遗传算法

一种基于生物遗传和进化机制的适合于复杂系统优化的自适应概率优化算法。

特点：

不依赖于梯度信息，不仅不受目标函数连续可微的约束，还可以通过编码来实现任意设定其定义域。

优点：

1.不是从单个点，而是从多个点构成的群体开始搜索

2.搜索过程不易陷入局部最优点

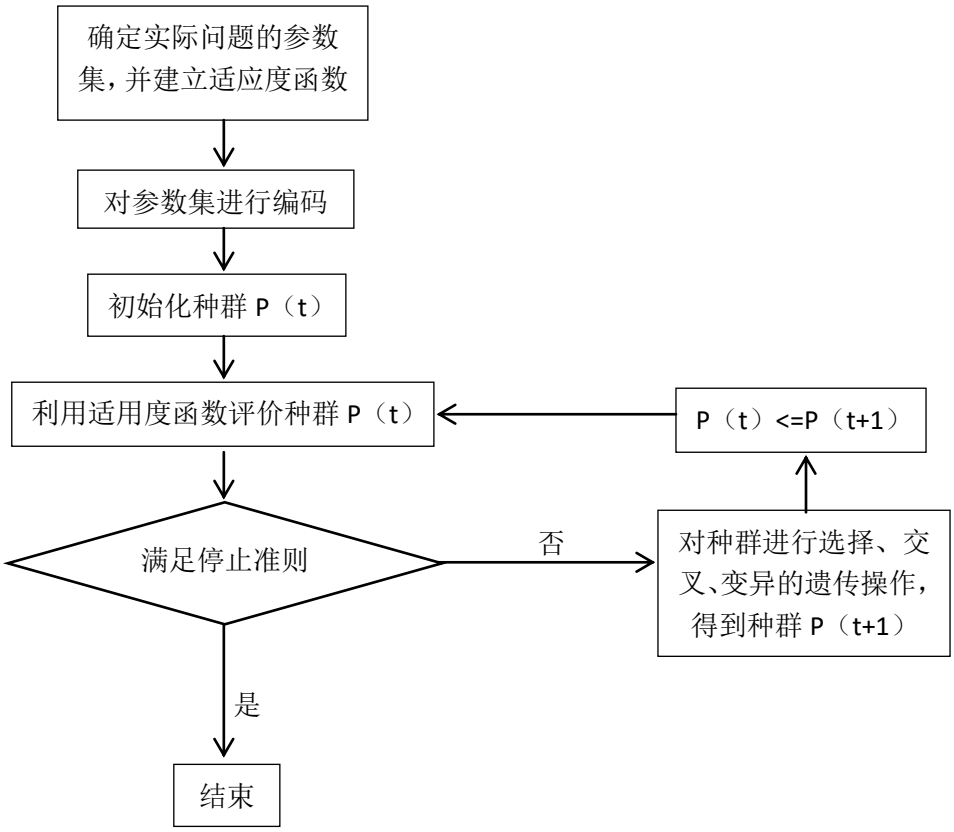
基本理论：

模仿生物的进化过程，模拟了自然选择和遗传中发生的复制、交叉和变异，对种群反复进行选择、交叉以及变异操作，估计各个个体的适应值，根据“适者生存、优胜劣汰”的进化规则，使得群体越来越向最优解的方向进化。这里的最优解可以是复杂函数的最大/小值、最短路程、最大利润等。

生物进化	遗传算法
适者生存，优胜劣汰	适应值函数越大的解被保留的概率大
个体	问题的一个可行解
染色体	可行解的编码
基因	编码的元素
群体	被选定的一组可行解
种群	根据适应值函数选择的一组可行解
选择	优胜劣汰
交叉	以交叉方式由双亲产生后代的过程
变异	编码的某些分量发生变化的过程

选择算子的作用：提高了群体的平均值。由于其没有产生新个体，所以群体中最好个体的适应值不会因选择操作而有所改进。

交叉、变异算子的作用：变异是对个体的某一个或某一些基因值按某一较小概率进行改变。从产生新个体的能力方面来说，交叉算子是产生新个体的主要方法，他决定了遗传算法的全局搜索能力，而变异算子只产生新个体的辅助方法，但也必不可少，因为它决定了遗传算法的局部搜索能力。



