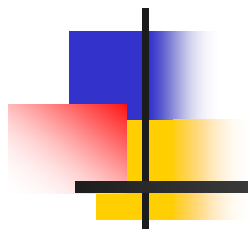
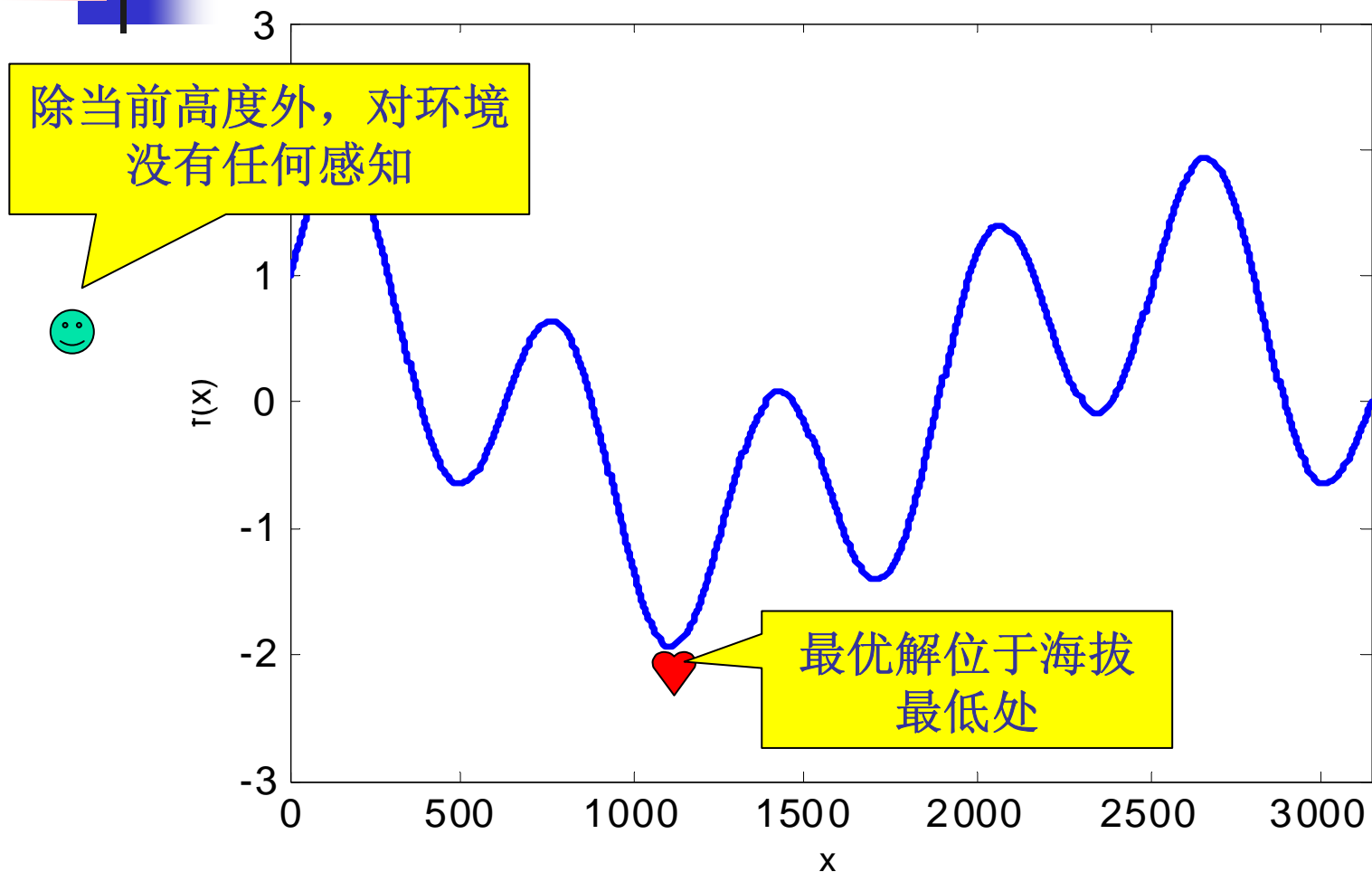


