

在开阔的草原上，一群狼四处巡视，试图寻找最佳猎物位置，经过一段时间探测后，确定了小肥羊，准备追捕，聪明的狼群们，想到了分组包围策略，他们通过FCM算法分成了Nz个小分队，每个小分队有一个小队长，每队的其他灰狼在分队长的引导下一起包围追击猎物，当小肥羊意识到危险时，心理特别急躁，于是四处寻找最优逃生地点，逃跑速度也很快，与此同时，聪明的灰狼们协同去追捕小肥羊，大部分灰狼们在各分队长的引导下下去追捕包围猎物，少部分采取缓慢靠近的方式警惕小肥羊逃跑，包围圈越来越小，小肥羊可逃窜范围也逐步缩小，最终灰狼们得到了美餐。

优化算法中有两个思想比较好的算法，一个是哈米斯鹰算法、一个是麻雀算法，前者设计猎物搜寻最佳逃跑位置，可以减少局部最优的概率，在处于局部时有几率跃出局部；后者设计不同位置移动方式，使得个体位移方式多样化，制定不同矢量速度，特别是复杂函数，可以增大寻找最优区域的能力。

#### 算法步骤

(尽可能预先确定最优区域)

授权给：FFF



1、初始化位置，探测数次，返回每个个体的最佳位置准备狩猎，探测公式如下

$$X_{i+1} = X_i \pm Mc \cdot r$$

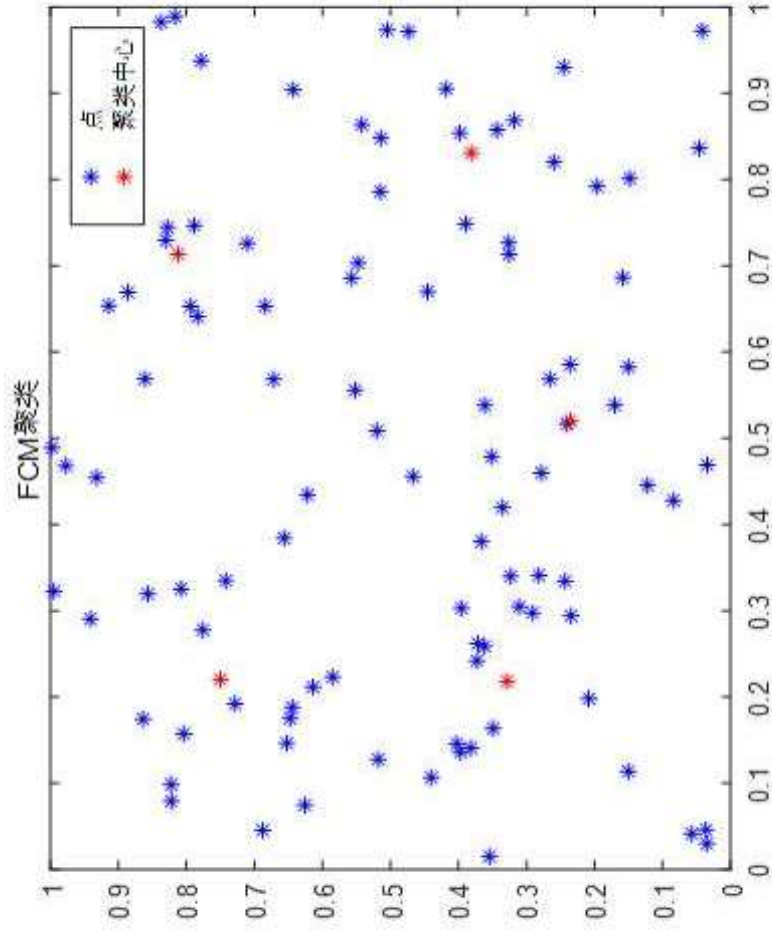
Mc为单次行动最大距离， $r \in [-1, 1]$ 随机数

返回个体的最佳位置程序如下

```
1 [f,b]=sort(f,2);
2 f=f(:,1);
3 xx=[];
4 for i=1:N+N1
5     xx(i,:)=x(i,[n*b(i,1)-1:n*b(i,1)]);
6 end
```