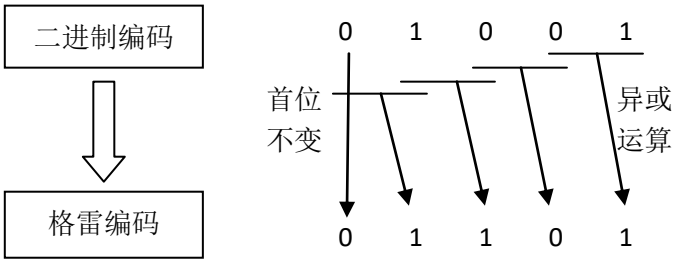
一般种群数目为 20-100
交叉概率一般为 0.4-0.99
变异概率为 0.0001-0.1

建模步骤：

1.编码

普通的二进制编码方式可能具有较大的汉明距离，采用格雷编码可避免这一缺陷。

（格雷编码是一种基于二进制编码的循环码，它需要对普通的二进制编码从最后一位起，依次将每一位与其左边一位作异运算，作为对应格雷编码在该位的值，最左边一位保持不变）



设变量为连续变量，染色体长度与设计变量的维数相同。

设计变量为：

$$X = [x_1, x_2, \ldots, x_n]$$

染色体为：

$$V_k = [v_{k1}, v_{k2}, \ldots, v_{kn}]$$

式中，x_1^u、x_i^v 分别为设计变量 x_i 的下限和上限，$x_i^u \le v_{ki} \le x_i^v$，m 为染色体的总数，称为种群规模，初始种群用随机方法产生。

2.适应度函数（以下例举四种）

①针对最小化问题，令

$$Fitness(f(X)) = \begin{cases} f(X) - C_{\min} & , f(X) > C_{\min} \\ 0 & , others \end{cases}$$

$C_{\min}$为最小估计值

②针对最大化问题，令

$$Fitness(f(X)) = \begin{cases} C_{\max} - f(X) & , f(X) < C_{\max} \\ 0 & , others \end{cases}$$

$C_{\max}$为最大估计值

③对于非线性规划问题：

$$\min \quad f(X)$$
$$s.t. \quad g_i(X) \geq 0(i = 1,2,\ldots,m)$$

④外点法构造惩罚函数：

$$\phi(X, r^{(k)}) = f(X) + r^{(k)}\left\{\sum_{i=1}^{m} \min[g_u(X),0]^2\right\}$$

式中，$r^{(k)}$为惩罚因子，是一个单调递增增正值序列，$r^{(k+1)} = er^{(k)}$，许多计算经验表明，若取$r^{(0)} = 1$和$e \in [5,10]$，可取得满意结果。

3.选择算子

在对个体的适应度进行评价的基础上，通过选择操作把优化的个体直接遗传到下一代，或通过配对交叉产生新的个体再遗传到下一代。设群体规模为 m，个体 i 的适应度为 F_i，则个体 i 被选中的概率 P_{is} 为

$$P_{is} = F_i / \sum_{i=1}^{m} F_i$$

4.交叉算子

定义交叉操作的概率 P_c，一般建议取值范围为 0.4-0.99.然后按概率 P_c 把两个父代个体的部分结构加以交换重组而产生新个体。一般用浮点数编码方法表示个体在进行交叉时一般是进行算术交叉。在这里可以结合格雷编码来优化下。假设在两个个体 X_A^t、X_B^t之间进行算数交叉，则交叉运算后产生的两个新个体是

$$\begin{cases} X_A^{t+1} = aX_B^t + (1-a)X_A^t \\ X_B^{t+1} = aX_A^t + (1-a)X_B^t \end{cases}$$

式中，a 为交叉参数，$a \in (0,1)$。它可以是一个常数，此时称均匀算数交叉；它也可以是一个由进化代数所决定的变量，此时称非均匀算数交叉。

5.变异算子

定义参数 P_m 作为变异操作的概率，建议取值范围为 0.0001-0.1.采用非均匀变异：折个体$X = x_1, x_2 \ldots x_k \ldots x_i$，若$x_k$为变异点，其取值范围为[$U_{min}$，$U_{max}$]，在该点对个体 X 进行变异后，可得到一个新个体$X = x_1, x_2 \ldots x_k' \ldots x_i$，其中变异点的新基因值为：

$$x_k' = \begin{cases} x_k + (U_{\max}^k - x_k)r^{(1-G/T)b}, Random(0,1) = 1 \\ x_k + (x_k - U_{\min}^k)r^{(1-G/T)b}, Random(0,1) = 0 \end{cases}$$

