

用遗传算法解决背包问题

2021.03.13

519021910418

王山木

一、问题描述：

有 16 件物品，各有不同的体积、重量和价值，具体如图 1 所示。要求选择其中若干件放入一个背包，使放入的物品总价值尽量高。该背包容量不超过 95（体积单位），可承受重量不超过 86（重量单位）。

物品	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
体积	7	4	8	11	20	5	3	9	16	7	8	5	4	4	3	12
重量	11	7	9	6	7	8	5	6	18	2	3	6	2	9	5	4
价值	9	8	7	8	18	7	3	10	19	4	4	11	3	5	4	6

图 1

二、求解结果：

根据 Matlab 仿真，选取第 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15 件物品放入背包时可使放入的物品总价值最高，为 105；此时对应的体积为 95，重量为 85。

三、Matlab 原始结果：

Matlab 原始运行结果如图 2，其中 1 表示选取对应位置的物品，0 表示不选取。图 3 为优化过程中总价值变化曲线。

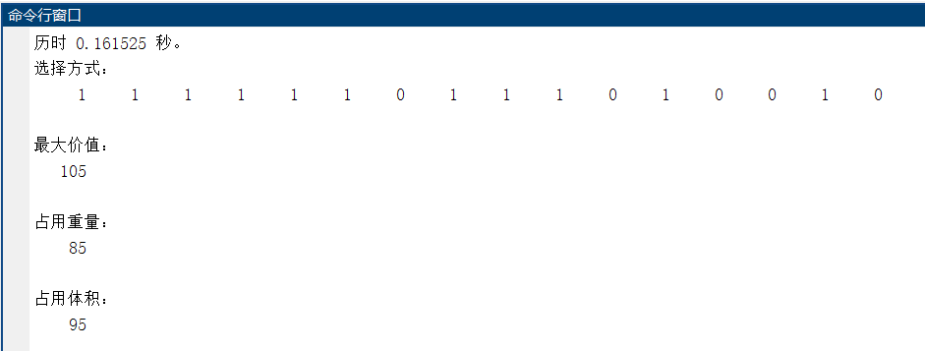


图 2

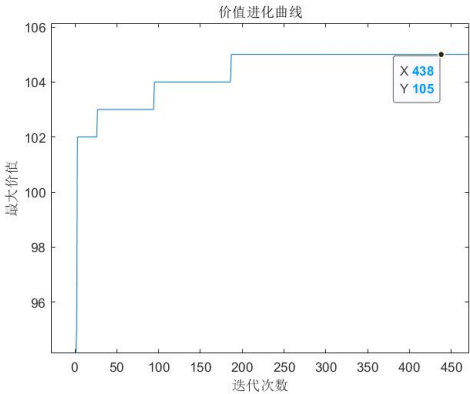


图 3

四、Matlab 原始代码：

%%代码功能：运用遗传算法 (GA) 解决背包问题

%%学号：519021910418

%%姓名：王山木

clear;

clc;

NP = 100; %种群规模

p1 = 0.8; %交叉概率（查阅资料：取 0.25~1 为宜）

p2 = 0.05; %变异概率（查阅资料：取 0.001~0.1 为宜）

N = 500; %最大迭代次数

max_size = 95; %背包最大承受体积

max_weight = 86; %背包最大承受重量

weight = [11 7 9 6 7 8 5 6 18 2 3 6 2 9 5 4]; %重量（要求取得小于 86）

size = [7 4 8 11 20 5 3 9 16 7 8 5 4 4 3 12]; %体积（要求取得小于 95）

value = [9 8 7 8 18 7 3 10 19 4 4 11 3 5 4 6]; %价值（要求最大化）

L = length(weight); %表明要选取的总物品的个数，用 1 代表选取之，0 代表不选取

f = randi([0,1],NP,L); %随机获得 NP 个初始种群

 %randi 函数：生成 NP*L 矩阵，元素为 0, 1 中随机一个

%%将初始种群中不符合要求的个体去掉

for i = 1:NP

 while ((f(i,:) * weight' > max_weight) || (f(i,:) * size' > max_size))