## 第二章 模拟退火算法 ( Simulated Annealing )

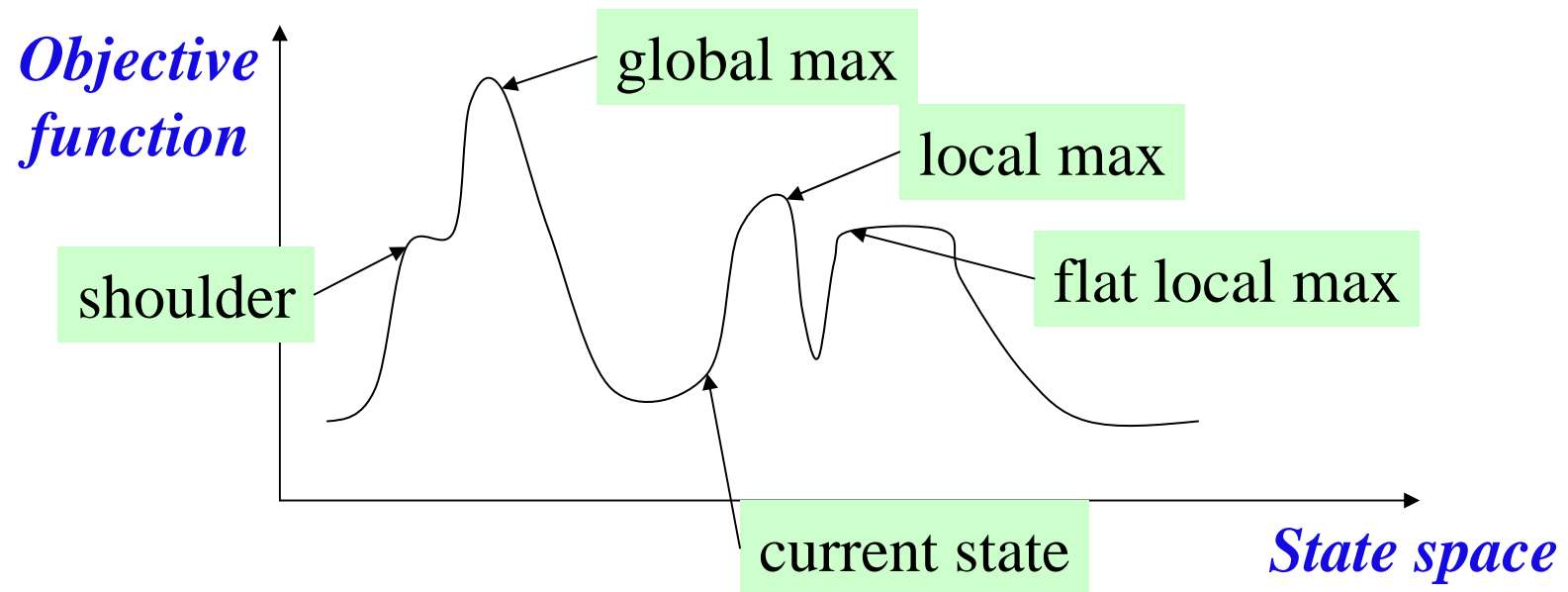


# 搜索问题描述



# 搜索问题描述

## ■ Landscape with various features





# 搜索算法

---

- 盲目搜索还是启发式搜索？
- 按照预定的控制策略实行搜索，在搜索过程中获取的中间信息不用来改进控制策略，称为盲目搜索，反之，称为启发式搜索。
- 关于“启发式”，可有两种看法：
  - 1)任何有助于找到问题的解，但不能保证找到解的方法均是启发式方法；
  - 2)有助于加速求解过程和找到较优解的方法是启发式方法。



# 搜索算法

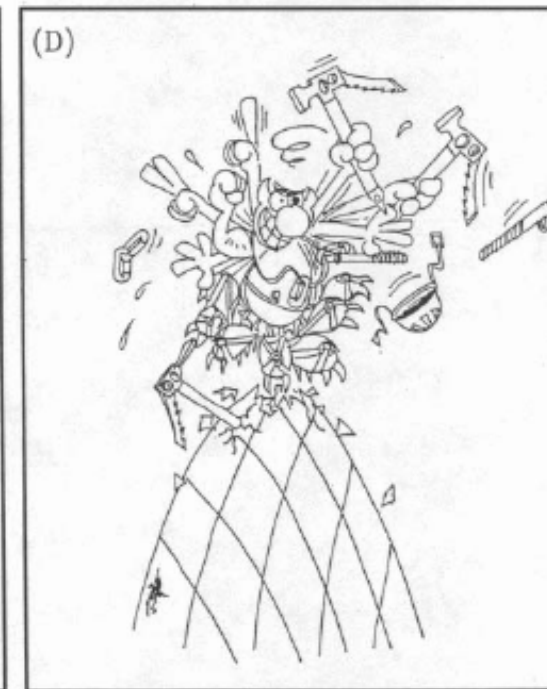
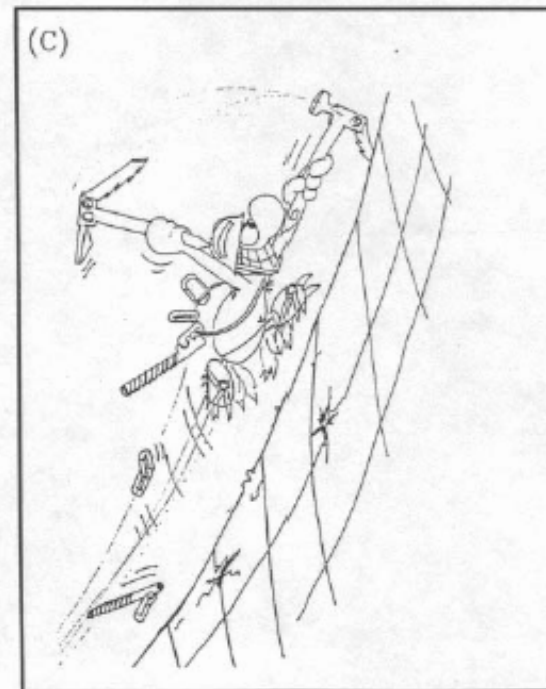
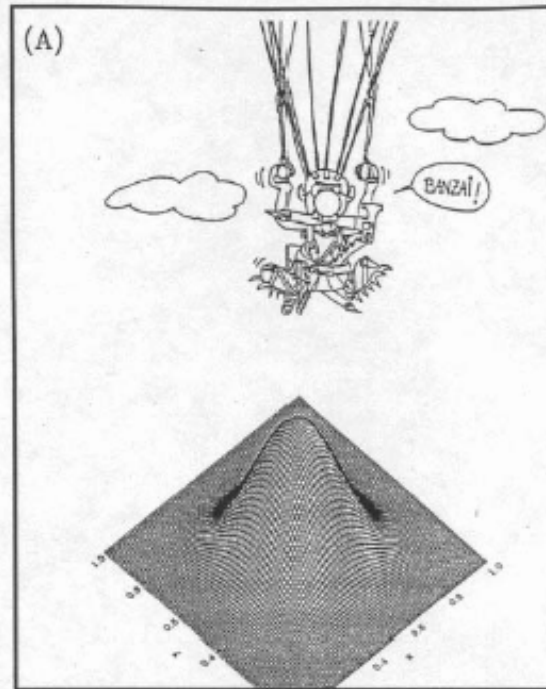
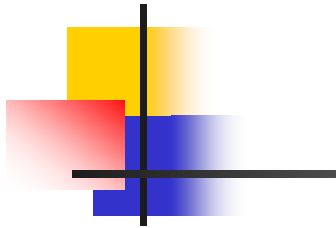
---

- 盲目搜索

- 深度优先、广度优先、代价优先、向前、向后、双向。。。

- 启发式搜索

- 爬山法、模拟退火算法、遗传算法、粒子群算法、蚁群算法。。。





# 贪心算法

---

1. 随机选定一个初始解 $x_0$ ;
2. **Do while** (中止条件不满足)
  1. 在某个邻域函数所定义的邻域范围内, 按照某个(随机)扰动 $\Delta$ 产生策略, 得到一个新解 $x_i'$ ;
  2. 对新解进行评估, 得 $f(x_i')$ ;
  3. 如果 $f(x_i') > f(x_i)$  (或者 $f(x_i') < f(x_i)$ ), 即新解比老解好, 则令 $x_{i+1} = x_i'$ ;
  4. 否则,  $x_{i+1} = x_i$ 。
3. **End Do**