式中，$Random(0,1)$表示以一定的概率从 0，1 中随机取的一个；r 为[0，1]范围内

符合均匀分布的一个随机数，即为 *Random*(0,1)；G 为当前代数；T 为终止代数；b 为调整变异步长的参数，随进化代数 G 而动态变化。

例题

①一元函数的优化问题

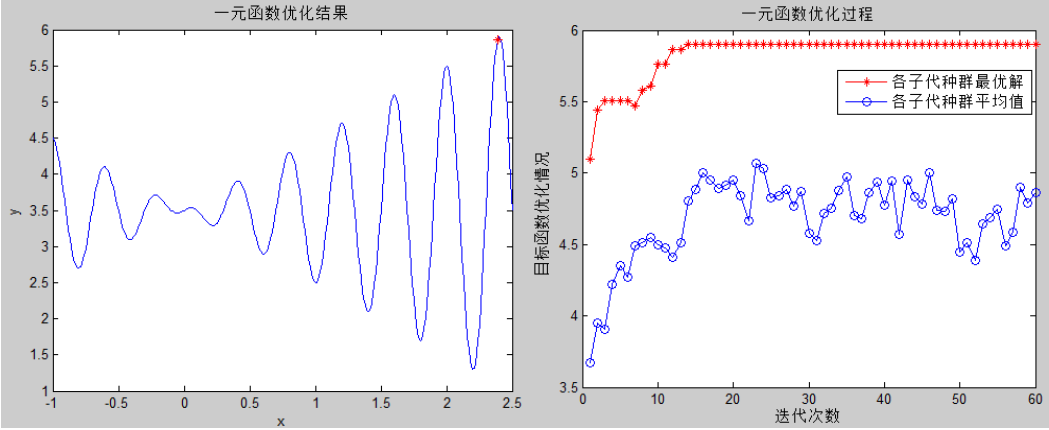
例：利用遗传算法计算函数 $f(x)=x\cos(5\pi x)+3.5$ 在区间$[-1,2.5]$上的最值。

程序：一元函数优化	运行环境 2011a

```
opt_minmax=-1;        %目标优化类型：1 最大化、-1 最小化
num_ppu=50;           %种群规模：个体个数
num_gen=60;           %最大遗传代数
len_ch=20;            %基因长度
gap=0.9;              %代沟
sub=-1;               %变量取值下限
up=2.5;               %变量取值上限
cd_gray=1;            %是否选择格雷编码方式：1 是 0 否
sc_log=0;             %是否选择对数标度：1 是 0 否
trace=zeros(num_gen,2);    %遗传迭代性能跟踪器，生成 60 行 2 列 0 矩阵
fieldd=[len_ch;sub;up;1-cd_gray;sc_log;1;1];    %区域描述器
chrom=crtbp(num_ppu,len_ch); %初始化生成种群，生成一个 50*20 的矩阵，矩阵元素是 0-1 随机数
k_gen=0;%初始化遗传次数
x=bs2rv(chrom,fieldd);                    %翻译初始化种群为 10 进制
fun_v=fun_sigv(x);                        %计算目标函数值
tx=sub:.01:up;
plot(tx,fun_sigv(tx))
xlabel('x')
ylabel('y')
title('一元函数优化结果')
hold on
while k_gen<num_gen
    fit_v=ranking(-opt_minmax*fun_v);          %计算目标函数的适应度
%ranking 函数为查询结果数据集分区中的每一行，并返回一个序列值。依据此函数，一些可能行可能取得和其他行一样的序列值
    selchrom=select('rws',chrom,fit_v,gap);   %使用轮盘度方式选择
    selchrom=recombin('xovsp',selchrom);      %交叉
    selchrom=mut(selchrom);                   %变异
    x=bs2rv(selchrom,fieldd);                 %子代个体翻译
    fun_v_sel=fun_sigv(x);                    %计算子代个体对应目标函数值
[chrom,fun_v]=reins(chrom,selchrom,1,1,opt_minmax*fun_v,opt_minmax*fun_v_sel);    %根据目标函数值将子代个体插入新种群
    [f,id]=max(fun_v);                        %寻找当前种群最优解
     x=bs2rv(chrom,fieldd);                   %翻译初始化种群为 10 进制
     f=f*opt_minmax;
     fun_v=fun_v*opt_minmax;
    k_gen=k_gen+1;%记录遗传次数
    trace(k_gen,1)=f;
    trace(k_gen,2)=mean(fun_v);
end
plot(x(id),f,'r*')
figure
```

