> 版权声明：本文为博主原创文章，未经博主允许不得转载https://blog.csdn.net/g425680992/article/details/80905019

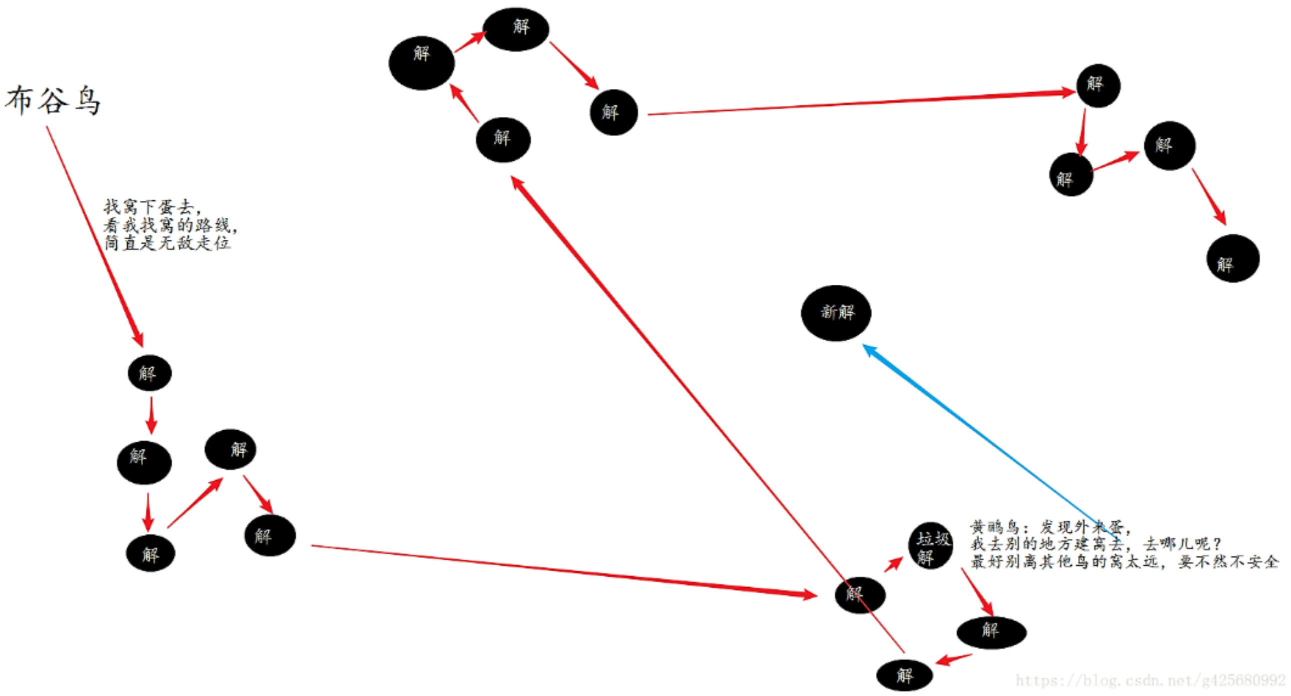
1 从布谷鸟的育雏到布谷鸟算法

布谷鸟不会做窝，也不会育雏，在春末夏初，向北飞，趁别的鸟(宿主鸟)外出觅食时，将卵蛋产在宿主鸟窝里，让宿主鸟抚养自己孩子 。当然，布谷鸟在产卵前，为了不被宿主鸟发现鸟窝的异常，会把宿主的卵移走。而一旦靠养母孵化的雏鸟，也有将宿主鸟本身的雏鸟推出巢穴的本性，并且会模仿其他鸟的行为来增大不被宿主鸟发现的概率[1]

2009年，Xin-She Yang[2] 与Suash Deb在《Cuckoo Search via Levy Flights》一文中提出了布谷鸟算法(简称CS)。假设每只布谷鸟一次只产一枚卵，并且宿主鸟发现外来鸟蛋后，就舍弃该鸟窝，另寻他地建造新的鸟窝，那么可以认为：鸟窝=卵蛋=解，卵蛋是否能够成功被宿主鸟孵化并茁长成长是衡量解好坏的唯一标准。布谷鸟寻找鸟窝下蛋的过程就是在D维空间中寻找解的过程，而鸟窝的好坏象征着解的好坏。

