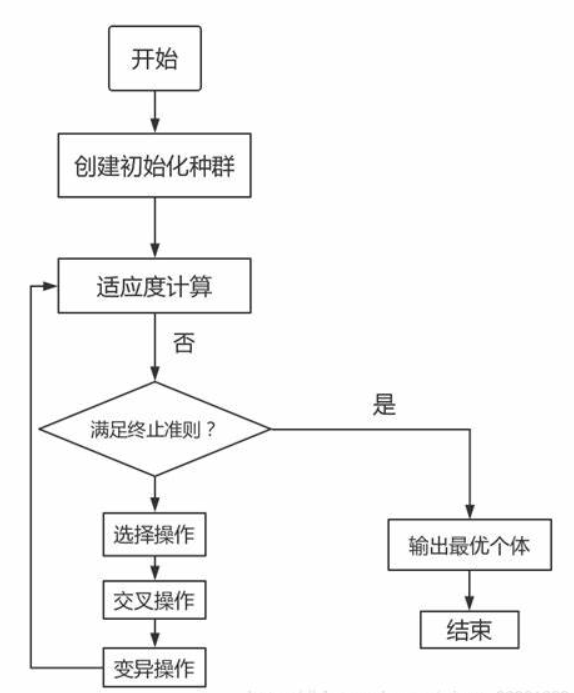
用遗传算法求解 f(x)=10\*sin(5x)+7\*cos(4x)，在x∈[0,10]上的最大值。给出详细的分析求解过程，并附上程序代码（不限编程语言）。

解：遗传算法流程图如下：

（1）确定遗传算法的运行参数

群体大小：N=50；终止代数：G=100

交叉概率：Pc=0.8；变异概率：Pm=0.1

区间上限：xs=10; 区间下限：xx=0

（2）确定编码方法

个体编码方法有二进制编码和实数编码，在解决三角函数最大值求解问题选择个体编码方法为二进制编码，用一个20位的二进制数表示x，则x=0+10\*b/(2^20), 其中 b 是 [0,2^20] 中的一个二值数。故二进制长度L=20。

