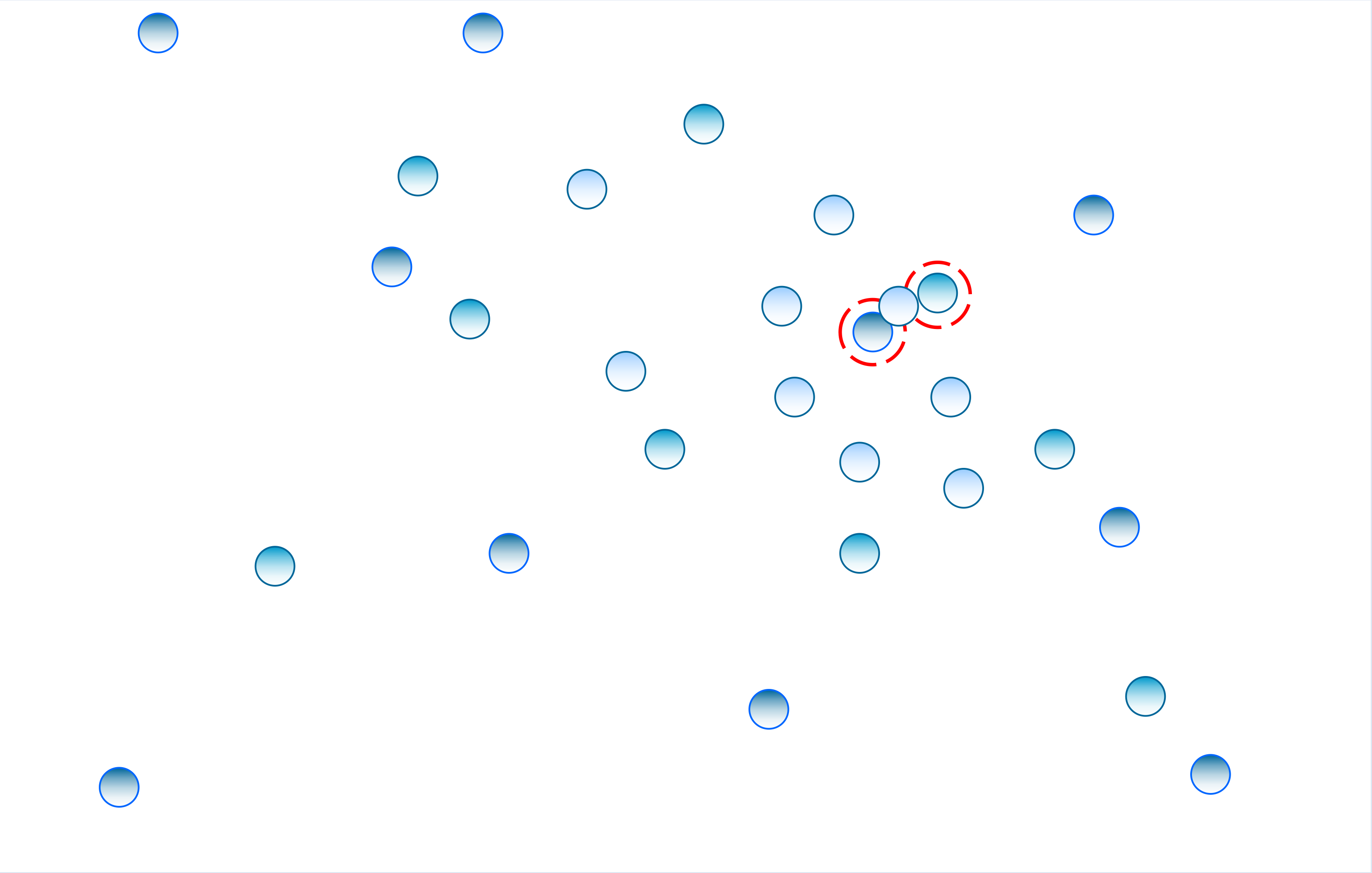
假设存在一个区域，所有的鸟都不知道食物的位置，但是它们知道当前的位置离食物还有多远。找到食物的最优策略是什么呢？搜寻目前离食物最近的鸟的周围区域。

在该算法中，每个解看作一只鸟，称为粒子(particle)，所有的粒子都有一个适应值，每个粒子都有一个速度决定它们的飞翔方向和距离，粒子们追随当前最优粒子在解空间中搜索。

- ## 粒子群算法的原理描述（2）

当其它鸟发现了更佳的觅食地点，鸟群间 会有某种类似广播的沟通行为，渐渐的将其它鸟群引领至较佳的地点。这样的觅食行为是利 用社会中所存在的互相影响的概念，来引领所有个体朝向最佳解位置。

- **粒子群寻优示意图**



• 基本粒子群算法的描述

1. 假设在D维搜索空间中，有m个粒子；

（1）其中第i个粒子的位置矢量表示为：

$$\vec{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$$

（2）飞翔速度矢量表示为：

$$\vec{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$$

（3）第 i个粒子搜索到的最优位置为：

$$\vec{p}_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$$

（4）整个粒子群搜索到的最优位置为：

$$\vec{p}_{gbest} = (p_{gbest1}, p_{gbest2}, \cdots, p_{gbestD})$$

- **基本粒子群算法的描述**

2. 粒子速度和位置的更新

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m; \quad d = 1,2,\cdots,D$$

其中w为惯性权重，

d=1,2, …, D，i=1,2, …, M。

c1 和c2 为两个正常数称为加速因子，rand( ) 为分布于[0,1]的随机数

- # 基本粒子群算法的描述

2. 粒子速度和位置的更新

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

$$i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

“惯性部分”，对自身运动状态的信任

“认知部分”，对微粒本身的思考，即来源于自己经验的部分

“社会部分”，微粒间的信息共享，来源于群体中的其它优秀微粒的经验

• 基本粒子群算法的描述

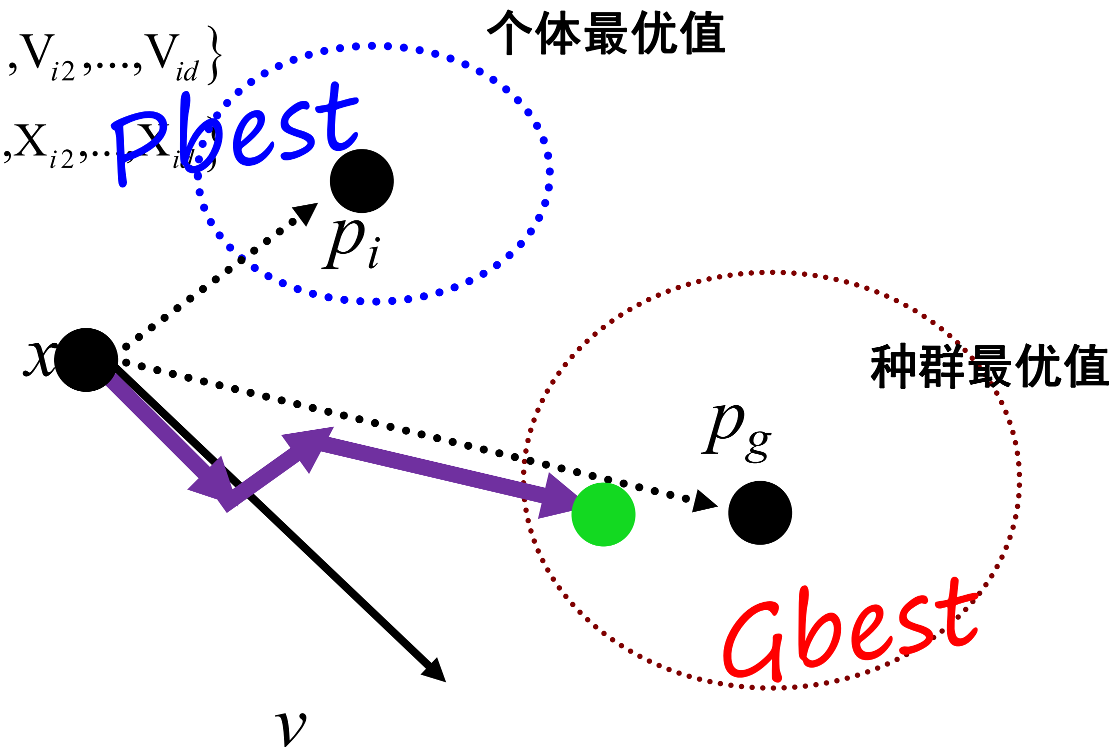
3. 粒子更新示意图

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot rand() \cdot (p_{id} - x_{id}(t)) + c_2 \cdot rand() \cdot (p_{gd} - x_{id}(t))$$