2 布谷鸟算法

布谷鸟算法是布谷鸟育雏行为和莱维飞行结合的一种算法。



在CS算法中，有两个路径(或者说成是两个位置的更新)备受关注：

- 一个是布谷鸟寻找鸟窝下蛋的寻找路径是采用早已就有的莱维飞行[2]，如上图所示，无敌的走位是一种长步长与短步长相间的走位，这其实就是莱维飞行的主要特点，学者们也证实了自然界中很多鸟类的飞行也遵从莱维飞行，这也是最有效寻找目标的方法之一。所以采用莱维飞行更新鸟窝位置的公式被定义如下：

  > $X_{t+1} = X_t + \alpha \bigotimes Levy(\beta)$， 公式(1)

  其中，α是步长缩放因子，$Levy(\beta)$是莱维随机路径，$\bigotimes$ 就是.*运算

- 另一个是宿主鸟以一定概率Pa发现外来鸟后重新建窝的位置路径，这个路径可以用莱维飞行或者随机方式[3]，(本文采用随机)，除此之外，这个位置普遍采用偏好随机游动的方式，即利用了其他鸟窝的相似性[4]。所以新建的鸟窝的位置的公式被定义如下：

  > $X_{t+1} = X_t + r \bigotimes Heaviside(Pa - \epsilon) \bigotimes (X_i - X_j)$， 公式(2)