# 爬山法

1. 随机选定一个初始解 $x_0$ ;
2. **Do while** (中止条件不满足)
  1. 在某个邻域函数所定义的邻域范围内, 按照某个(随机)扰动 $\Delta$ 产生策略, 得到多个新解 $X_{\text{new}}=\{x_i^1, x_i^2, \dots, x_i^k\}$ ;
  2. 对这组新解进行评估, 得 $\{f(x_i^1), f(x_i^2), \dots, f(x_i^k)\}$ ;
  3.  $x_{i+1} = x_i'$ ,  $x_i' \in X_{\text{new}}$ ,  $\forall x_i^j$ , ( $i=1,2,\dots,n; j=1,2,\dots,k$ ),  $f(x_i') > f(x_i)$  且  $f(x_i') > f(x_i^j)$  (或者  $f(x_i') < f(x_i)$  且  $f(x_i') < f(x_i^j)$ ), 即新的当前解比老解好, 并且是所有新解中最好的一个;
  4. 如果,  $\forall x_i^j$ , ( $i=1,2,\dots,n; j=1,2,\dots,k$ ),  $f(x_i) > f(x_i^j)$  (或者  $f(x_i) < f(x_i^j)$ ), 则  $x_{i+1} = x_i$ 。
3. **End Do**