 f(i,:) = randi([0,1],1,L);

 end

end

tic

%%遗传算法的 N 轮循环

for k = 1:N

 %计算每一个个体的适应度，函数为 suitability

 for i = 1:NP

 fit(i) = suitability(f(i,:),size,weight,value,max_size,max_weight);

 end

```

maxfit = max(fit); %获得最大适应度
minfit = min(fit); %获得最小适应度

location = find(fit == maxfit); %最优个体的位置
fbest = f(location(1,1),:); %历代最优个体

%%轮盘赌决定去留
sum_fit = sum(fit);
fitvalue = fit./sum_fit;
fitvalue = cumsum(fitvalue); %这里的第 n 个元素是初始 fitvalue 中
                             %前 n 个元素的累加
change_p = sort(rand(NP,1)); %生成 NP*1 的矩阵,
                             %数值为 0~1 随机并从小到大排好序, 用作概率

fiti = 1;
newi = 1;
while newi <= NP
    if(change_p(newi) < fitvalue(fiti)) %说明落在该区域, 留下 fiti 对应的个体
        new_f(newi,:) = f(fiti,:);
        newi = newi + 1; %newi + 1,直至筛选出 NP 个新的个体
    else %概率不在 fiti 指示的个体内, 该个体被筛掉
        fiti = fiti + 1;
    end
end

%%基因交叉操作
for i = 1:2:NP %隔一个是保证有一条基因能与之交叉
    p = rand;
    if p < p1 %说明满足了交叉的概率
        q = randi([0,1],1,L); %随机生成哪一个基因要被换
        for j = 1:L
            if q(j) == 1 %为 1 则交换该位置 i 和 i+1 对应的基因
                temp = new_f(i+1,j);
                new_f(i+1,j) = new_f(i,j);
                new_f(i,j) = temp;
            end
        end
    end
end
end
end

```

```

%%基因变异操作
for m = 1:NP
    for n = 1:L
        r = rand;
        if r < p2 %说明满足变异概率
            new_f(m,n) = ~new_f(m,n); %直接取反，设为变异
        end
    end
end

f = new_f;
f(1,:) = fbest;%人为保留上一轮最佳个体，防止因意外导致该个体被淘汰/往差的方向变异
maxvalue(k) = maxfit;
end

toc
disp('选择方式: ')
disp(fbest)
disp('最大价值: ')
disp(fbest*value')
disp('占用重量: ')
disp(fbest*weight')
disp('占用体积: ')
disp(fbest*size')
figure
plot(maxvalue)
xlabel('迭代次数')
ylabel('最大价值')
title('价值进化曲线')

%%计算适应度的函数代码:
function result = suitability(f,size,weight,value,max_size,max_weight)
%该函数用于计算某一个个体的适应度，输入参数依次为
%f:个体的基因（即选择方式）
%size:每件物品对应的体积    weight:每件物品对应的重量    value:每件物品对应的价值
%max_size:背包最大承载体积    max_weight:背包最大承受重量
%返回一个结果，结果代表着当前个体对应的适应度（即装入物品总 value 值）

fit = sum(f.*value);
totalsize = sum(f.*size);
totalweight = f * weight';

```

```

if ((totalweight <= max_weight)&&(totalsize <= max_size))
%如果满足容量条件，则适应度就是总价值
    fit = fit;
else
    if (totalsize > max_size)
        %如果不满足体积条件，则令其适应度变小，方式为减去超出体积部分的 20 倍
        %操作的目的是使得超出体积限度的个体的适应度大大减小，从而能被更好的筛掉
        %因此此处系数 20 应根据实际数据进行调整，实测过低时，无法体现出超过容量限制对其的影响之大
        fit = fit - 20 * (totalsize - max_size);
    else %同理
        fit = fit - 20 * (totalweight - max_weight);
    end
end

result = fit;

```