  其中，r, ϵ 是服从均匀分布的随机数，$Heaviside(x)$ 是跳跃函数(x>0,=1;x<0,=0)，X_i, X_j 是其他任意的连个鸟窝。

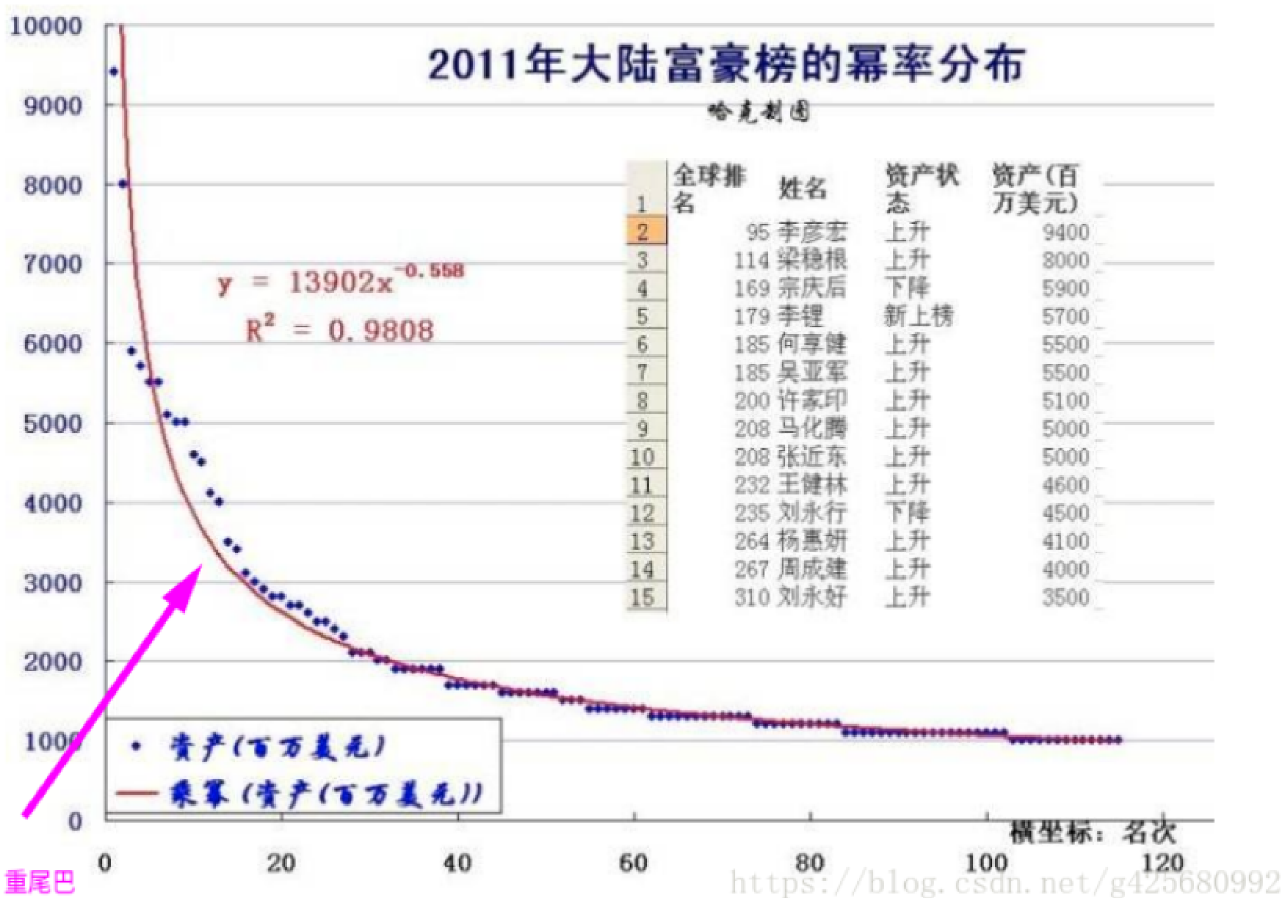
CS算法的执行过程如下：

1 Cuckoo Search via Levy Flights

目标函数$f(\boldsymbol{x})$, $\quad \boldsymbol{x} = (x_1, ..., x_d)^T$
产生n个寄主的初始种群$\boldsymbol{x}_i$
while ($t <$ MaxGeneration) or (stop criterion) **do**
    随机取一个布谷鸟
    通过Levy飞行产生一个解
    评估解的质量或目标函数值f_i
    从n个巢中随机选择一个（假设为j）
    if $f_i < f_j$ **then**
        将j用解i代替
    end if
    一部分（p_a）糟糕的巢被抛弃
    新巢/解由式(1)产生
    保存最佳的解（或者，高质量解的巢）
    排列解找出当前最佳
    更新$t \leftarrow t + 1$
end while
后处理与可视化

3 莱维飞行与公式(1)的深层含义

从数学的发展史上说，早在1937年， P. Levy[5]确定了对称Levy稳定分布的积分形式为$Levy(s)=\frac{1}{\pi}\int_0^{+\infty}exp(-\beta|k|^{\lambda})cos(ks)dk$,但是该积分并没有明确的解析，要生成一个服从该分布的随机数是难上加难的问题，不过当$s\gg s_0>0$，即$s\to\infty$时，$Levy(s)\approx\frac{\lambda\beta\Gamma(\lambda)sin(\frac{\pi\lambda}{2})}{\pi}\cdot\frac{1}{s^{1+\lambda}}$，通常$\beta=1$ 。这个近似的分布呈现幂律行为(重尾或长尾巴)，这个行为类似于二八原则[6],或者说少部分人集中了世界大部分的财富，正如下图所示的，这个分布总是有一个长尾巴或者称之为重尾巴，有时也叫做一个翼。



莱维飞行的方差随时间呈现指数的关系，即$\sigma^2(t)\sim t^{3-\beta},1\leq\beta\leq 3$,所以莱维飞行比布朗运动更加的出色。