$$x_i(t+1) = x_i(t) + v_i(t)$$

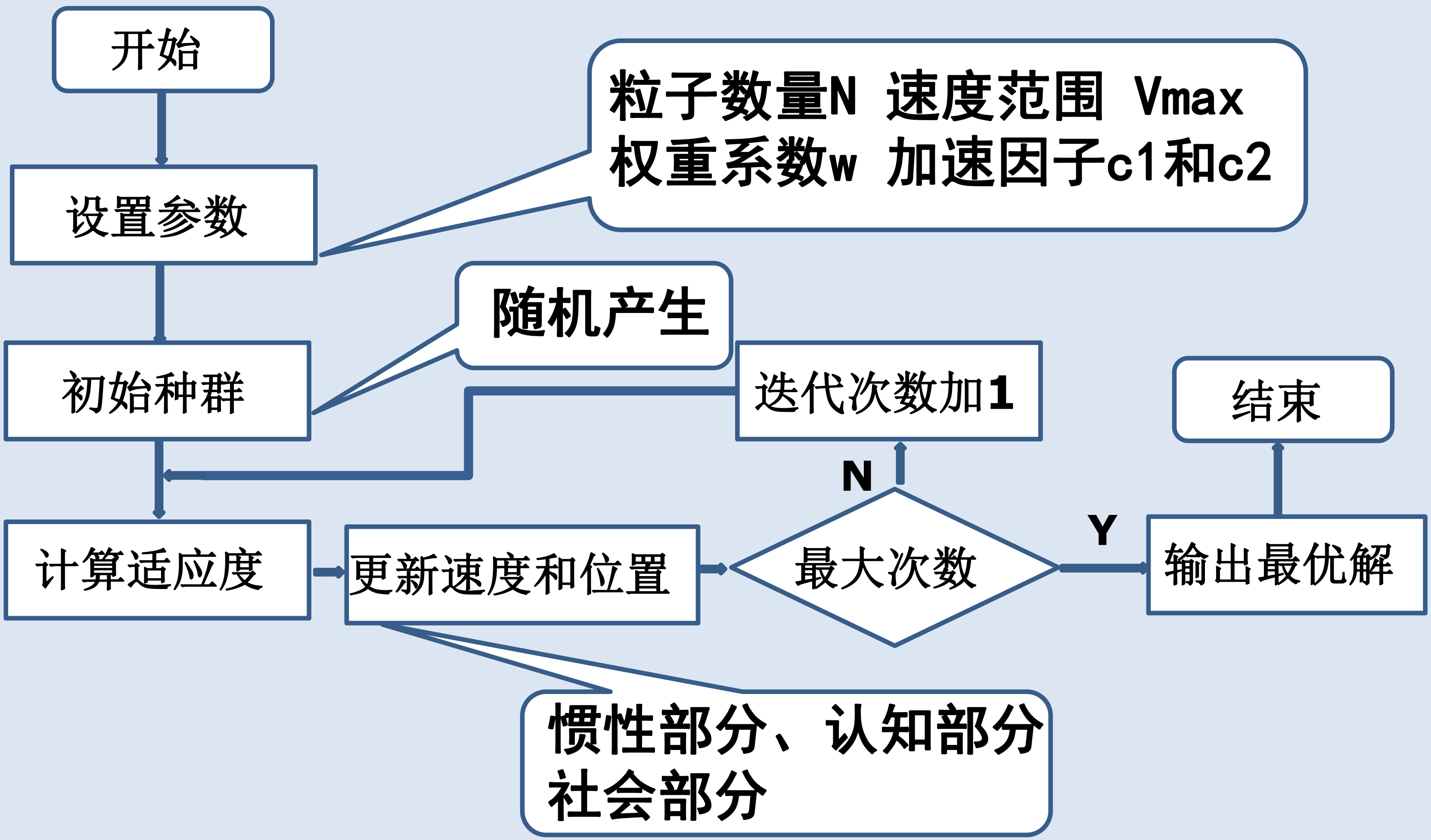
$$\mathrm{V}_i = \{\mathrm{V}_{i1}, \mathrm{V}_{i2}, \ldots, \mathrm{V}_{id}\}$$

$$\mathrm{X}_i = \{\mathrm{X}_{i1}, \mathrm{X}_{i2}, \ldots, \mathrm{X}_{id}\}$$



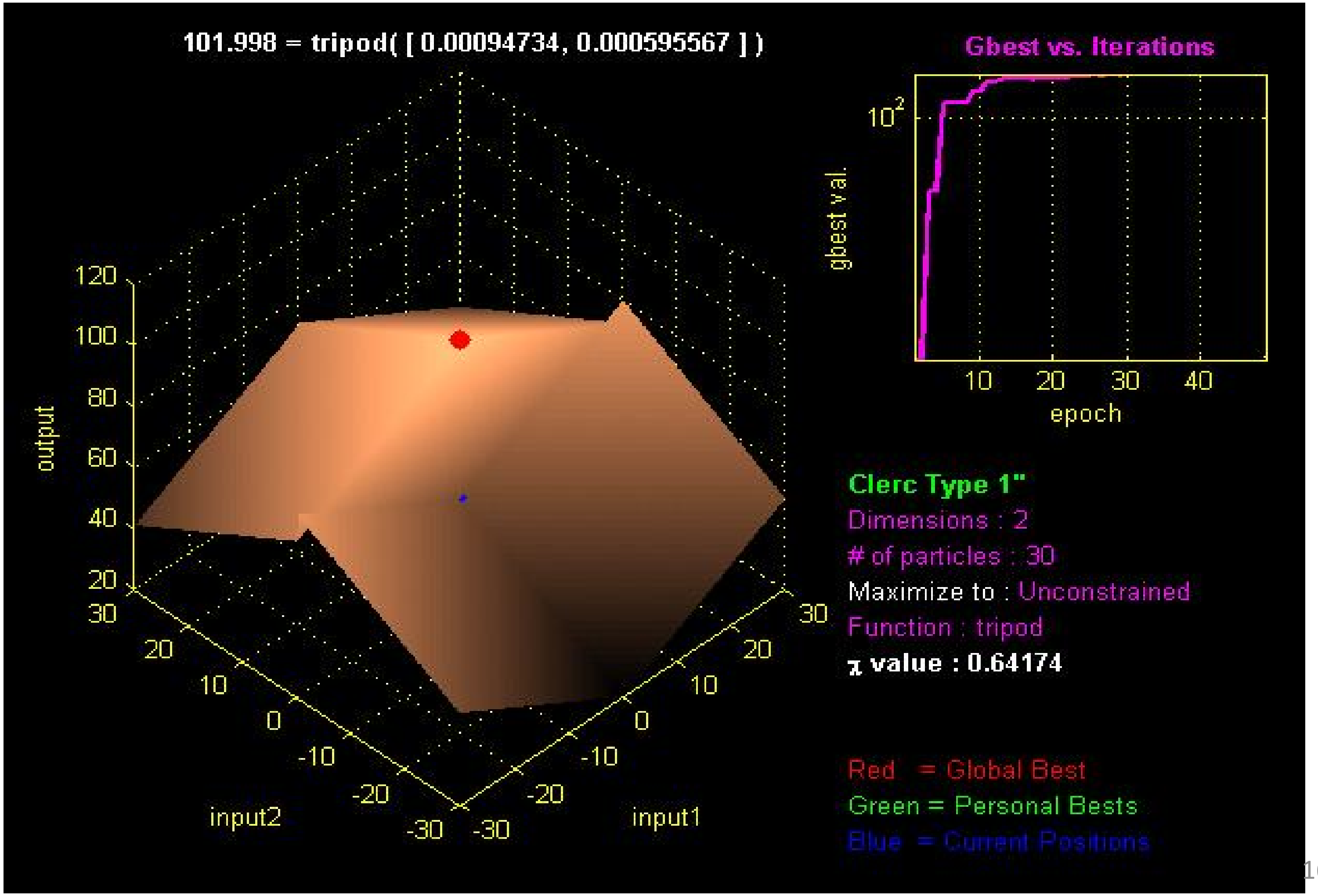
基本粒子群算法的描述

4. 粒子群算法流程图



• 基本粒子群算法的描述

5. 粒子群算法迭代过程



$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

6. 粒子群算法参数分析

（1）惯性权重w

使粒子保持运动惯性，使其有搜索扩展空间的趋势，有能力探索新的区域。

也表示微粒对当前自身运动状态的信任，依据自身的速度进行惯性运动。

较大的w有利于跳出局部极值，而较小的w有利于算法收敛。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

6. 粒子群算法参数分析

（2）改进的惯性权重w

在优化实际优化问题时，往往希望先采用全局搜索，使搜索空间快速收敛于某一区域，然后采用局部精细搜索以获得高精度的解。

因此提出了自适应调整的策略，即随着迭代的进行，线性地减小 w 的值。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

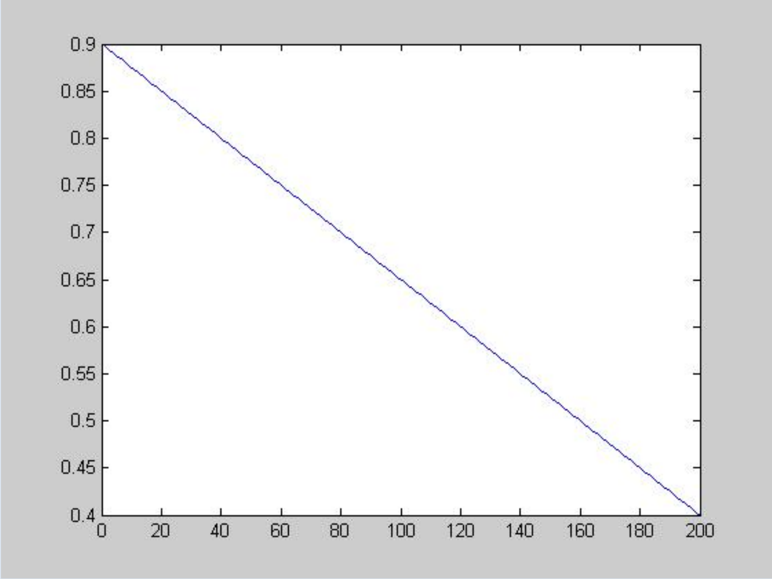
6. 粒子群算法参数分析

（2）改进的惯性权重w

wmax、wmin分别是w的最大值和最小值；

iter、itermax分别是当前迭代次数和最大迭代次数。

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter$$



$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

6. 粒子群算法参数分析

（3）加速因子c1和c2

使代表将微粒推向pbest和gbest位置的统计加速项的权重。

表示粒子的动作来源于自己经验的部分和其它粒子经验的部分。

低的值粒子在目标区域外徘徊，而高的值导致粒子越过目标域。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

6. 粒子群算法参数分析

（4）改进的加速因子c1和c2

通常将c1和c2统一为一个控制参数，φ= c1+c2

如果φ很小，微粒群运动轨迹将非常缓慢；

如果φ很大，则微粒位置变化非常快；

通过仿真可以获得φ的经验值，当φ=4.0（c1=2.0，c2=2.0）时，具有很好的收敛效果。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

