

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));
```