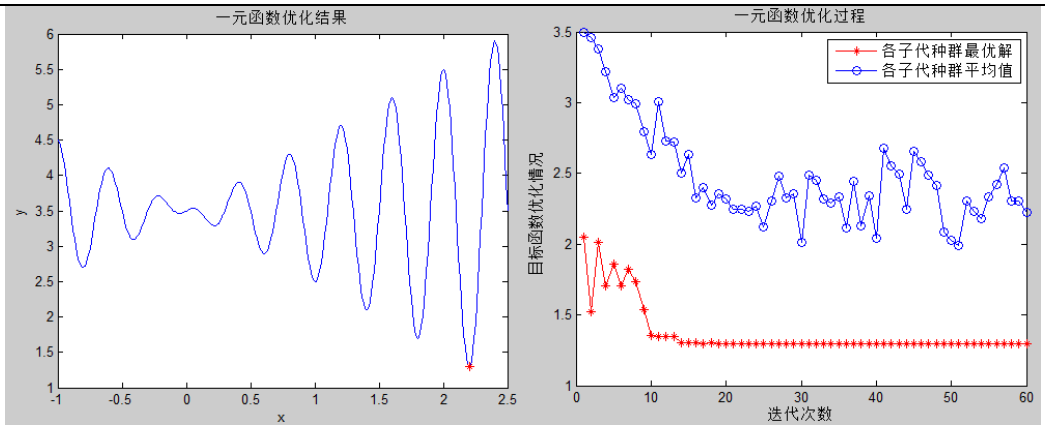
```
plot(trace(:,1),'r-*')
hold on
plot(trace(:,2),'b-o')
legend('各子代种群最优解','各子代种群平均值')
xlabel('迭代次数')
ylabel('目标函数优化情况')
title('一元函数优化过程')
%---------------------------------------------------------------------
function y=fun_sigv(x)
y=x.*cos(5*pi*x)+3.5;
```

求得最大值



结果看表格 x 的最后一个（自变量）和表格 trace 第一列最后一个（函数值）

求最小值



结果看表格 x 的最后一个（自变量）和表格 trace 第一列最后一个（函数值）

②多元函数的优化问题

例：利用遗传算法求解多峰的 Shubert 函数 $f(x,y)$ 在区域[−10,10]×[−10,10]上的最值，其中 $f(x,y)=\sum_{k=1}^{5}k\cos\left[(k+1)x+k\right]\text{g}\sum_{k=1}^{5}k\cos\left[(k+1)y+k\right]$。

程序：多元函数优化	运行环境 2014a

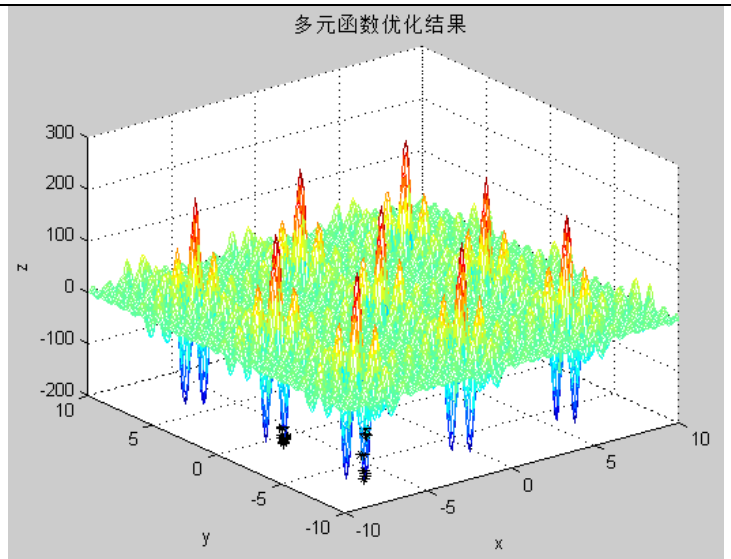
```
clc
clear all
opt_minmax=-1;     %目标优化类型：1 最大化、-1 最小化
num_ppu=60;        %种群规模：个体个数
num_gen=100;       %最大遗传代数
num_v=2;           %变量个数
```