6. 粒子群算法参数分析

（5）粒子数

通常一般取20～40，对较难或特定类别的问题可以取100～200。

（6）最大速度vmax

决定粒子在一个循环中最大的移动距离，通常设定为粒子的范围宽度。

7. 粒子群算法与遗传算法的比较

共性：

（1）都属于仿生算法；

（2）都属于全局优化方法；

（3）都属于随机搜索算法；

（4）都隐含并行性；

（5）根据个体的适配信息进行搜索，因此不受函数约束条件的限制，如连续性、可导性。

（6）对高维复杂问题，无法保证收敛到最优点。

7. 粒子群算法与遗传算法的比较

差异：

（1）PSO有记忆，所有粒子都保存较优解的知识，而GA，以前的知识随着种群的改变被改变；

（2）PSO中的粒子是一种单向共享信息机制。而GA中的染色体之间相互共享信息；

（3）GA需要编码和遗传操作，粒子只是通过内部速度进行更新，实现更容易。

蚁群算法

- **蚁群算法的思想和起源**

20 世纪90 年代初，意大利学者Dorigo 等受蚂蚁觅食行为的启发，在其博士论文中首次提出了蚂蚁系统（Ant System）。

近年来，Dorigo等人进一步将蚂蚁算法发展为一种通用的优化技术--蚁群优化（Ant Colony Optimization）。

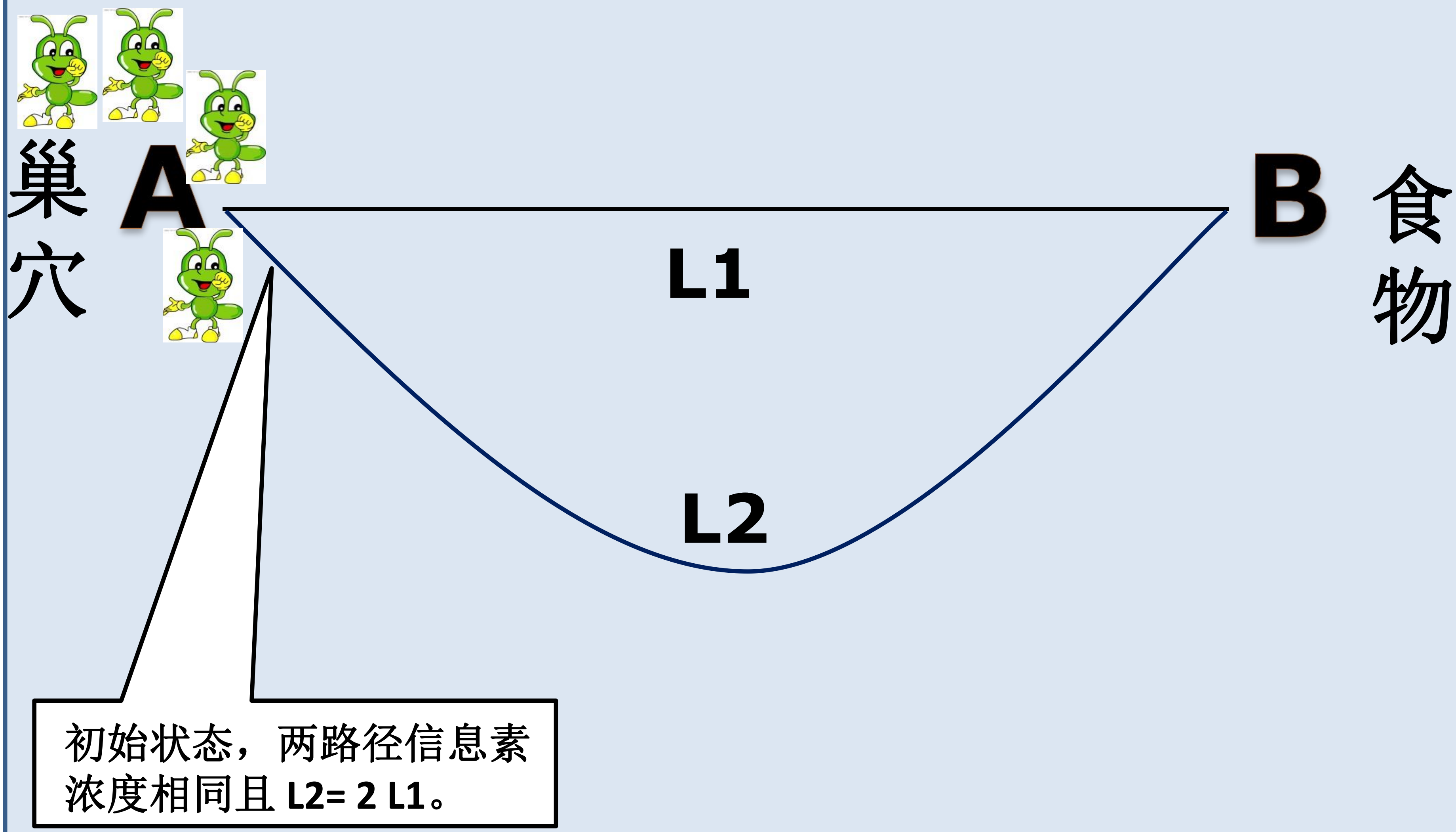
蚂蚁觅食是寻找从巢穴到食物的最佳路径的行为。蚁群算法是一种仿生算法。

- **蚂蚁可以找出最短路径，为什么?**

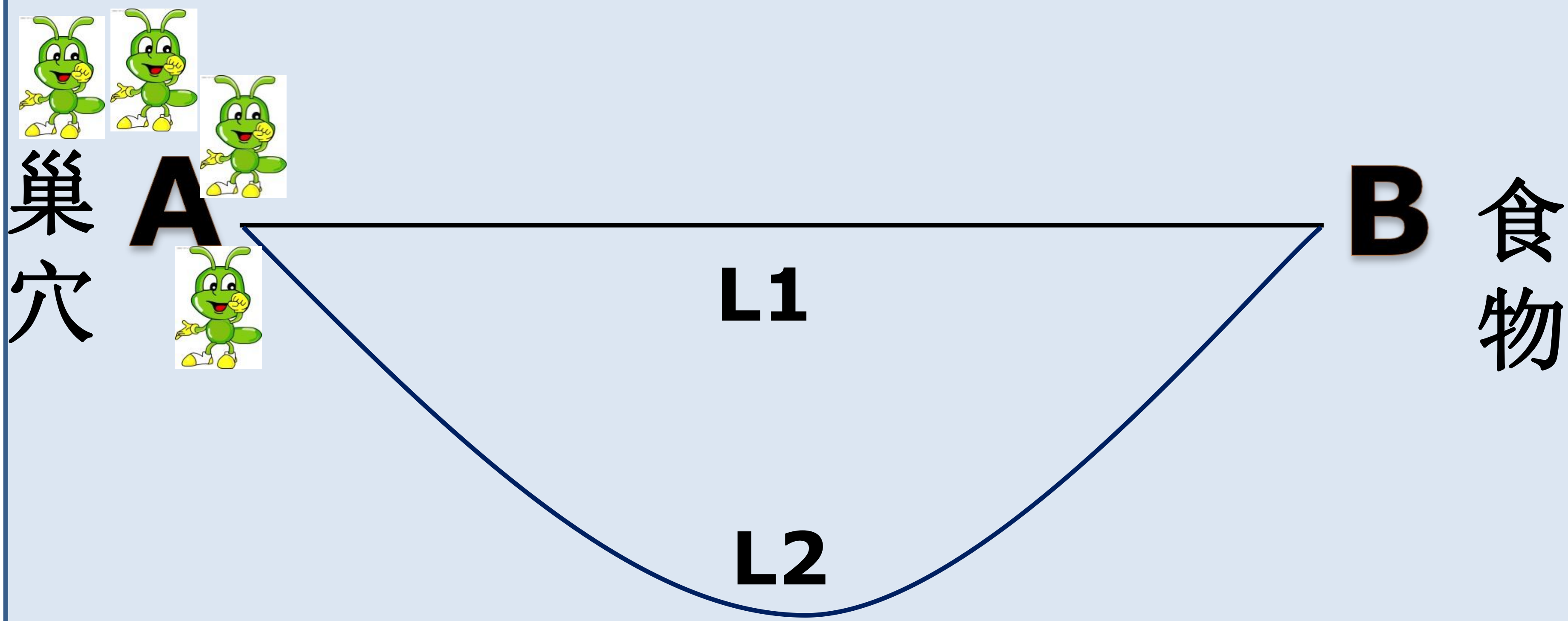
  1. 信息素（**pheromone**）：蚂蚁在寻找食物时，其经过的路 上释放的一种易挥发的物质。该信息素 可以被其它的蚂蚁感知，并且信息素的浓度越高，对应的路径越短。