（3）确定适应度函数

由于需要求最大值，所以可以直接将目标函数作为适应度函数，

对适应度函数进行归一化：



（4）选择操作

按适应度比例分配：目前遗传算法中最基本且最常用的算法，各个个体被选择的概率和其适应度值成比例。设群体规模大小为N，个体i的适应度值为f，则这个个体被选择的概率为



（5）交叉操作

单点交叉：先对群体进行随机配对，其次随机设置交叉点位置，最后再相互交换配对染色体之间的部分基因。

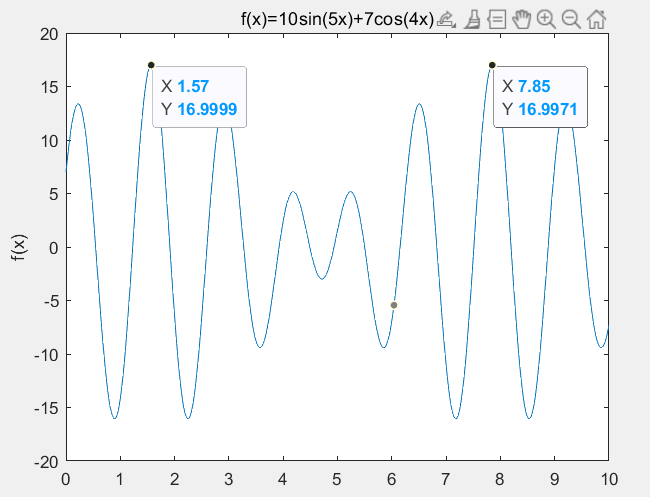
（6）变异操作

基本位变异：首先确定变异的染色体，然后确定染色体变异位置，依照某一概率将变异点的原有基因值取反，即0->1，1->0。

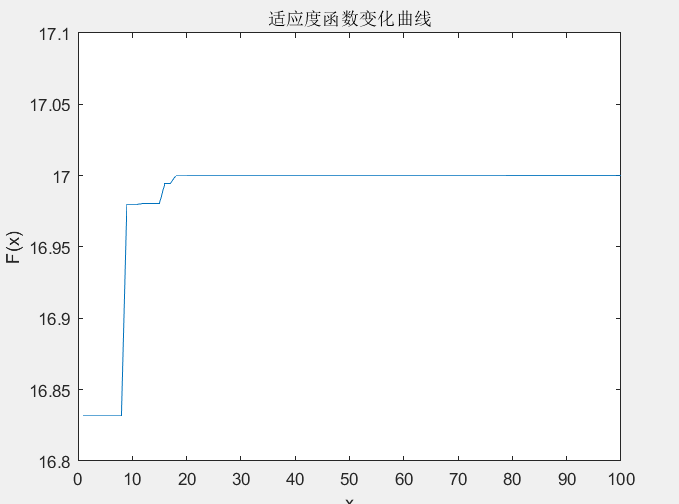
（7）求解结果

使用遗传算法求解，函数f(x)=10sin(5x)+7cos(4x)有两个相等的最大值，为17，对应坐标的位置为（1.5709，17），（7.8541，17）。



 画出函数的图像如下，与遗传算法求解结果相同：

适应度函数变化曲线如下：

代码如下：

clc,clear

N = 50;%种群数量

L = 20;%二进制位串长度

pc =0.8;%交叉率

pm =0.1;%变异率

g =100;%最大遗传代数

xs =10;%上限

xx =0;%下限

f = round(rand(N,L));%随机初始化种群（二维数组）

%遗传算法循环

for k =1:g

%将二进制解码为定义域范围内十进制

for i = 1:N

u =f (i,:);

m = 0;

for j =1:L

m = u(j)\*2^(j-1)+m;%求出该个体的二进制编码的十进制数

end

x(i) = xx + m\*(xs-xx)/(2^L-1);%映射到所要求的自变量区间上

fit(i) = func1(x(i));%调用适应度函数

end

maxfit = max(fit);%定义适应度中的最大值

minfit = min(fit);%定义适应度中的最小值

rr = find(fit==maxfit);

fbest = f(rr(1,1),:);%得到第k代最优个体的染色体编码数组

xbest = x(rr(1,1));%得到最优个体对应十进制映射数值

fit = (fit-minfit)/(maxfit-minfit);%归一化适应度函数值

%基于轮盘赌的复制操作

sum\_fit = sum(fit);%适应度函数的和

fitvaule = fit./sum\_fit;%计算概率

fitvaule = cumsum(fitvaule);%计算累加概率

ms = sort(rand(N,1));%随机生成N行1列在[0，1]范围内的列向量并升序排列

fiti = 1;%原种群当前被比较的个体序号

newi = 1;%被选择进入下一代的个体序号

while newi <= N

if (ms(newi)) < fitvaule(fiti)

nf(newi,:) = f(fiti,:);

newi = newi +1;

else

fiti = fiti +1;

end

end

%基于概率的交叉操作

for i = 1:2:N %选择两个优秀个体

p = rand;%生成[0,1]的随机小数

if p <pc%若p小于交叉概率

q = round(rand(1,L));%生成一条（0，1）分布的二进制数串

for j =1:L

if q(j) == 1%如果第j位上的值为1，则进行第i组个体的第j位交叉操作

temp = nf(i+1,j);

nf(i+1,j) = nf(i,j);

nf(i,j) = temp;%完成第j位交叉互换

end

end

end

end

%基于概率的变异操作

i =1;

while i <= round(N\*pm)%四舍五入取整

h = randi(N);%随机选取一条需要变异的染色体

for j = 1:round(L\*pm)%在需要变异的染色体上进行L\*pm个记忆变异

g = randi(L);%随机选择变异的基因序号

nf(h,g) = ~nf(h,g);%取反完成变异

end

i = i+1;

end

f = nf;

f(1,:) =fbest;

value(k) = maxfit;

end

disp(['最终函数最大值点为',num2str(xbest)])

disp(['最终函数最大值为',num2str(func1(xbest))])

figure(1);

xy = 0:0.01:10;

y = 10\*sin(5\*xy)+7\*cos(4\*xy);

plot(xy,y)

xlabel('x')

ylabel('f(x)')

title('f(x)=10sin(5x)+7cos(4x)')

figure(2);

plot(value)

xlabel('x')

ylabel('F(x)')

title('适应度函数变化曲线')

axis([0 100 16.8 17.1])

function result = func1(x)

fit = 10\*sin(5\*x)+7\*cos(4\*x);

result = fit;

end