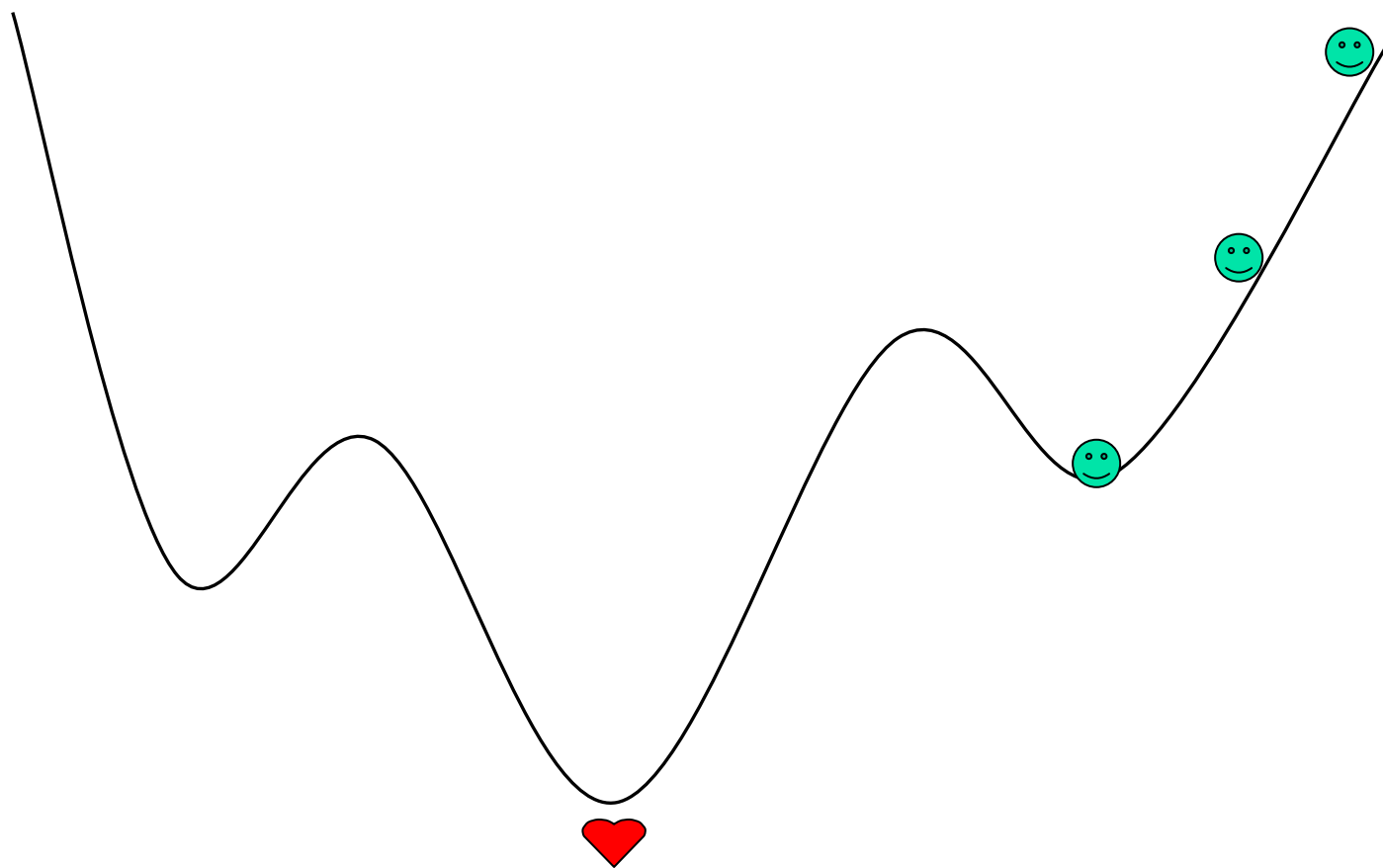




# 特点

---

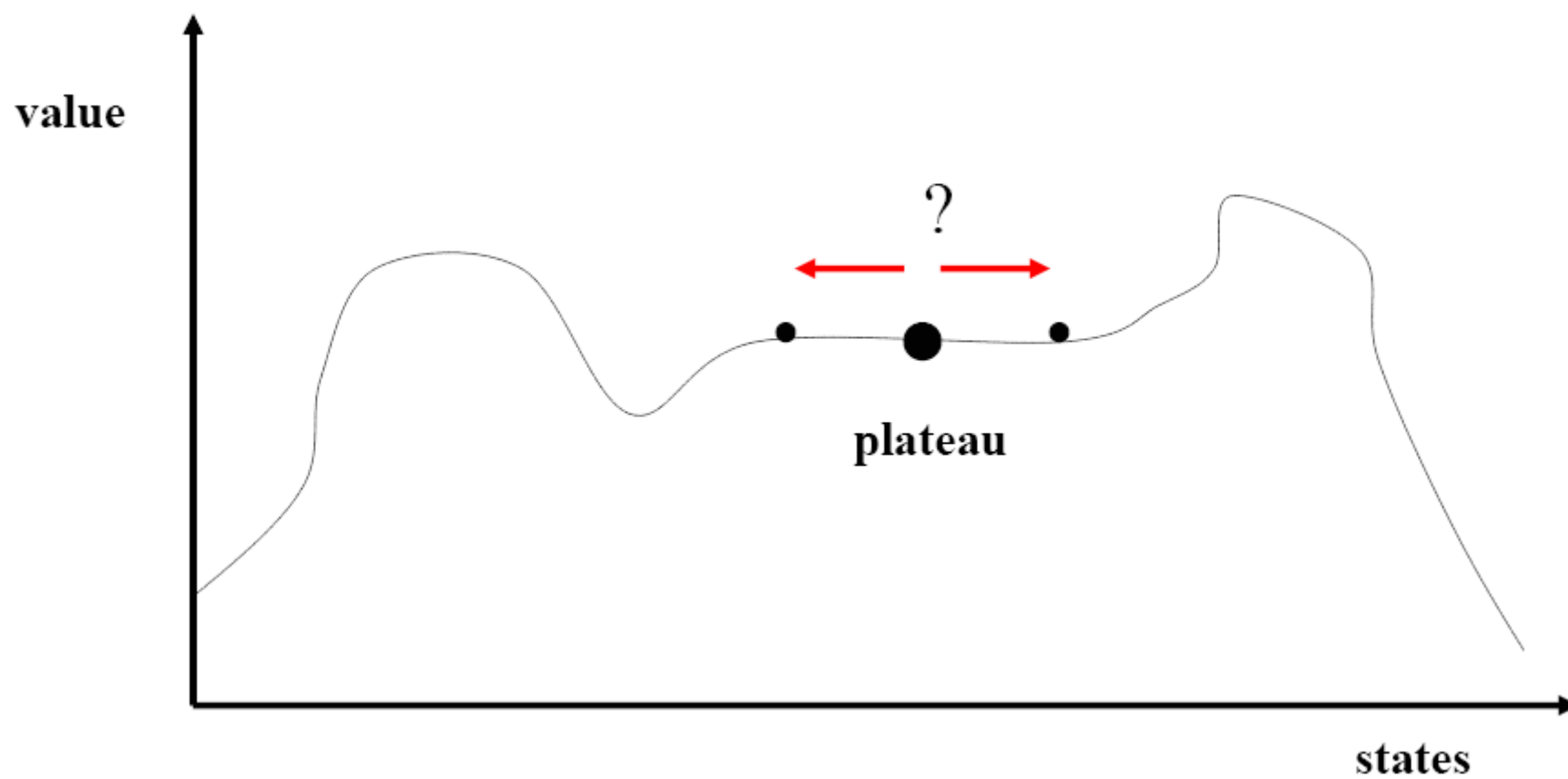
- 快速收敛于局部最优解





## 特点

- 遇到平台则无以事从





# 算法设计要素

---

- 编码策略（“个体表示”与“问题解”的映射关系）
- 初始解的产生（从什么位置开始搜索）
- 邻域函数的设计（下一个解的产生概率与当前解之间距离[包括方向和步长]的关系）
- 新解产生策略（随机，确定）
- 接受策略（贪心）

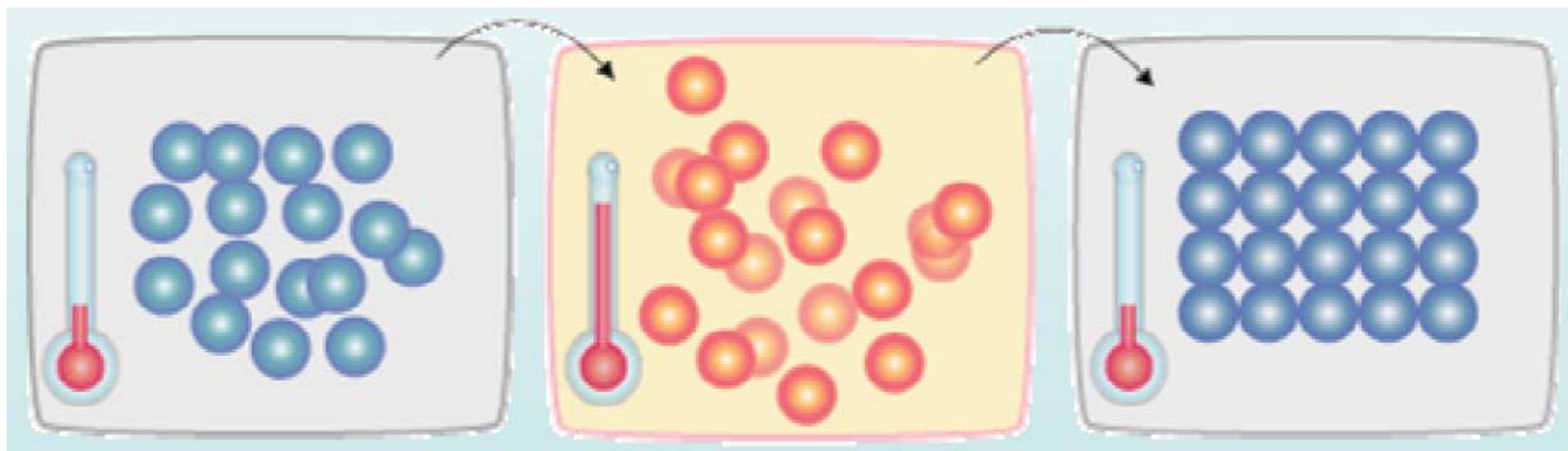


## 存在问题:

---

- 对初始解（状态）敏感
- 容易陷入局部最优

# 模拟退火算法（起源）



物理退火原理

## Real annealing: Sword

- He heats the metal, then slowly cools it as he hammers the blade into shape.
  - If he cools the blade too quickly the metal will form patches of different composition;
  - If the metal is cooled slowly while it is shaped, the constituent metals will form a uniform alloy.





# 模拟退火算法（起源）

---

- 物理退火过程：
  - 加温过程
  - 等温过程
  - 冷却（退火）过程
- 等温下热平衡过程可用Monte Carlo方法模拟，计算量大。
- 1953年，Metropolis提出重要性采样法，即以概率接受新状态，称Metropolis准则，计算量相对Monte Carlo方法显著减少。
- 1983年，Kirkpatrick等提出模拟退火算法，并将其应用于组合优化问题的求解。



# 模拟退火算法（Metropolis准则）

## ■ Metropolis准则

假设在状态 $x_{old}$ 时，系统受到某种扰动而使其状态变为 $x_{new}$ 。与此相对应，系统的能量也从 $E(x_{old})$ 变成 $E(x_{new})$ ，系统由状态 $x_{old}$ 变为状态 $x_{new}$ 的接受概率 $p$ ：

$$p = \begin{cases} 1 & \text{if } E(x_{new}) < E(x_{old}) \\ \exp\left(-\frac{E(x_{new}) - E(x_{old})}{T}\right) & \text{if } E(x_{new}) \geq E(x_{old}) \end{cases}$$