  2. 正反馈：蚂蚁会以较大的概率选择信息素浓度较高的路径，并释放一定量的信息素以增强该路径上的信息素浓素，从而距离较短的路径被加强，形成一个正反馈。

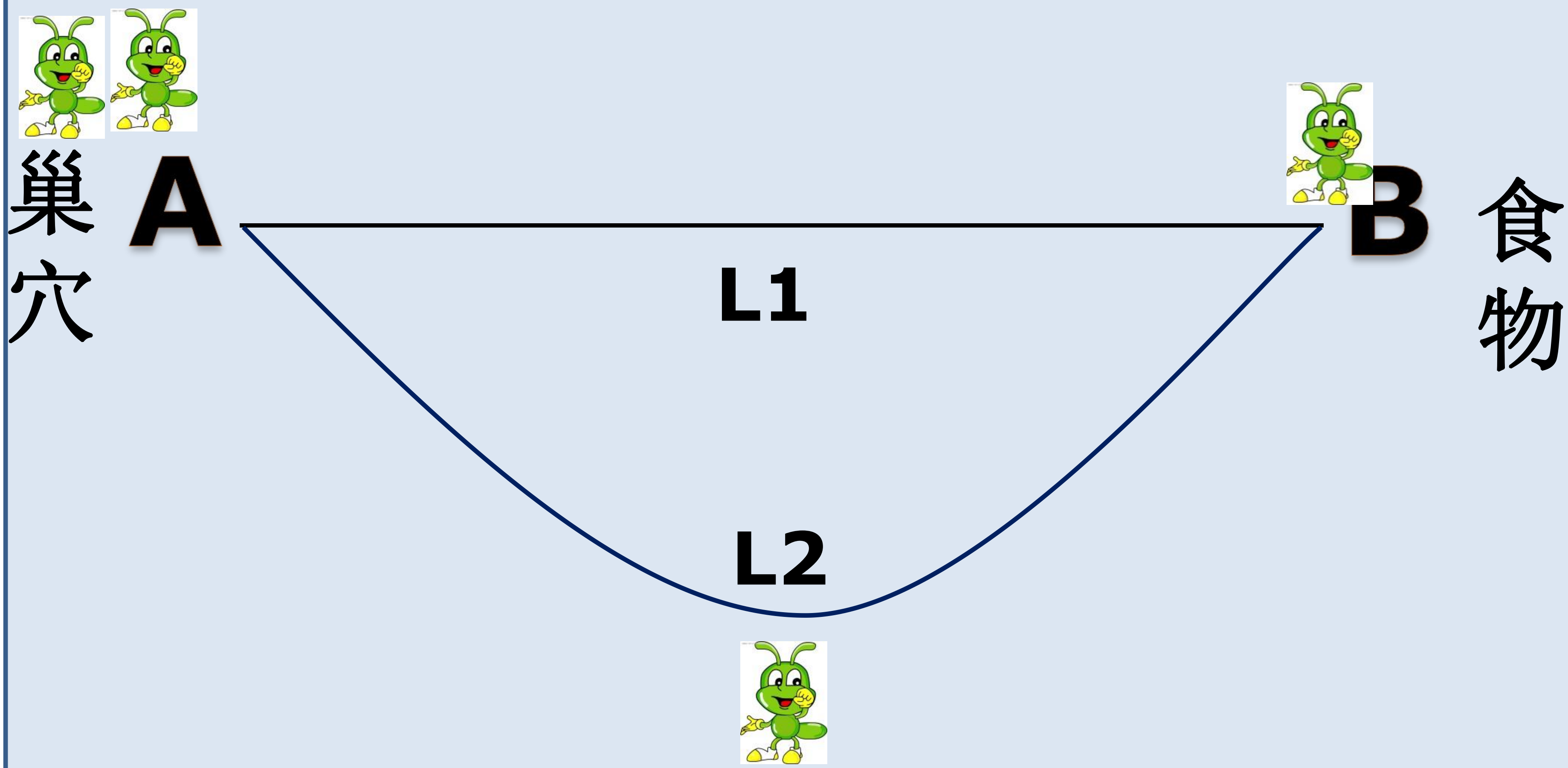
蚂蚁寻找最短路径示意图



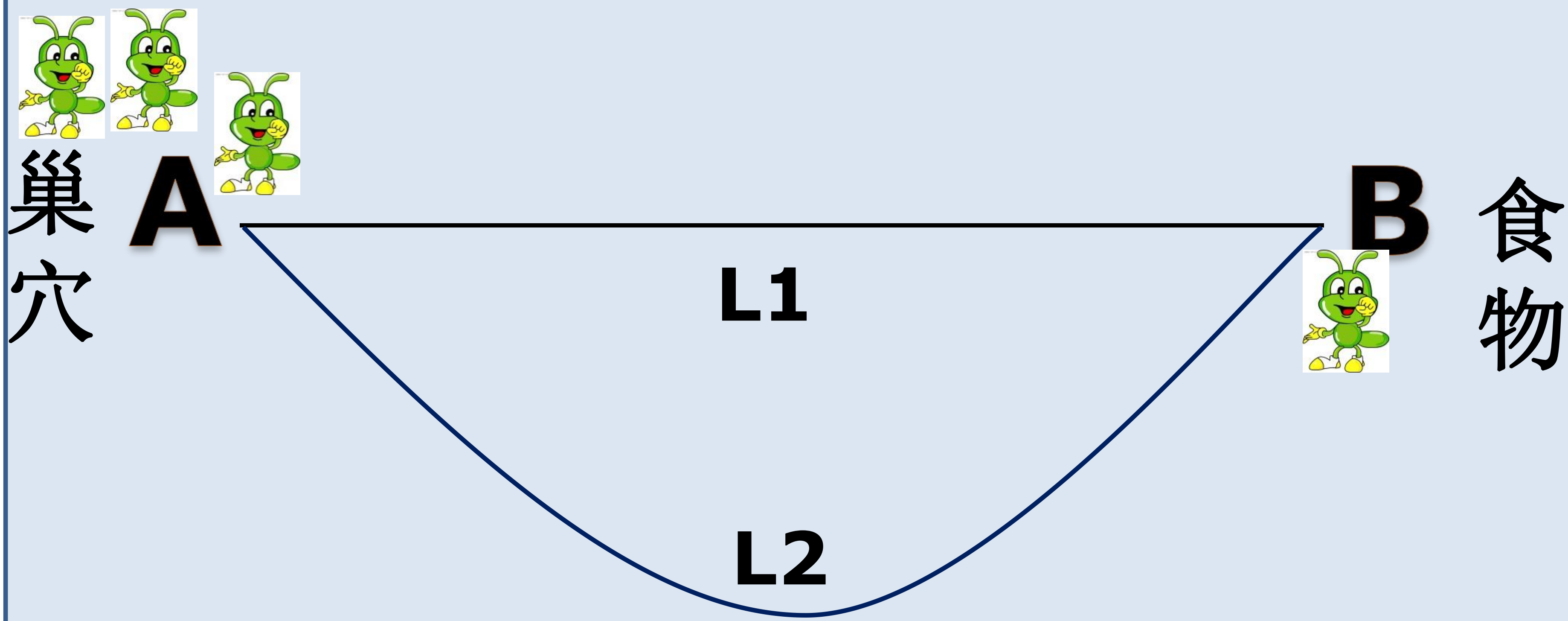
蚂蚁寻找最短路径示意图



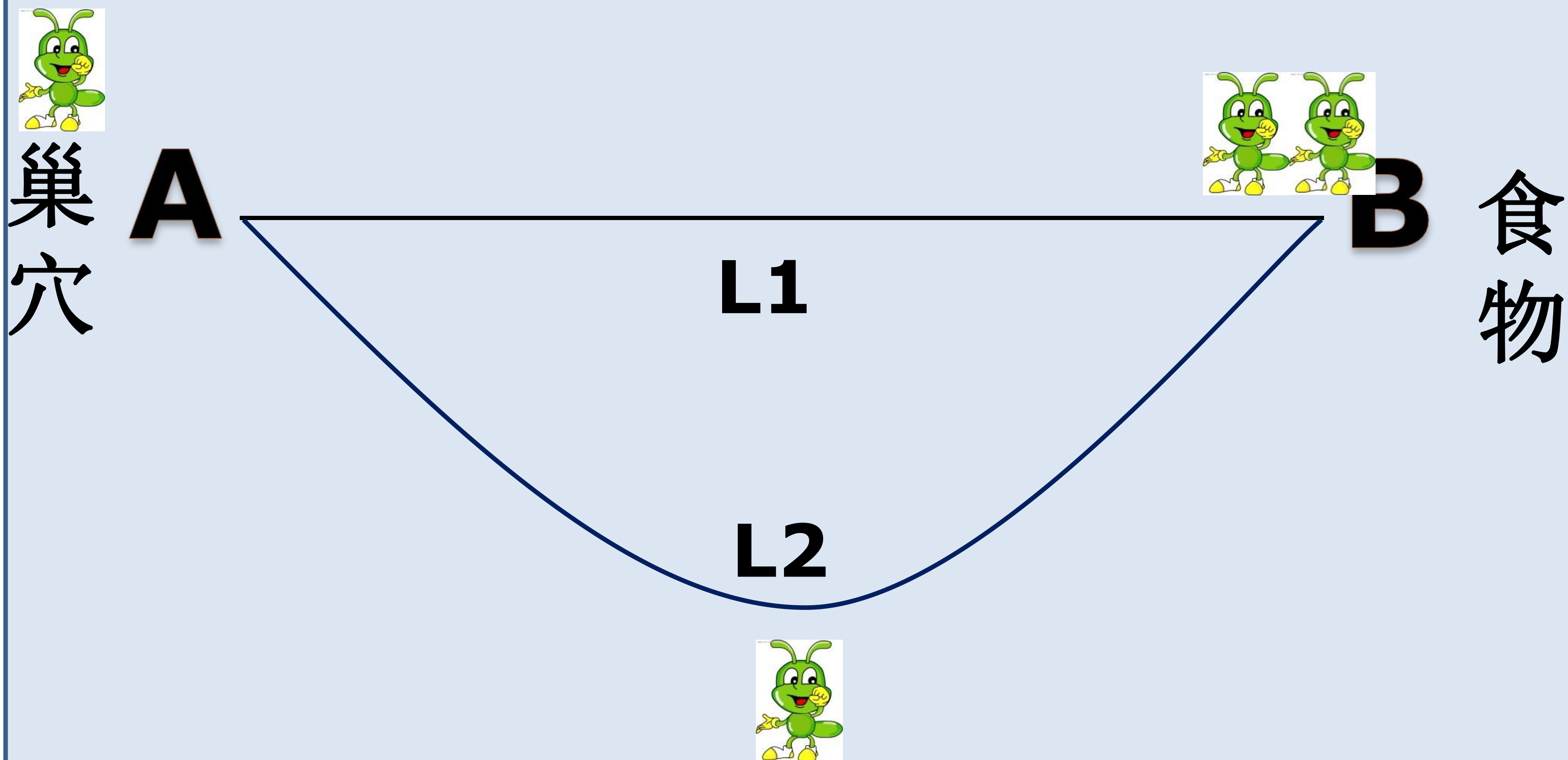
蚂蚁寻找最短路径示意图



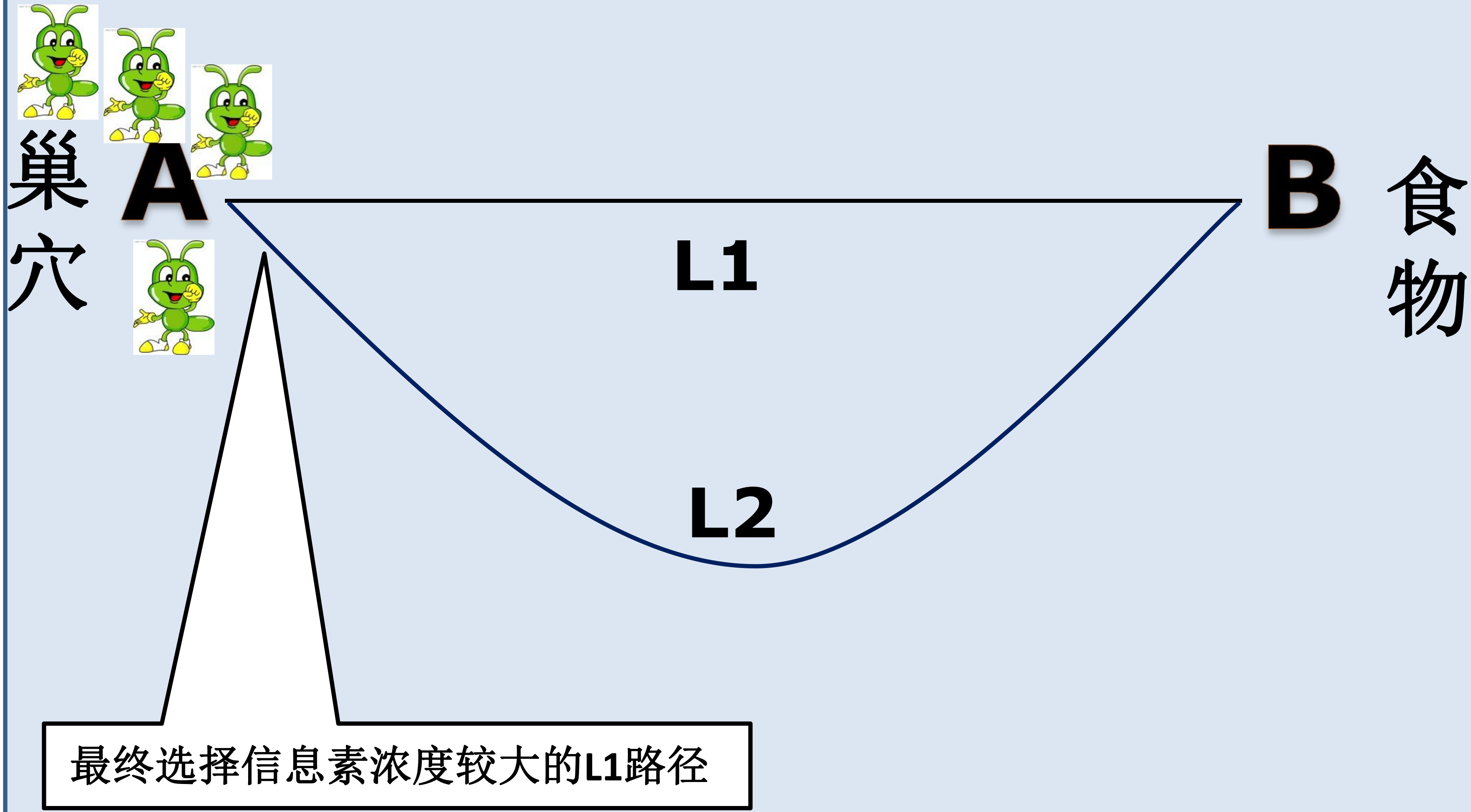
蚂蚁寻找最短路径示意图



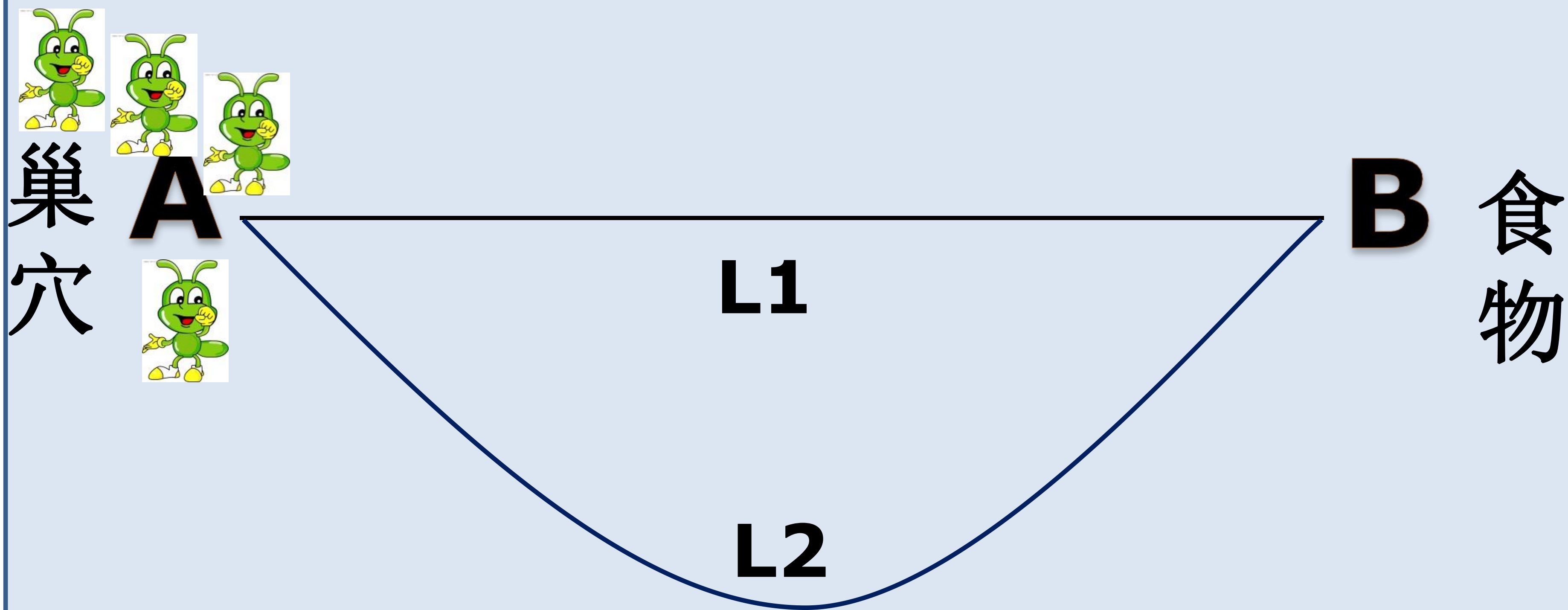
蚂蚁寻找最短路径示意图



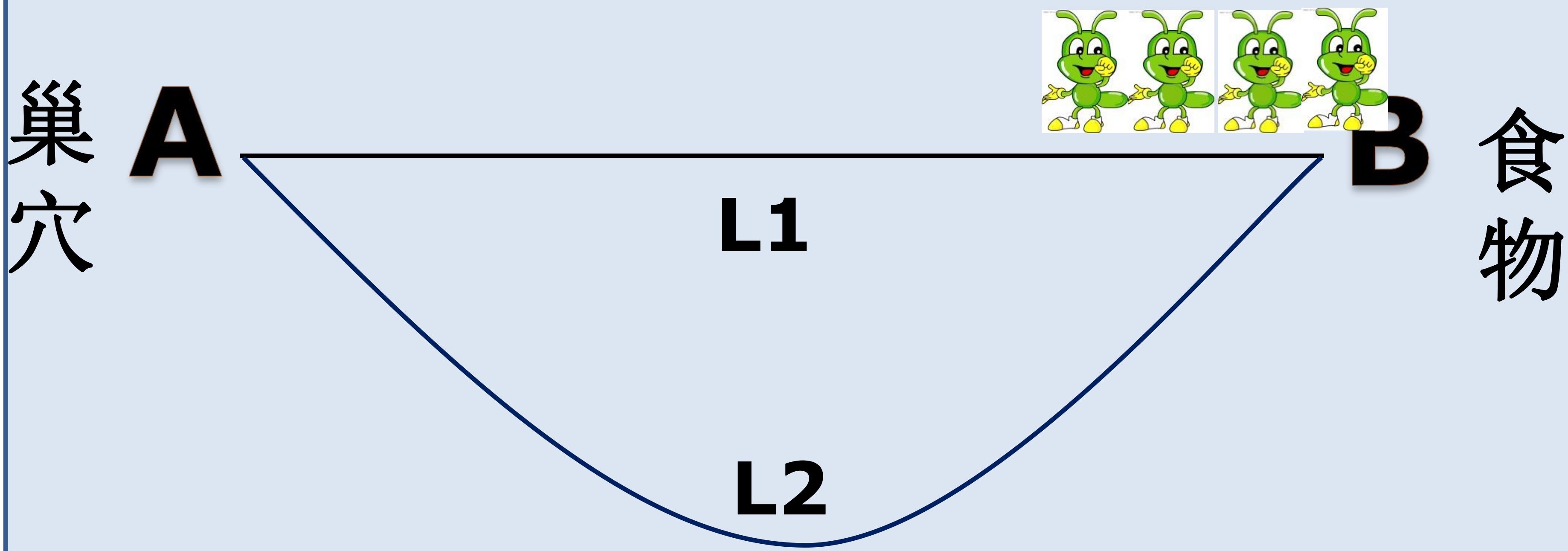
蚂蚁寻找最短路径示意图



蚂蚁寻找最短路径示意图



蚂蚁寻找最短路径示意图



蚁群算法的模型与实现----TSP

1. 不失一般性设蚂蚁的数量为 m，城市的数量为 n，城市 i和城市 j的距离为 $d(i, j)$ 距离选用欧式距离，t时刻城市 i和城市 j连接路径的信息素浓度为 $\tau(i, j)$ 。

2. 在算法初始时刻，设各城市连接路径的信息素浓度具有相同的值，m只蚂蚁放到 n座城市。

蚁群算法的模型与实现----TSP

3. 蚂蚁的初始分布