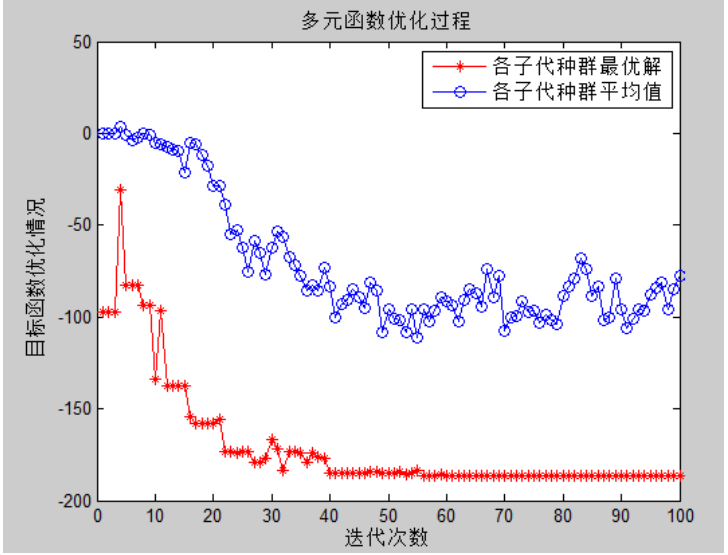
```
len_ch=20;          %基因长度
gap=0.9;            %代沟
sub=-10;            %变量取值下限
up=10;              %变量取值上限
cd_gray=1;          %是否选择格雷编码方式：1 是 0 否
sc_log=0;           %是否选择对数标度：1 是 0 否
trace=zeros(num_gen,2);     %遗传迭代性能跟踪器
fieldd=[rep([len_ch],[1,num_v]);rep([sub;up],[1,num_v]);rep([1-cd_gray;sc_log;1;1],[1,num_v])];
    %区域描述器
chrom=crtbp(num_ppu,len_ch*num_v);                %初始化生成种群
k_gen=0;
x=bs2rv(chrom,fieldd);                            %翻译初始化种群为 10 进制
fun_v=fun_mutv(x(:,1),x(:,2));                    %计算目标函数值
[tx,ty]=meshgrid(-10:.1:10);
mesh(tx,ty,fun_mutv(tx,ty))%画三维图
xlabel('x')
ylabel('y')
zlabel('z')
title('多元函数优化结果')
hold on
while k_gen<num_gen
      fit_v=ranking(-opt_minmax*fun_v);           %计算目标函数的适应度
      selchrom=select('rws',chrom,fit_v,gap);     %使用轮盘度方式选择
      selchrom=recombin('xovsp',selchrom);        %交叉
      selchrom=mut(selchrom);                     %变异
      x=bs2rv(selchrom,fieldd);                   %子代个体翻译
      fun_v_sel=fun_mutv(x(:,1),x(:,2));%x.*sin(10*pi*x)+2;          %计算子代个体对应
目标函数值
      fit_v_sel=ranking(-opt_minmax*fun_v_sel);
      [chrom,fun_v]=reins(chrom,selchrom,1,1,opt_minmax*fun_v,opt_minmax*fun_v_sel);     %
根据目标函数值将子代个体插入新种群
      [f,id]=max(fun_v);                          %寻找当前种群最优解
      x=bs2rv(chrom(id,:),fieldd);
      f=f*opt_minmax;
      fun_v=fun_v*opt_minmax;
      plot3(x(1,1),x(1,2),f,'k*')
      hold on
      k_gen=k_gen+1;
      trace(k_gen,1)=f;
      trace(k_gen,2)=mean(fun_v);
end
figure
plot(trace(:,1),'r-*')
hold on
plot(trace(:,2),'b-o')
legend('各子代种群最优解','各子代种群平均值')
xlabel('迭代次数')
ylabel('目标函数优化情况')
title('多元函数优化过程')%其中，fun_mutv(x,y)为多元目标函数，可以根据具体的需要来定
义。本题中 fun_mutv(x,y)的定义如下：

%-------------------------------------------------------------------
function my=fun_mutv(x,y)
t1=zeros(size(x));
```

```
t2=t1;
for i=1:5
      t1=t1+i*cos((i+1)*x+i);
      t2=t2+i*cos((i+1)*y+i);
end
my=t1.*t2;
```

运行结果，以最小值为例





结果看 x 矩阵里面的两个数（x，y）和表格 trace 第一列最后一个（函数值）

提示：

你们注意下，网上下载的函数工具箱可能会有些问题，比如

```
title('一元函数优化过程')
??? Error: File: ranking.m Line: 45 Column: 1
At least one END is missing: the statement may beg
```

，点开之后

```
        Fi
End
```

，改成小写后就对了。