

Guide of Matlab

内容概要： 数学建模算法

创建时间： 2022/4/7 13:41

更新时间： 2022/4/17 15:27

作者： TwinkelStar

模拟退火算法

Simulated Annealing Algorithm

1、操作系统相关环境

1) 硬件环境：

➤ 电脑

2) 软件环境：

➤ Matlab2018a(程序设计软件)

3) 操作系统(2 选 1)：

➤ Windows7

➤ Windows10

2、模拟退火算法

模拟退火算法(Simulated Annealing, SA)的思想借鉴于固体的退火原理，当固体的温度很高的时候，内能比较大，固体的内部粒子处于快速无序运动，当温度慢慢降低的过程中，固体的内能减小，粒子的慢慢趋于有序，最终，当固体处于常温时，内能达到最小，此时，粒子最为稳定。模拟退火算法便是基于这样的原理设计而成。

1) 基本原理

模拟退火算法从某一较高的温度出发，这个温度称为初始温度，

伴随着温度参数的不断下降，算法中的解趋于稳定，但是，可能这样的稳定解是一个局部最优解，此时，模拟退火算法中会以一定的概率跳出这样的局部最优解，以寻找目标函数的全局最优解。如上图中所示，若此时寻找到了 A 点处的解，模拟退火算法会以一定的概率跳出这个解，如跳到了 D 点重新寻找，这样在一定程度上增加了寻找到全局最优解的可能性。

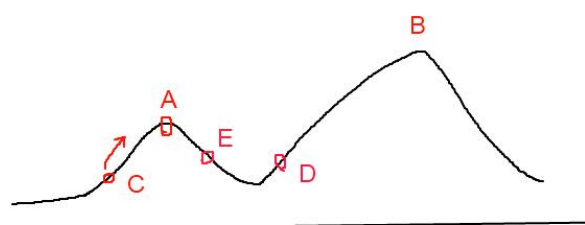


图 1.模拟退火算法寻找最优解
fig1. Simulated annealing algorithm to find the optimal solution

3、程序步骤：

①随机挑选一个单元 k ，并给它一个随机的位移，求出系统因此而产生的能量变化 ΔE ；

②若 $\Delta E_k \leq 0$ ，该位移可采纳，而变化后的系统状态可作为下次变化的起点；

若 $\Delta E_k > 0$ ，位移后的状态可采纳的概率为：

$$P_k = \frac{1}{1 + e^{-\Delta E_k / T}} \quad (1)$$

式 (1) 中 T 为温度，然后从 $(0, 1)$ 区间均匀分布的随机数中挑选一个数 R ，若 $R < P_k$ ，则将变化后的状态作为下次的起点；否

则，将变化前的状态作为下次的起点。

③转第①步继续执行，直到达到平衡状态为止。

程序框图如图 2 所示：

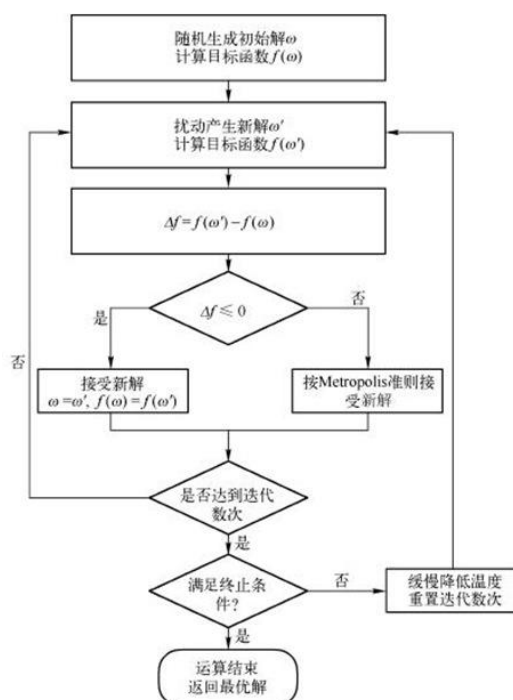


图 2.程序框图

fig2. Program flowchart

4、例题与程序设计

1) 例题

求解函数 $f(x) = (x - 2)^2 + 4$ ，其中 $-2 \leq x \leq 2$ 的最小值，其中，x 的取值范围为[-2, 2]。

2) 代码实现

编写目标函数值的函数代码，如图 3 所示：

```

%%Fx.m
function fx=Fx(x)
    fx=(x-2)^2+4;
end
  
```

图 3.目标函数代码

fig3. Objective function code

随机生成扰动 x_1 , x_2 , 如图 4 所示:

```
%%getX.m
function x=getX
    x=4*rand-2;
end
```

图 4. 目标函数代码

fig4. Objective function code

随机下降法则, 防止函数陷入局部最优, 如图 5 所示:

```
%%getP.m
function p=getP(c,t)
    p=exp(-c/t);
end
```

图 5. 随机下降法则

fig5. Random descent rule

主函数代码如图 6 所示:

```
2      %%main.m
3      T=1000; %初始化温度值
4      T_min=1e-12; %设置温度下界
5      alpha=0.98; %温度的下降率
6      k=1000; %迭代次数 (解空间的大小)
7
8      x=getX; %随机得到初始解
9      while(T>T_min)
10         for I=1:100
11             fx=Fx(x);
12             x_new=getX;
13             if(x_new>=-2 && x_new<=2)
14                 fx_new=Fx(x_new);
15                 delta=fx_new-fx;
16                 if (delta<0)
17                     x=x_new+(2*rand-1);
18                 else
19                     P=getP(delta,T);
20                     if(P>rand)
21                         x=x_new;
22                     end
23                 end
24             end
25         end
26         T=T*alpha;
27     end
28     disp('最优解为: ')
29     disp(x)
```

图 6. 主函数

fig6. Main

函数符号说明如表 1 所示：

表 1.符号描述图表
table1.Symbol Description

Symbol	Description
T	初始温度
T_min	设置温度下界
alpha	降温系数
k	迭代次数

3) 代码调用

运行代码 sa.m 直接得出结果，函数返回结果如图 8 所示：

```
>> sa
最优解为：
1.8837
```

图 8.函数返回结果
fig8. Function returns result

从结果中可知，结果 1000 次的迭代之后，函数值收敛。

函数的最优解为 1.9548，x 的取值为-0.7019。