（1）所有蚂蚁初始时刻放在同一城市。

（2）所有蚂蚁初始时刻分布在不同城市中。

显而易见，第二种方法将蚂蚁放在不同的城市中算法具有较高的性能。在不同城市 分布时，随机分布与统一均匀分布的效果 差别不大。

蚁群算法的模型与实现----TSP

4. 每只蚂蚁根据路径上的信息素和启发式信息，独立地访问下一座城市，概率公式如下：

$$P^k(i,j)=\begin{cases}\dfrac{[\tau(i,j)]^\alpha\cdot[\eta(i,j)]^\beta}{\sum\limits_{s\notin tabu_k}[\tau(i,s)]^\alpha\cdot[\eta(i,s)]^\beta}, & if\ j\notin tabu_k\\ 0, & otherwise\end{cases}$$

$$\eta(i,j)=1/d(i,j)$$

是启发函数，表示蚂蚁从城i到城市j的期望程度，距离越短函数值越大。

蚁群算法的模型与实现----TSP

$$P^k(i,j)=\begin{cases}\dfrac{[\tau(i,j)]^{\alpha}\cdot[\eta(i,j)]^{\beta}}{\sum\limits_{s\notin tabu_k}[\tau(i,s)]^{\alpha}\cdot[\eta(i,s)]^{\beta}}, & if\ j\notin tabu_k\\ 0, & otherwise\end{cases}$$