## 模拟退火算法与物理退火过程的相似关系

模拟退火	物理退火
解	粒子状态
最优解	能量最低态
设定初温	溶解过程
Metropolis采样过程	等温过程
控制参数的下降	冷却
目标函数	能量



## 模拟退火算法（流程）

- 1) 随机产生一个初始解 $x_0$ ，令 $x_{best} = x_0$ ，并计算目标函数值 $E(x_0)$ ;
- 2) 设置初始温度 $T(0)=T_0$ ，迭代次数 $i = 1$ ;
- 3) Do while  $T(i) > T_{min}$ 
  - 1) for  $j = 1 \sim k$
  - 2) 对当前最优解 $x_{best}$ 按照某一邻域函数，产生一新的解 $x_{new}$ 。计算新的目标函数值 $E(x_{new})$ ，并计算目标函数值的增量 $\Delta E = E(x_{new}) - E(x_{best})$ 。
  - 3) 如果 $\Delta E < 0$ ，则 $x_{best} = x_{new}$ ;
  - 4) 如果 $\Delta E > 0$ ，则 $p = \exp(-\Delta E / T(i))$ ;
    - 1) 如果 $c = \text{random}[0,1] < p$ ， $x_{best} = x_{new}$ ；否则  $x_{best} = x_{best}$ 。
  - 5) End for
- 4)  $i = i + 1$ ;
- 5) End Do
- 6) 输出当前最优点，计算结束。



# 模拟退火算法（要素）

## 1、状态空间与状态产生函数（邻域函数）

- 搜索空间也称为状态空间，它由经过编码的可行解的集合所组成。
- 状态产生函数(邻域函数)应尽可能保证产生的候选解遍布全部解空间。通常由两部分组成，即产生候选解的方式和候选解产生的概率分布。
- 候选解一般采用按照某一概率密度函数对解空间进行随机采样来获得。
- 概率分布可以是均匀分布、正态分布、指数分布等等。



## 模拟退火算法（要素）

### 2、状态转移概率（接受概率） $p$

- 状态转移概率是指从一个状态 $x_{old}$ (一个可行解)向另一个状态 $x_{new}$ (另一个可行解)的转移概率;
- 通俗的理解是接受一个新解为当前解的概率;
- 它与当前的温度参数 $T$ 有关, 随温度下降而减小。
- 一般采用**Metropolis准则**

$$p = \begin{cases} 1 & \text{if } E(x_{new}) < E(x_{old}) \\ \exp(-\frac{E(x_{new}) - E(x_{old})}{T}) & \text{if } E(x_{new}) \geq E(x_{old}) \end{cases}$$



## 模拟退火算法（要素）

### 3、冷却进度表 $T(t)$

冷却进度表是指从某一高温状态 $T_0$ 向低温状态冷却时的降温管理表。

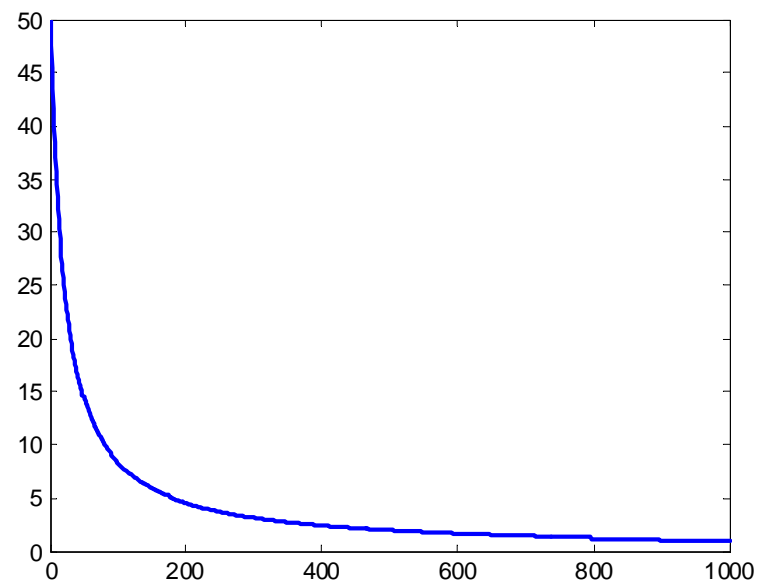
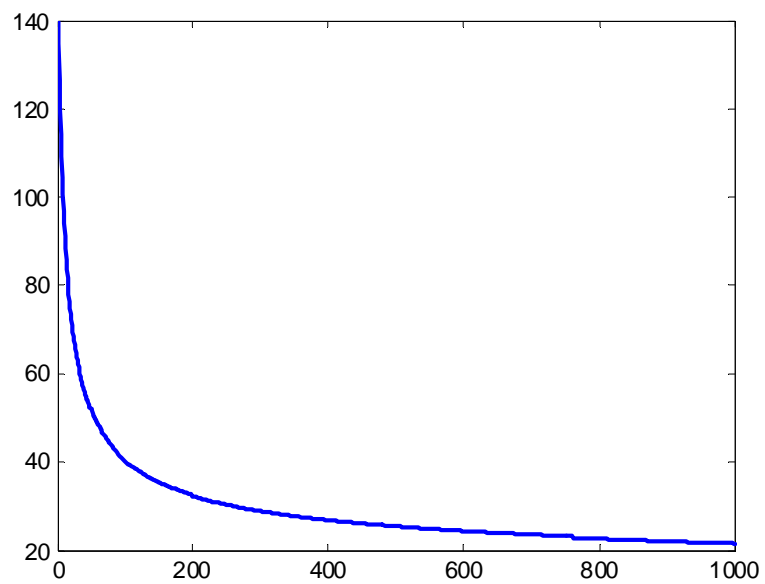
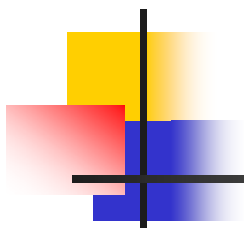
假设时刻 $t$ 的温度用 $T(t)$ 来表示，则经典模拟退火算法的降温方式为：

