# 模拟退火算法解决背包问题

代码：

clear;

clc;

Price=[5;10;13;4;3;11;13;10;8;16;7;4];

Weight=[2;5;18;3;2;5;10;4;11;7;14;6];

MaxWeight=45;%背包最大重量

N=size(Price,1);%物品个数(求列的长度)

Num=1:N;

Goods=zeros(N,4);

Goods=[Num' Price Weight];

%说明：第一列为物品序号，第二列为价格，第三列为重量

% 第四列为每单位重量多少钱

T0=100;

Tf=0.1;

lambda=0.95;

Markov\_length=100;%任意温度的迭代次数

%模拟退火算法初始参数设定

T\_now=T0;

%产生初始解

first\_sol=ones(1,N);%产生初始解

now\_sol=first\_sol;

best\_sol=zeros(1,N);

now\_sol\_val=now\_sol\*Goods(:,2);

best\_sol\_val=0;

%1表示拿，0表示不拿

%开始退火

p=1;

while T\_now>=Tf

for num=1:Markov\_length

%产生新解

%方法：随机扰动12个1

rnd=ceil(unifrnd(0.5,N-0.5));

new\_sol=now\_sol;

new\_sol(rnd)=~new\_sol(rnd);

while 1

q=(new\_sol\*Goods(:,3)<=MaxWeight);

if q==0

p=~p;

rnd=find(new\_sol==1);

if p

new\_sol(rnd)=0;

else

new\_sol(rnd(end))=0;

end

else

break;

end

end

%产生新解完毕

new\_sol\_val=new\_sol\*Goods(:,2);

if new\_sol\_val>now\_sol\_val

%新解大，无条件接受

now\_sol=new\_sol;

now\_sol\_val=new\_sol\_val;

if now\_sol\_val>best\_sol\_val

%记录最好解

best\_sol=now\_sol;

best\_sol\_val=now\_sol\_val;

end

else

%新解小，选择性接受

E=exp(-abs(new\_sol\_val-now\_sol\_val)/T\_now);

rnd=unifrnd(0,1);

if rnd<E

%选择性接受

now\_sol=new\_sol;

now\_sol\_val=new\_sol\_val;

else

%不接受

new\_sol=now\_sol;

end

end

end

T\_now=T\_now\*lambda;

end

%输出结果：

disp('最优解为：');

disp(best\_sol);

disp('最优解的值为：');

disp(best\_sol\_val);

disp('背包的重量为：');

disp(best\_sol\*Goods(:,3));