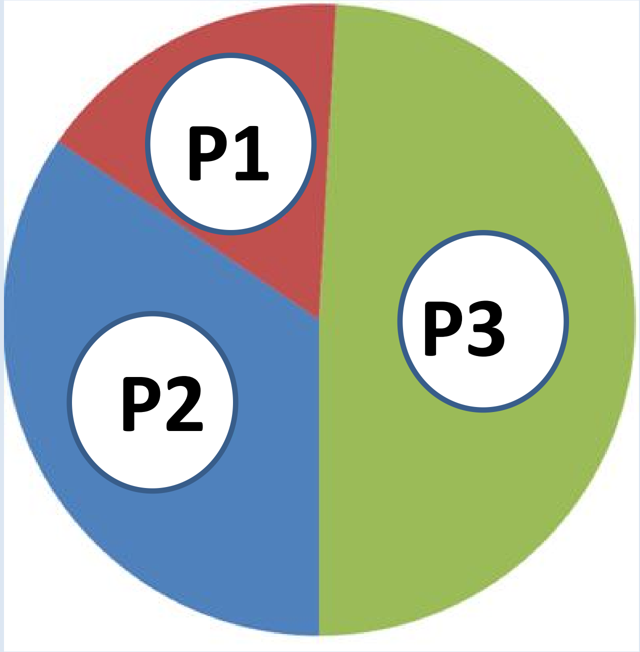
α 是信息素重要程度因子。

β 是启发函数重要程度因子。

$tabu_k$为禁忌表，表示已经访问的城市集合。

蚁群算法的模型与实现----TSP

5. 蚂蚁从当前城市访问下一城市的概率确定后，通常采用轮盘赌法选择下一城市，概率大被选中机会就大。



蚁群算法的模型与实现----TSP

6. 当所有蚂蚁完成一次访问后，各路径上的信息素将进行更新，信息素公式更新如下：

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

$$\Delta\tau_{ij} = \sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

其中 ρ 的取值为 $0<\rho<1$，表示路径上信息素的挥发系数。

蚁群算法的模型与实现----TSP

7. 针对蚂蚁释放信息素问题，比较常用的有如下三种模型：

(1). Ant cycle system

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & ij \in l_k \\ 0 & otherwise \end{cases}$$

Q为正常数，L_k表示第k只蚂蚁在本次访问城市中所走过路径的长度

蚁群算法的模型与实现----TSP

7. 针对蚂蚁释放信息素问题，比较常用的有如下三种模型：

(2). Ant quantity system

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{D_{ij}} & ij \in l_k \\ 0 & otherwise \end{cases}$$

Q为正常数，D_{ij}表示第k只蚂蚁在本次访问中城市 i和城市 j的距离。

蚁群算法的模型与实现----TSP

7. 针对蚂蚁释放信息素问题，比较常用的有如下三种模型：

(3). Ant density system

$$\Delta\tau_{ij}^{k} = \begin{cases} Q & ij \in l_k \\ 0 & otherwise \end{cases}$$