(设置多个参考点，增加个体位移的多样性，增大寻优能力)

2、分组，采用FCM算法尽可能使得分组均匀，例如下图



聚类中心公式:

$$P(\mu_i|x_j) = \frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^c (1/d_{rj})^{1/(b-1)}}$$

$$\mu_i = \frac{\sum_{j=1}^N P(\mu_i|x_j)^b x_j}{\sum_{j=1}^N P(\mu_i|x_j)^b}$$

d为欧式距离， P为隶属度矩阵、 μ为聚类中心更新公式

### 3、各组分别选择组内最优个体

```
1 X{Nz}=[];  
2 Y{Nz}=[];%因为f已经排序，这里直接根据隶属度矩阵U分组即可  
3 for i=1:N  
4     X{u(i)}=[X{u(i)};xx(i,:)];  
5     Y{u(i)}=[Y{u(i)};f(i,:)];  
6 end
```

(设置多种位移方式，使个体位移方式多样化，增大算法寻优能力)

### 4、狩猎开始

猎物急躁系数，pdist为距离函数

$$d = pdist(best, x)$$
$$z = e^{-\frac{d_{\min}}{Mc}}$$

(增加跃出局部最优概率)

蒙特卡洛模拟猎物寻找最佳逃避位置

```
1 for i=1:100%猎物环顾四周，找寻最佳逃亡点（蒙特卡罗随机模拟）  
2     dx=[dx;bestx+Mc*z*((det-ii)/det)*rand*((-1)^randi([1,2]))];%(det-ii)/det表示随着追捕，猎物可逃窜的范围越来越小  
3     yx=[yx;fun(dx(i,1),dx(i,2))];  
4 end
```

```

5  if min(yx)<bestf%如果找到更优逃窜方向及位置，就朝其移动，否则停留原地
6      [bestt,in]=min(yx);
7      bestc=dx(in,:);
8  else
9      bestt=bestf;
10     bestc=bestx;
11 end

```

组内最优个体位移公式

$$V_{i+1} = c_1 \cdot rand \cdot (bestx - X_i) + c_2 \cdot rand \cdot (bestc - X_i)$$

$$w = w_{\min} + \frac{(Y_i - Y_{\min})(w_{\max} - w_{\min})}{Y_{ave} - Y_{\min}}$$

$$X_{i+1} = X_i + w \cdot V_{i+1}$$

c1、c2为学习因子，wmin、wmax为惯性权重，bestx为当前猎物位置，bestc为猎物逃避位置

组内其余个体位移公式

①（狼群算法位移公式+粒子群算法位移公式）

$$C_{ij} = rand$$

$$A_{ij} = 2 \cdot a \cdot rand - a$$

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{c_i} X_i - A_{ij} \cdot C_{ij} \cdot (X_i - X_{ij})$$

$$V_{i+1} = c_1 \cdot rand \cdot (\bar{X} - X_i) + c_2 \cdot rand \cdot (bestc - X_i)$$

$$X_{i+1} = X_i + V_{i+1}$$

$a \in [2, 0]$  梯度下降

② (蝙蝠算法位移公式)

$$v = (bestc - X_i)$$

$$w = w_{\min} + (w_{\max} - w_{\min}) e^{-2 \left( \frac{\det_j}{\det} \right)^2}$$

$$X_{i+1} = X_i + w \cdot v \cdot rand$$

det为最大迭代次数，detj为当前第j次迭代

(增加收敛速度，提高运算效率)