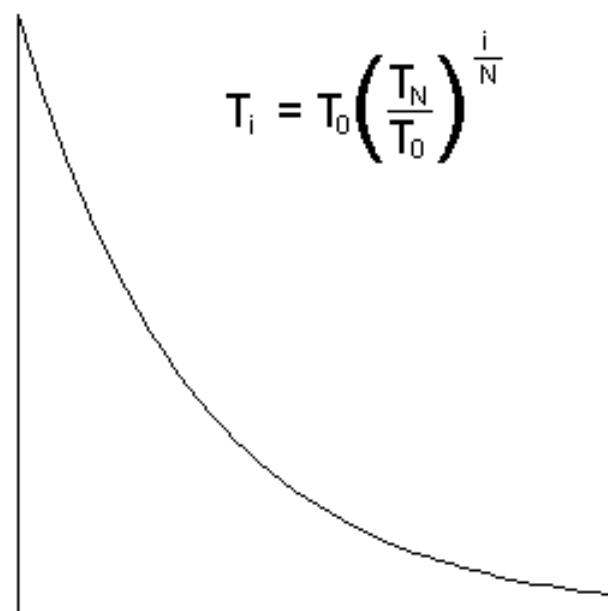
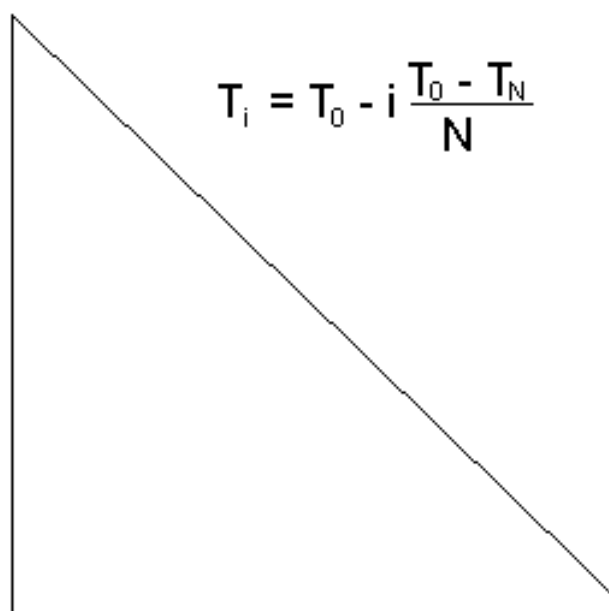
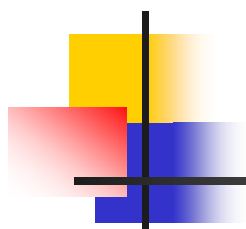
$$T(t) = \frac{T_0}{\lg(1+t)}$$

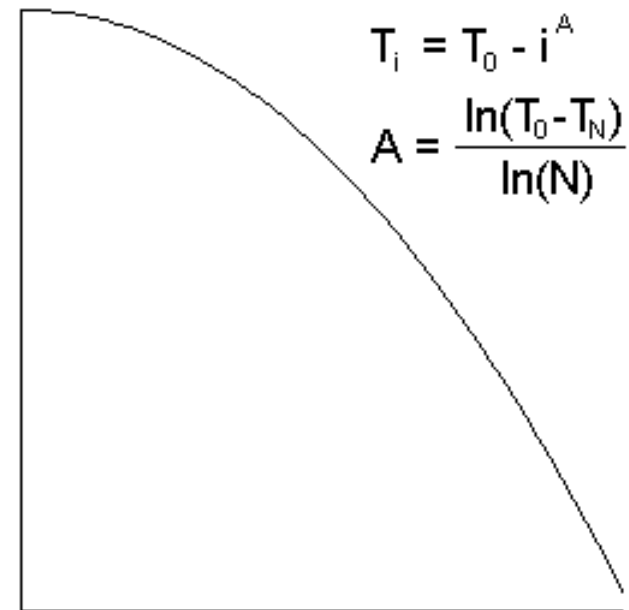
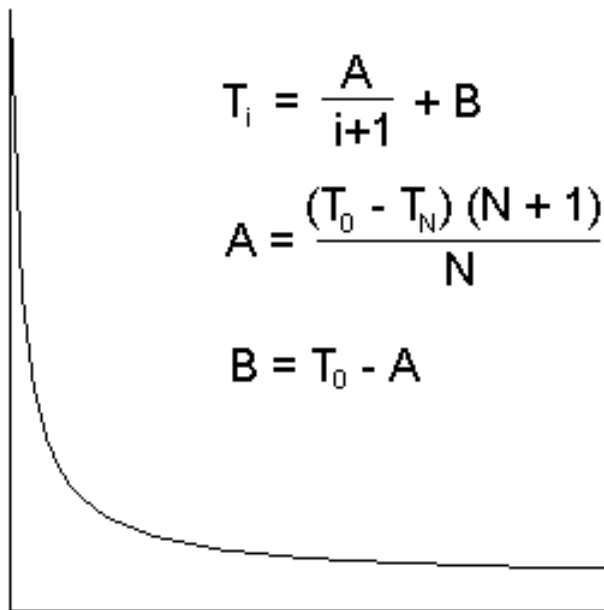
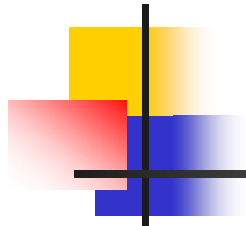
而快速模拟退火算法的降温方式为：

$$T(t) = \frac{T_0}{1+t}$$

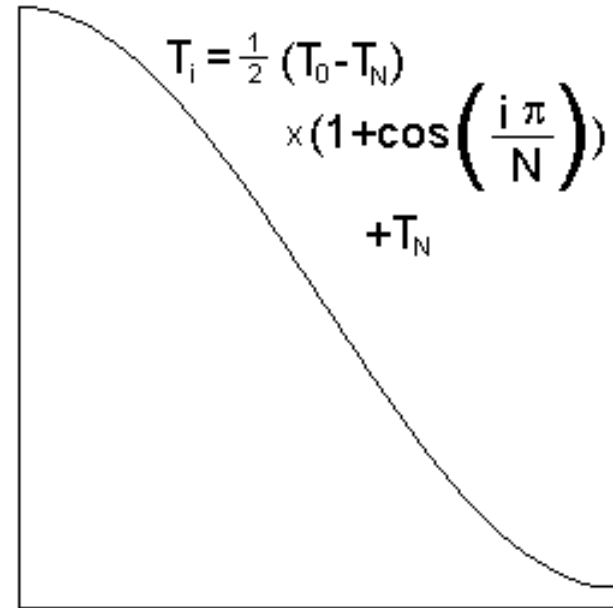
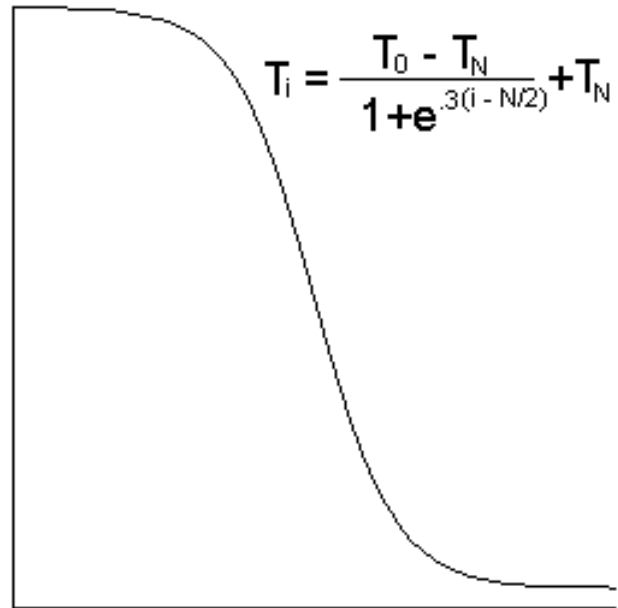
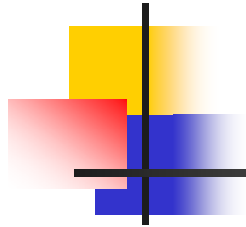
这两种方式都能够使得模拟退火算法收敛于全局最小点。