Q为正常数，在整个访问城市的过程中，密度始终保持不变。

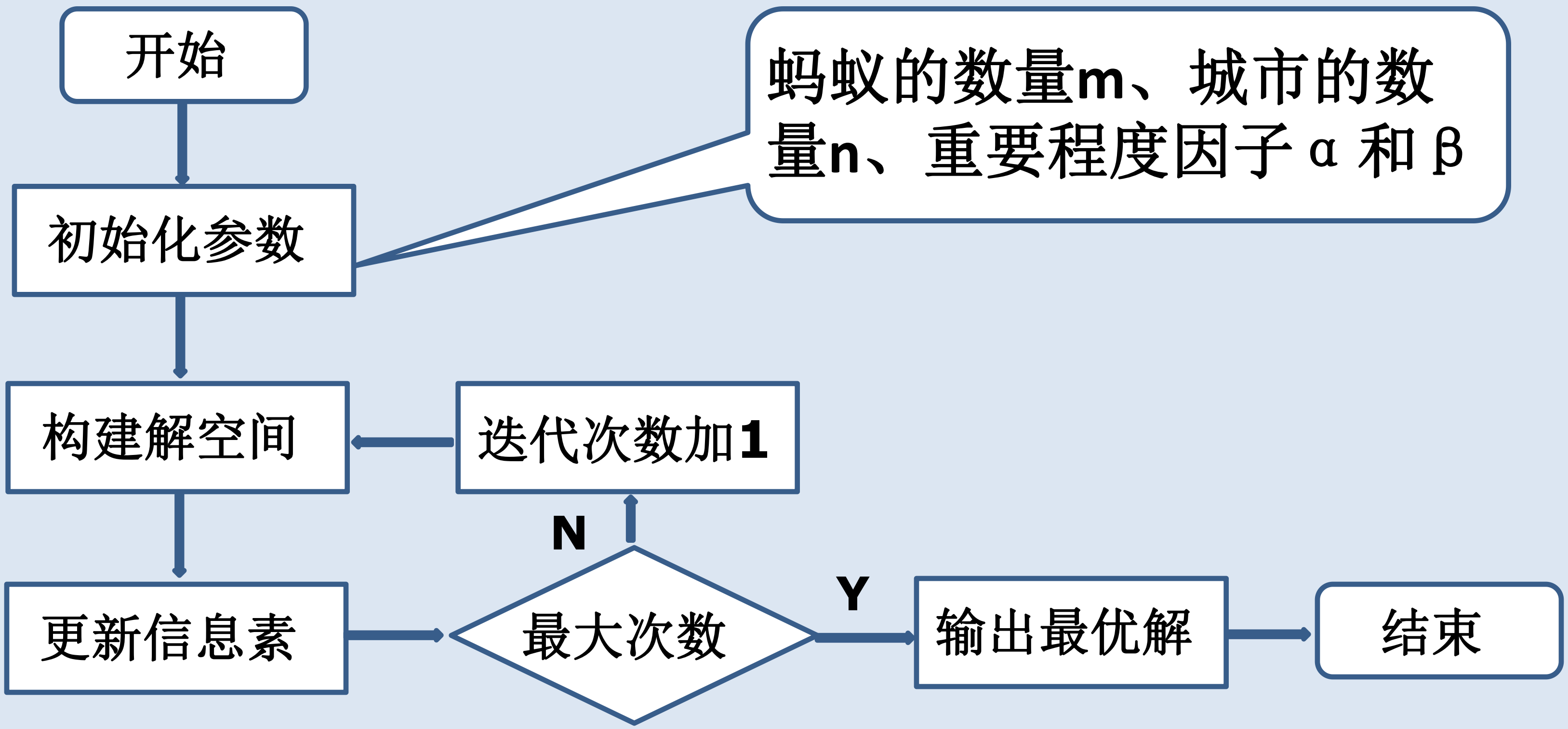
蚁群算法的模型与实现----TSP

7. 针对蚂蚁释放信息素问题，这三种模型分别对应路径的整体信息（蚂蚁所访问路径的总长）、局部信息（蚂蚁所访问城市间的距离）和不考虑路径信息。

以下优化TSP问题，选用ant cycle system模型，即路径的整体信息路径越短，释放的信息素度越高。

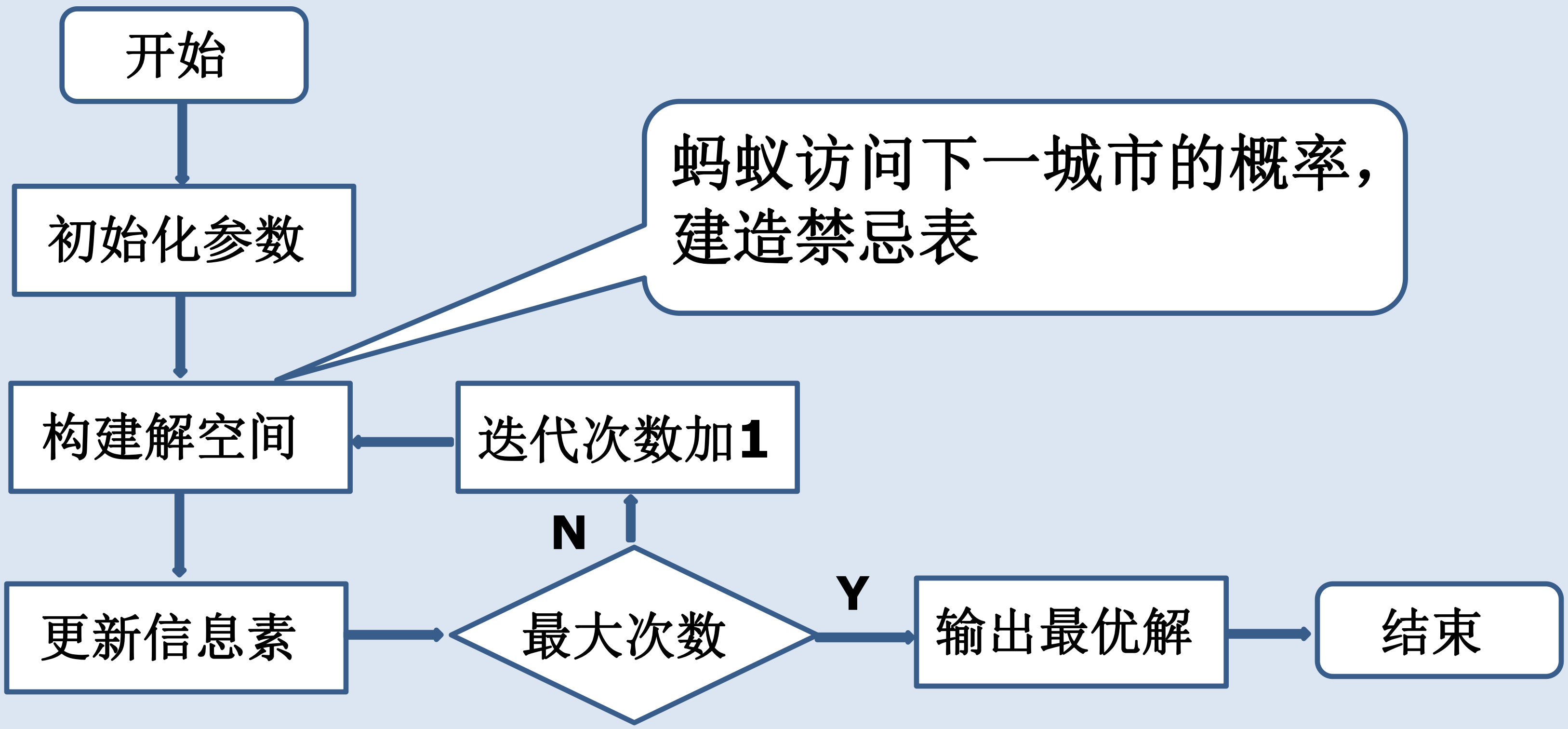
蚁群算法的模型与实现---TSP

8． 蚁群算法解决TSP问题基本流程



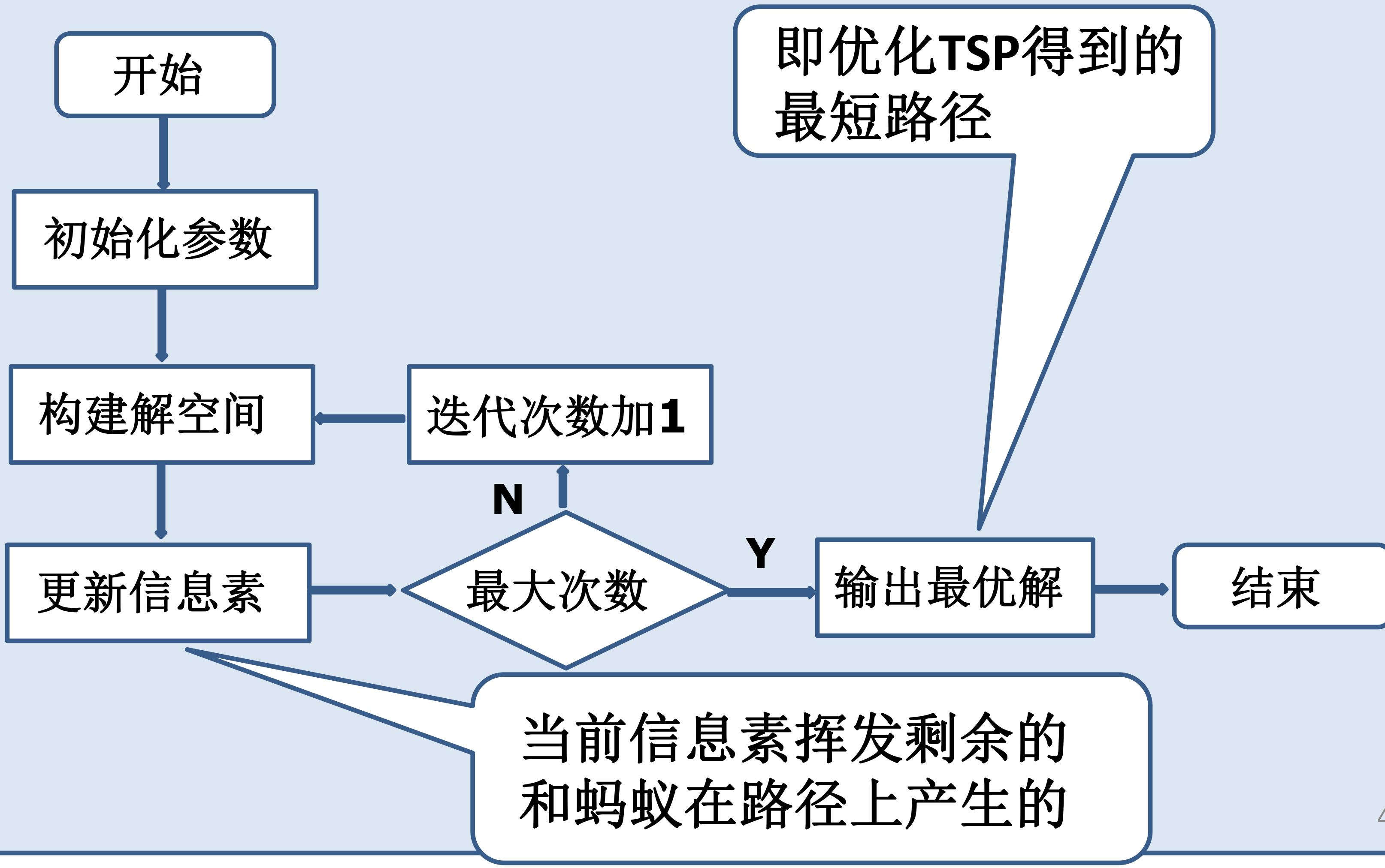
蚁群算法的模型与实现----TSP

8. 蚁群算法解决TSP问题基本流程



蚁群算法的模型与实现----TSP

8. 蚁群算法解决TSP问题基本流程



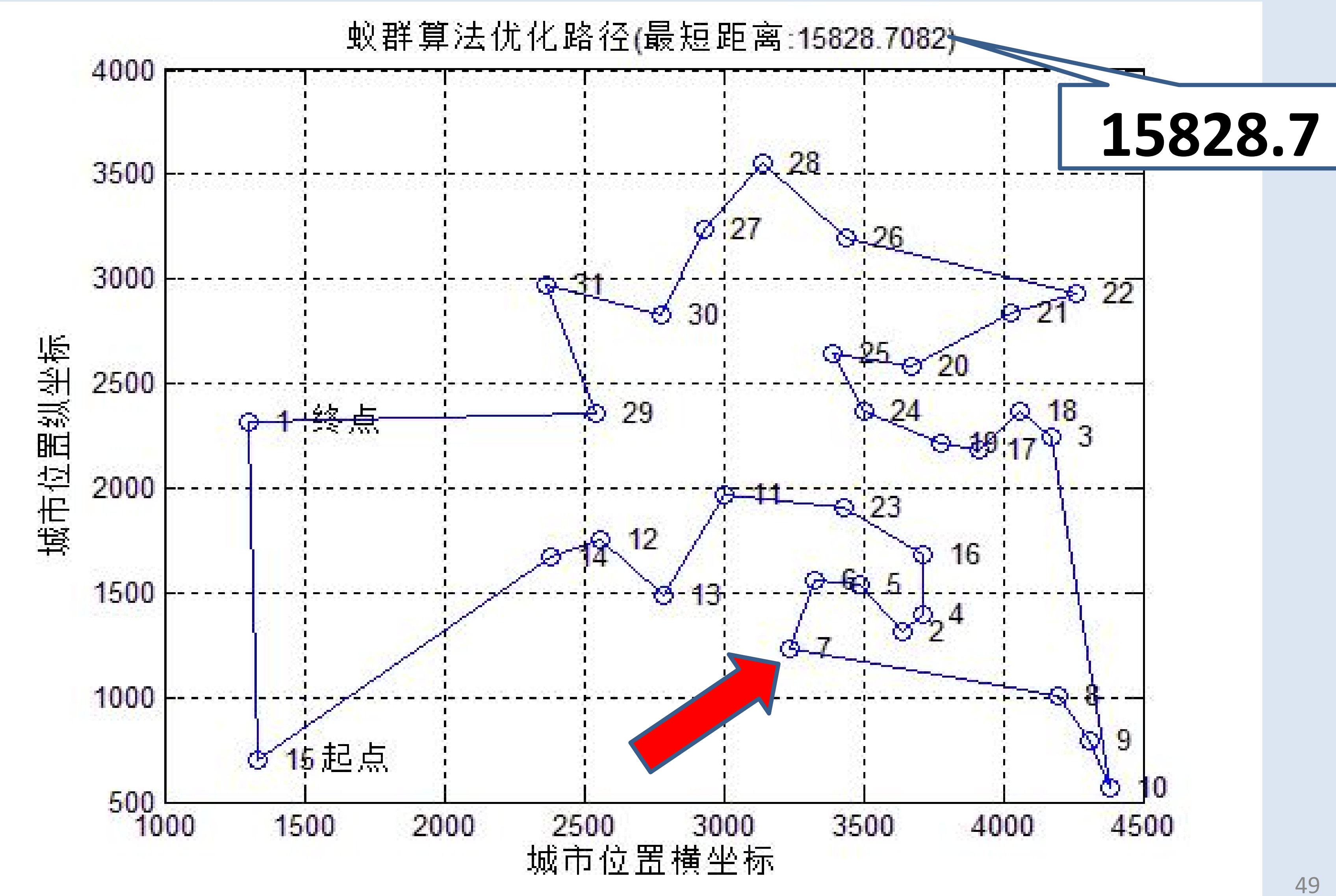
蚁群算法的模型与实现----TSP

1. 31个城市坐标如下：

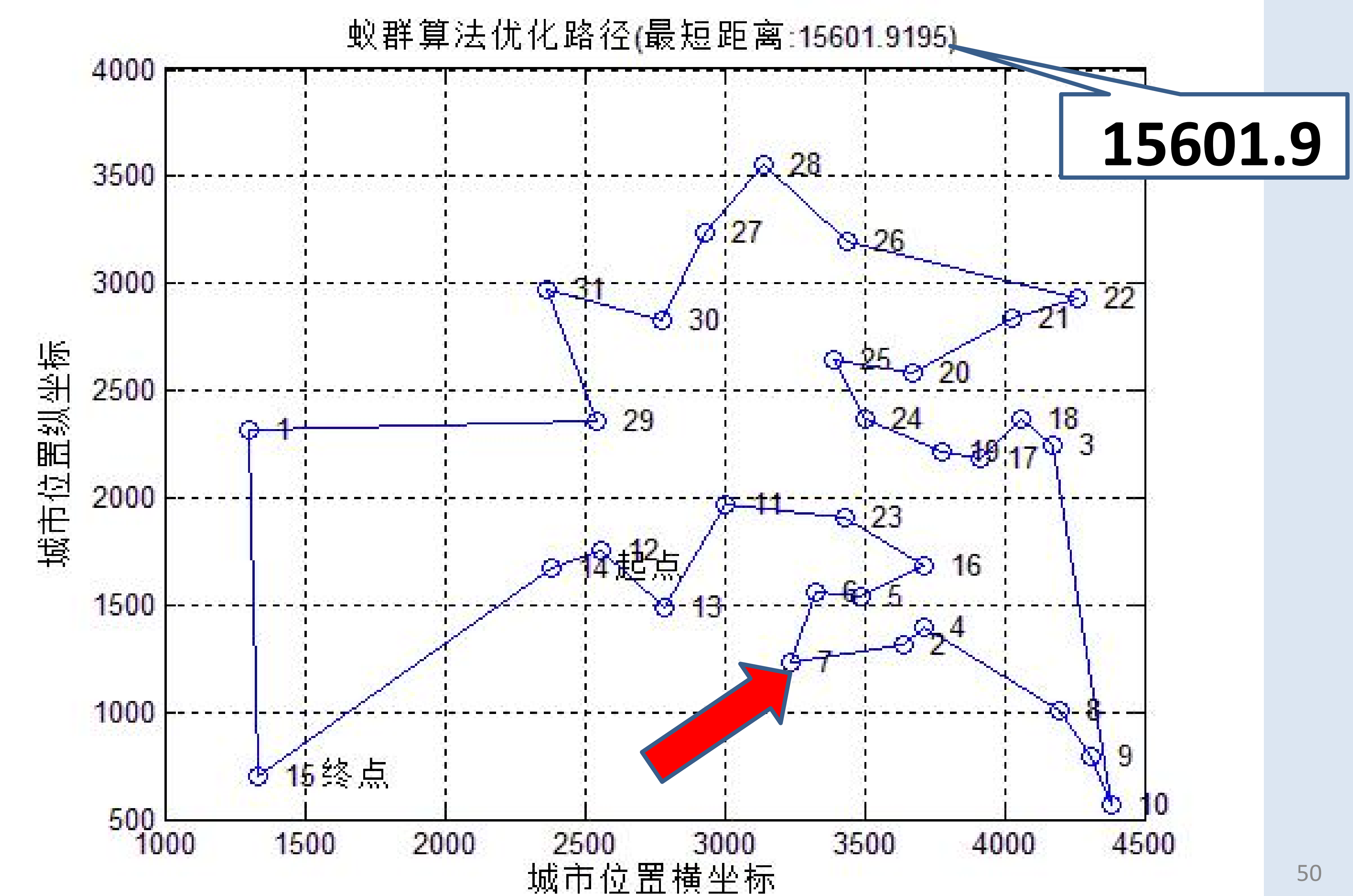
x=[1304,3639,4177,3712,3488,3326,3238,4196,4312,4386,
3007,2562,2788,2381,1332,3715,3918,4061,3780,3676,4029,
4263,3429,3507,3394,3439,2935,3140,2545,2778,2370];

y=[2312,1315,2244,1399,1535,1556,1229,1004,790,570,1970,
1756,1491,1676,695,1678,2179,2370,2212,2578,2838,2931,
1908,2367,2643,3201,3240,3550,2357,2826,2975]

蚁群算法的模型与实现----TSP



蚁群算法的模型与实现----TSP



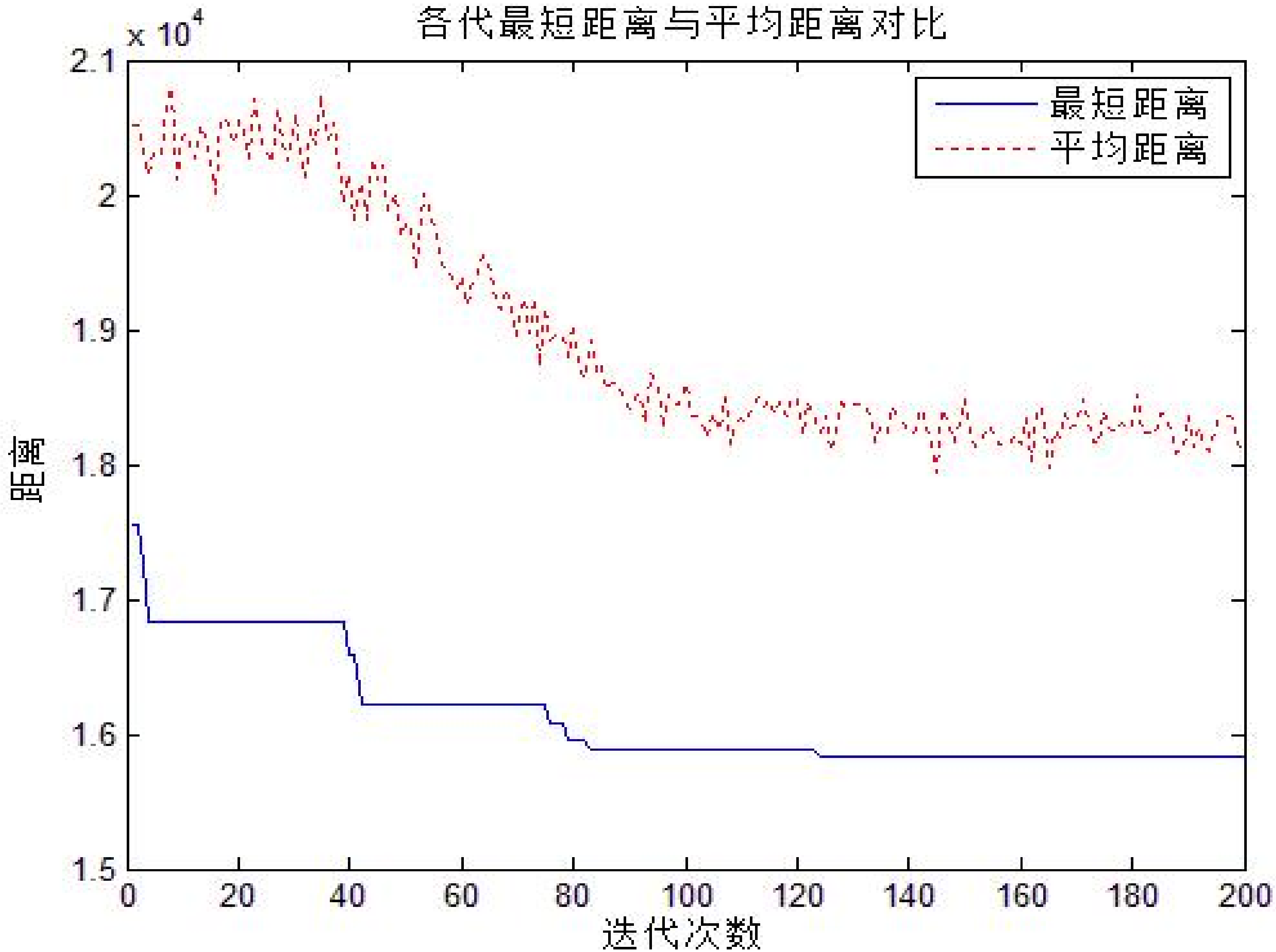
蚁群算法的模型与实现----TSP

1. 同一个函数，两次的结果并不是完全的一致，因为在蚁群初始化时位置是随机产生的。

说明该算法的解具有不确定性。

2. 两次优化的结果分别为15828km和15601km，而现在有的算法可以优化到15377km，因此寻找的最佳路径即为局部最优值并不是全局最优值。

蚁群算法的模型与实现----TSP



蚁群算法的模型与实现----TSP

1. 路径的最短距离在算法迭代到约120次时即蓝色实线基本保持不再发生变化，说明选找到极值点。

2. 同时路径的平均距离在算法迭代到100次左右时，在值为18500上下小范围内波动，基本趋于稳定。