5、根据记忆库重新制定猎物及捕猎者角色，重新分组

**完整代码**

```
1 clear
2 clc
3 tu
4 N=300;%狼群大小
5 N1=1;%猎物数为1
```

```

6  Nz=5;%狼群最大分组数
7  tc=4;%探测次数
8  det=50;%迭代次数
9  c1=0.2;c2=0.8;%学习因子最大最小(粒子群算法参数)
10 wmin=0.3;wmax=0.7;%惯性权重最大最小(粒子群算法、蝙蝠算法参数)
11 sub=[-50,-50];%自变量下限
12 up=[50,50];%自变量上限
13 Mc=(up-sub)/20;%单次行动最大范围，传统算法最初几次迭代，某些复杂函数可能跃过最优到局部最优，这里起一个限制作用
14 n=2;%自变量个数
15 x=[];
16 f=[];
17 for i=1:N+N1
18     x(i,:)=(up-sub).*rand(1,n)+sub;%初始化位置
19     f(i,1)=fun(x(i,1),x(i,2));%初始适应度值
20 end
21 %探测阶段
22 for ii=1:tc
23     for i=1:N+N1
24         %为了还原真实捕猎场景，这里的位置变化是连续的，并不是全局随机产生（这里肯定全局随机要好，但一般都是借物说物别人才会理解，
25         a=x(i,(ii-1)*n+1:ii*n)+Mc*rand*(-1)^randi([1,2]));%探测阶段位置更新
26         %如果超出范围就取边界值
27         a=max(sub,a);
28         a=min(up,a);
29         x(i,ii*n+1:(ii+1)*n)=a;
30         f(i,ii+1)=fun(x(i,ii*n+1),x(i,(ii+1)*n));%更新探测位置适应度值
31     end
32 end

```

```
33
34 %每只狼返回最优探测位置，准备狩猎，猎物也在随机移动
35 [f,b]=sort(f,2);
36 f=f(:,1);
37 xx=[];
38 for i=1:N+N1
39     xx(i,:)=x(i,[n*b(i,1)-1:n*b(i,1)]);
40 end
41 %制定剧本，最优值为猎物，其余为狩猎者
42 [f,b]=sort(f);
43 xx=xx(b,:);
44 bestf=[];
45 bestx=[];
46 bestf=f(1);%猎物适应度值
47 bestx=xx(1,:);%猎物位置
48 fmin=min(f);%统计当前个体最小值，粒子群算法位置更新公式会用到
49 fave=mean(f);%统计当前个体平均值，粒子群算法位置更新公式会用到
50 f(1)=[];
51 xx(1,:)=[];
52 trace=bestf;%记录最优值
53
54 %狼群分组，采用fcm聚类，吹起牛来说这里模仿狼群协同狩猎
55 U=[];
56 [~,U]=FC(xx, Nz);%FCM聚类
57 %别问为什么要用聚类算法，问就是为了帮助算法跃出局部最优
58 u=[];
59 [~,u]=max(U);
```



```
60 X=[];
61 Y=[];
62 X{Nz}=[];
63 Y{Nz}=[];%因为f已经排序，这里直接根据隶属度矩阵u分组即可
64 for i=1:N
65     X{u(i)}=[X{u(i)};xx(i,:)];
66     Y{u(i)}=[Y{u(i)};f(i,:)];
67 end
68
69 %绘制迭代过程个体位置分布
70 figure(2)
71 subplot(3,3,1)
72 hold on
73 plot(xx(:,1),xx(:,2),'b*')
74 plot(bestx(1),bestx(2),'r*')
75 title('初始位置分布')
76 axis([sub(1),up(1),sub(2),up(2)])
77 I=[1,2,3,5,7,9,11,13];%需要绘制的迭代次数
78 I0=1;
79
80 %狩猎阶段
81 for ii=1:det
82     %猎物更新位置(哈里斯鹰算法猎物位置更新)
83     d=[];
84     d=pdist2(bestx,xx);
85     z=exp(-min(d)/min(Mc));%猎物急躁系数，总所周知，有捕猎者靠近，猎物就会变得急躁起来，逃窜的速度就会越大
86     dx=[];
```

```

87 yx=[];
88 bestt=[];
89 bestc=[];
90 for i=1:100%猎物环顾四周，找寻最佳逃亡点（蒙特卡罗随机模拟）
91     dx=[dx;bestx+Mc*z*((det-ii)/det)*rand*(-1)^randi([1,2])];%(det-ii)/det表示随着追捕，猎物可逃窜的范围越来越小
92     yx=[yx;fun(dx(i,1),dx(i,2))];
93 end
94 if min(yx)<bestf%如果找到更优逃窜方向及位置，就朝其移动，否则停留原地
95     [bestt,in]=min(yx);
96     bestc=dx(in,:);
97 else
98     bestt=bestf;
99     bestc=bestx;
100 end
101 %参考麻雀算法使移动方式多样化
102 %每组头狼更新位置（粒子群算法位置更新步骤）
103 v=[];
104 w=[];
105 for i=1:Nz
106     if isempty(X{i})==1
107         continue
108     end
109     %头狼会首先根据猎物逃窜方向进行追捕
110     v=c1.*rand(1,n).*(bestx-X{i}(1,:))+c2.*rand(1,n).*(bestc-X{i}(1,:));%粒子群算法速度更新公式
111     v=max(v,-Mc);
112     v=min(v,Mc);
113     w=wmin+(Y{i}(1)-fmin)*(wmax-wmin)/(fave-fmin);%粒子群算法自适应权重更新公式