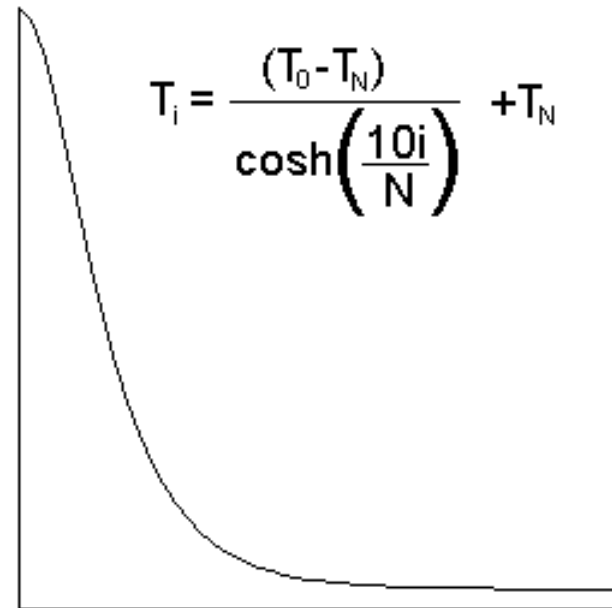
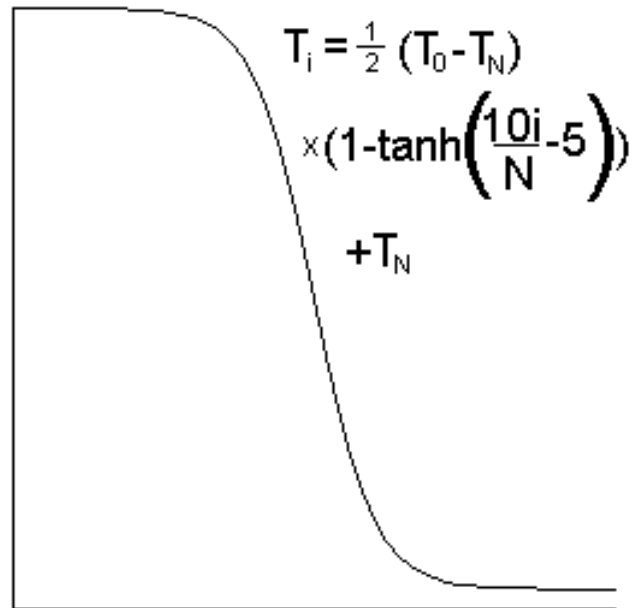
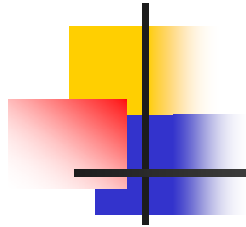


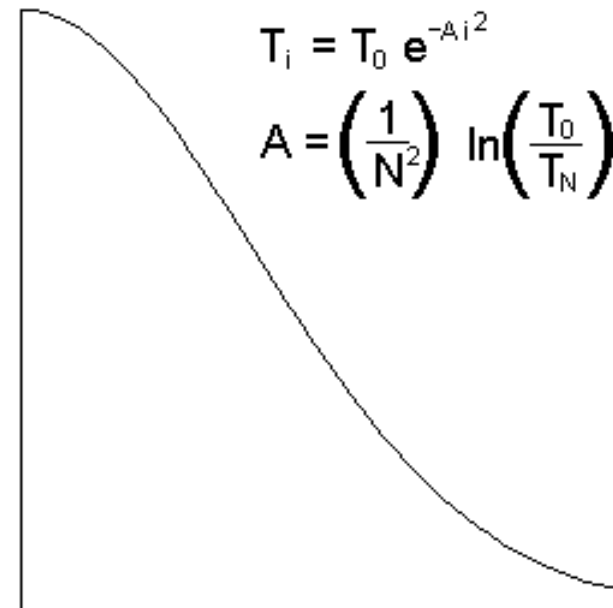
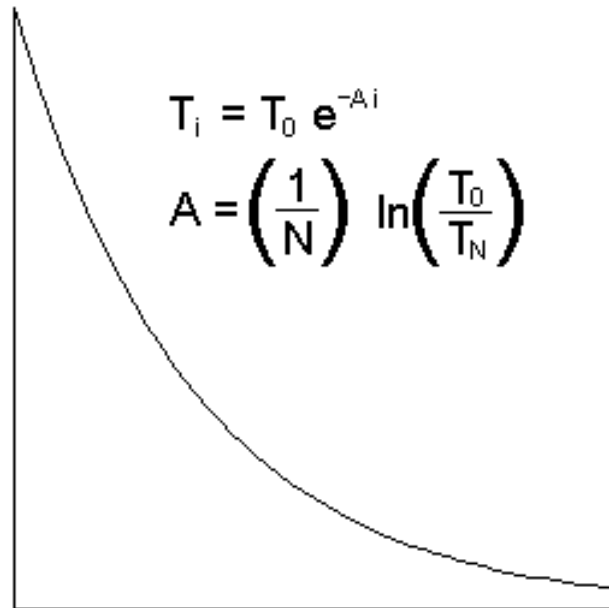
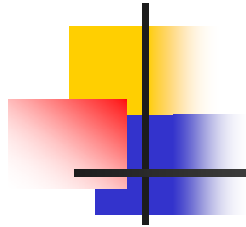














# 模拟退火算法（要素）

## 4、初始温度 $T_0$

实验表明，初温越大，获得高质量解的几率越大，但花费的计算时间将增加。因此，初温的确定应折衷考虑优化质量和优化效率，常用方法包括：

(1) 均匀抽样一组状态，以各状态目标值的方差为初温。

(2) 随机产生一组状态，确定两两状态间的最大目标值差 $|\Delta_{max}|$ ，然后依据差值，利用一定的函数确定初温。比如， $t_0 = -\Delta_{max}/p_r$ ，其中 $p_r$ 为初始接受概率。