蚁群算法

- **蚁群算法的优点**

1. 蚁群算法与其他启发式算法相比，在求解性能上，具有很强的鲁棒性（对基本蚁群算法模型稍加修改，便可以应用于其他问题）和搜索较好解的能力。

2. 蚁群算法是一种基于种群的进化算法，具有本质并行性，易于并行实现。

3. 蚁群算法很容易与多种启发式算法结合如遗传算法、粒子群算法，以改善算法性能。

蚁群算法

- **蚁群算法的不足**

1. 如果初始化参数设置不当，导致求解速度很慢且所得解的质量特别差。

2. 基本蚁群算法即无改进的蚁群算法计算量大，求解所需时间较长。

3. 基本蚁群算法理论上要求所有的蚂蚁选择同一路线，该线路即为所求的最优线路；但在实际计算中，在给定一定循环数的条件下很难达到这种情况。

蚁群算法

- **蚁群算法的改进**

  1. 最优解保留策略（**Ant System with Elitist**）

  该策略能够以更快的速度获得最好解，但是如果选择的精英过多则算法会由于较早收敛于局部次优解而导致搜索的过早停滞。

  2. 局部信息素更新

  使已选的路径对后来的蚂蚁具有较小的影响力，从而使蚂蚁对没有选中的路径有更强的探索能力。

蚁群算法

- **蚁群算法的改进**

  3. 最大--最小蚂蚁系统（**max-min ant system**）

  （1）每次迭代后，只有最优解（最优蚂蚁）所属路径上的信息被更新；

  （2）为了避免过早收敛，将各条路径可能的信息素限制于[τmin ,τmax]；

  （3）在算法初始时刻，ρ取较小值，算法有更好的发现较好解的能力。 随着迭代次数的增加， ρ变大加快算法的收敛。

蚁群算法

- **蚁群算法的应用**

1. 控制工程、电力工程、交通工程
2. 网络工程、通信、计算机科学
3. 管理工程、化工、信息科学
4. 组合优化问题：

作业调度问题（**Job-shop sheduling problem**）

二次分配问题（**quadratic assignment problem**）

旅行商问题（**traveling salesman problem**）

群智能优化的特点与不足

共同特点：

基于概率计算的随机搜索进化算法，在结 构、研究内容、方法以及步骤上有较大的相似性；结果偏随机性。

存在的问题：

（1）数学理论基础相对薄弱；

（2）参数设置没有确切的理论依据，对具体问题和应用环境的依赖性大；

群智能优化的特点与不足

进一步的改进：

（1）进一步研究真实群居动物的行为特征，建立合适的数学模型；

（2）进一步研究算法的收敛性；

群智能优化的特点与不足

进一步的改进：

（3）进一步提高收敛速度，从而解决大规模优化问题；

（4）进一步研究各种参数设置问题；

（5）研究群智能的并行算法；

（6）进一步研究各算法的适用范围；

（7）研究与其它算法的混合技术。