```

```

114     X{i}(1,:)=X{i}(1,:)+w.*v;
115     Y{i}(1)=fun(X{i}(1,1),X{i}(1,2));
116 end
117 %每组其他狼更新位置(结合灰狼算法位置更新公式、粒子群算法更新公式、蝙蝠算法位置更新公式)
118 for i=1:Nz
119     if isempty(X{i})==1
120         continue
121     end
122     if size(X{i},1)>=2
123         for j=2:size(X{i},1)
124             if rand>=0.3%追捕者，跟随各组头狼信号追捕猎物
125                 %但会参考每组头狼奔跑的矢量位置以及猎物的矢量位置
126                 C=[];
127                 A=[];
128                 X123=[];
129                 for k=1:Nz
130                     if isempty(X{k})==1
131                         continue
132                     end
133                     C=rand(1,n);
134                     A=2*((det-ii)/det)*rand(1,n)-((det-ii)/det);
135                     X123=[X123;X{k}(1,:)-A.*C.*(X{k}(1,:)-X{i}(j,:))];%灰狼算法位置更新公式
136                 end
137                 X123=mean(X123);
138                 v=c1.*rand(1,n).*(X123-X{i}(j,:))+c2.*rand(1,n).*(bestc-X{i}(j,:));%结合粒子群算法速度更新公式；一是参
139                 v=max(v,-Mc);
140                 v=min(v,Mc);

```

```

141 X{i}(j,:)=X{i}(j,:)+v;%位置更新
142 else%警惕者，会直接朝猎物靠近，随着追捕，离猎物越近，越放慢步伐
143 w=(wmax-wmin)*exp(-2*(ii/det)^2)+wmin;%蝙蝠算法惯性权重公式
144 v=bestc-X{i}(j,:);
145 v=max(v,-Mc);
146 v=min(v,Mc);
147 X{i}(j,:)=X{i}(j,:)+w.*v.*rand(1,n);
148 end
149 Y{i}(j)=fun(X{i}(j,1),X{i}(j,2));
150 end
151 end
152 end
153 %根据记忆库重新制定角色及位置，思路鉴于免疫算法
154 xxx=[];
155 fff=[];
156 for i=1:Nz
157     xxx=[xxx;X{i}];
158     fff=[fff;Y{i}];
159 end
160 xxx=[xxx;xx;bestc];
161 fff=[fff;f;bestt];
162 [fff,b]=sort(fff);
163 xxx=xxx(b,:);
164 f=fff(1:N+N1,:);
165 xx=xxx(1:N+N1,:);
166 bestf=f(1);%更新猎物适应度
167 bestx=xx(1,:);%更新猎物位置

```