(3) 利用经验公式给出。



## 模拟退火算法（要素）

---

### 5、内循环终止准则

或称Metropolis抽样稳定准则，用于决定在各温度下产生候选解的数目。常用的抽样稳定准则包括：

- (1) 检验目标函数的均值是否稳定；
- (2) 连续若干步的目标值变化较小；
- (3) 按一定的步数抽样。



## 模拟退火算法（要素）

---

### 6、外循环终止准则

即算法终止准则，常用的包括：

- (1) 设置终止温度的阈值；
- (2) 设置外循环迭代次数；
- (3) 算法搜索到的最优值连续若干步保持不变；
- (4) 检验系统熵是否稳定。



# 模拟退火算法的改进

---

- (1) 设计合适的状态产生函数，使其根据搜索进程的需要表现出状态的全空间分散性或局部区域性。
- (2) 设计高效的退火策略。
- (3) 避免状态的迂回搜索。
- (4) 采用并行搜索结构。
- (5) 为避免陷入局部极小，改进对温度的控制方式
- (6) 选择合适的初始状态。
- (7) 设计合适的算法终止准则。



# 模拟退火算法的改进

也可通过增加某些环节而实现对模拟退火算法的改进。主要的改进方式包括:

- (1) **增加升温或重升温过程**。在算法进程的适当时机, 将温度适当提高, 从而可激活各状态的接受概率, 以调整搜索进程中的当前状态, 避免算法在局部极小解处停滞不前。
- (2) **增加记忆功能**。为避免搜索过程中由于执行概率接受环节而遗失当前遇到的最优解, 可通过增加存储环节, 将“Best So Far”的状态记忆下来。
- (3) **增加补充搜索过程**。即在退火过程结束后, 以搜索到的最优解为初始状态, 再次执行模拟退火过程或局部性搜索。
- (4) 对每一当前状态, 采用多次搜索策略, 以概率接受区域内的最优状态, 而非标准SA的单次比较方式。
- (5) 结合其他搜索机制的算法, 如遗传算法、混沌搜索等。
- (6) 上述各方法的综合应用。





# 算法实现与应用

## ■ TSP问题的求解

- 编码（城市编号顺序编码）

- 状态产生函数（逆转算子）

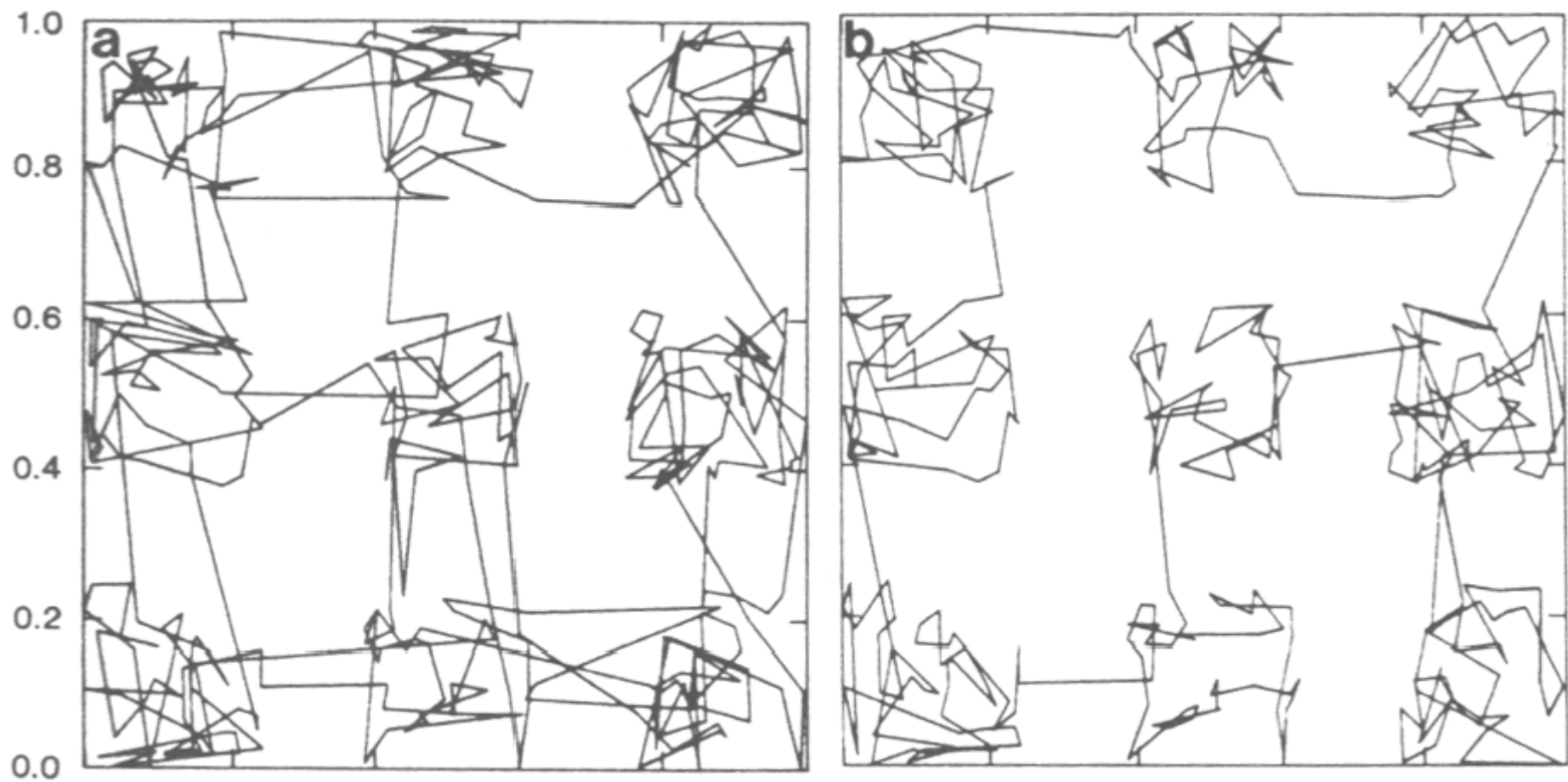
- 状态接受函数 
$$p = \begin{cases} 1 & \text{if } E(x_{new}) < E(x_{old}) \\ \exp(-\frac{E(x_{new}) - E(x_{old})}{T}) & \text{if } E(x_{new}) \geq E(x_{old}) \end{cases}$$

- 初温与初始状态,  $T_0 = -\Delta_{max}/p_r$

- 降温函数设计

- 温度修改准则和算法终止准则

# 结果（400个城市）



## 结果（400个城市）