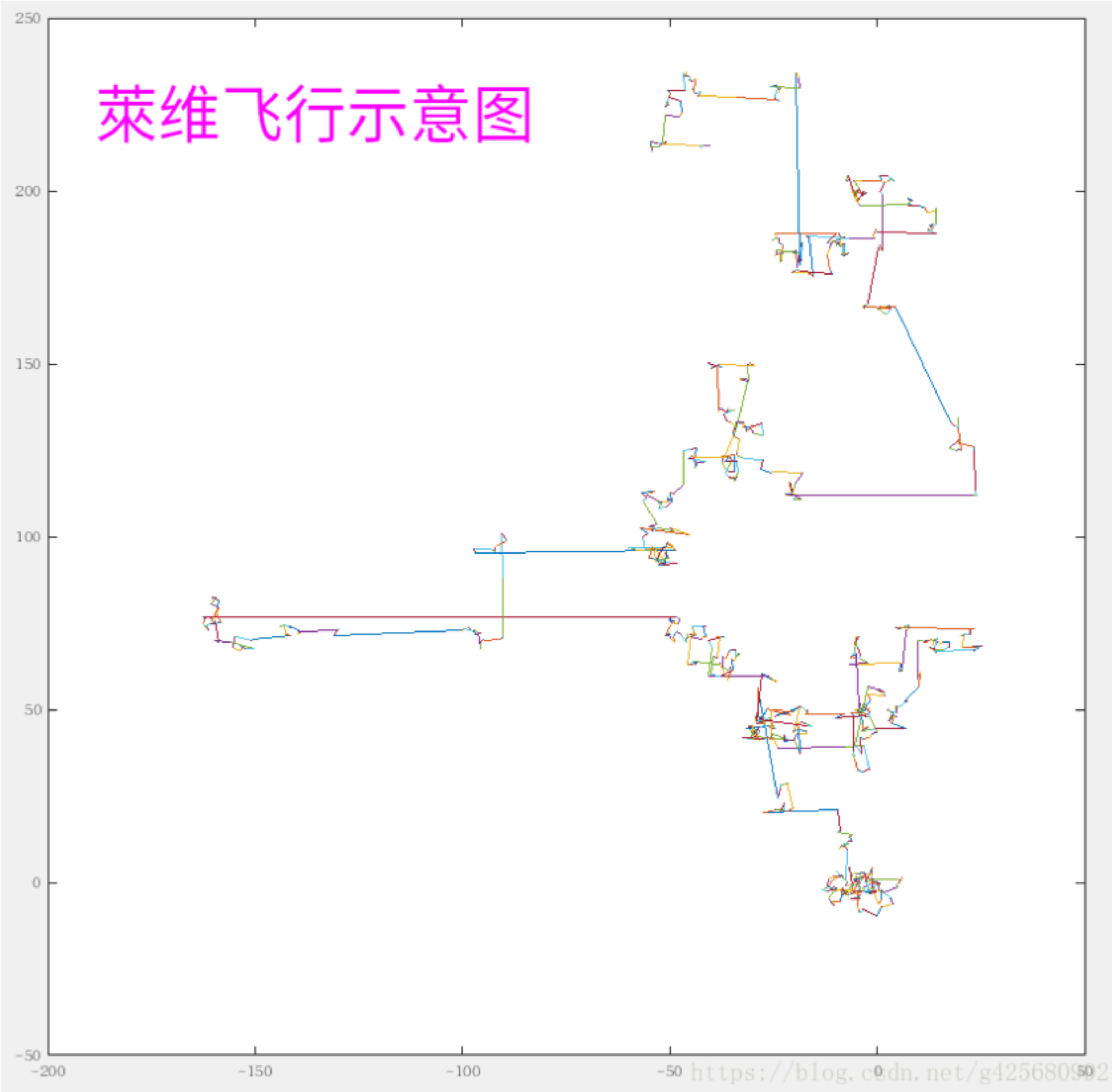
此后，不少学者根据这个近似部分提出很多用于生成服从莱维分布的随机数的实现方法，其中就包含了Mantegna[7]在1994年提出的一种用正太分布求解随机数的方法，有时也叫Mantegna方法，生成服从莱维分布的随机步长的方法如下：

> $s=\frac{u}{|v|^{\frac{1}{\beta}}}$

其中，$u \sim N(0,\sigma^2), v \sim N(0,1), \sigma = \{\frac{\Gamma(1+\beta)sin(\frac{\pi\beta}{2})}{\beta\Gamma(\frac{1+\beta}{2})2^{\frac{\beta-1}{2}}}\}^{\frac{1}{\beta}}$

在matlab中用Mantegna方法模拟二维平面莱维飞行:

```
% Mantegna方法模拟莱维飞行
%author zhaoyuqiang
x = [0,0];
y = [0,0];
beta = 1.5;
sigma_u =
(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta
-1)/2)))^(1/beta);
sigma_v = 1;
for i=1:1000
    u = normrnd(0,sigma_u);
    v = normrnd(0,sigma_v);
    s = u/(abs(v))^(1/beta);
    x(:,1) = x(:,2);
    x(:,2) = x(:,1)+1*s;
    u = normrnd(0,sigma_u);
    v = normrnd(0,sigma_v);
    s = u/(abs(v))^(1/beta);
    y(:,1) = y(:,2);
    y(:,2) = y(:,1)+1*s;
    plot(x,y);
    hold on;
end
axis square;
```



从模拟上来看，图形的路径确实符合莱维飞行的长短相间的特征，Mantegna用正太分布实现了生成服从莱维分布随机步长的方法是可靠的 。

时间到了2009年，Xin-She Yang 与Suash Deb提出了布谷鸟算法，同时，Yang把Levy分布函数经过简化和傅立叶变换后得到其幂次形式的概率密度函数[7]：$Levy \sim u = t^{-\beta}, 1 \leq \beta \leq 3$ 。并把莱维飞行用在了鸟窝位置的更新上，于是产生了公式(1)$X_{t+1} = X_t + \alpha \bigotimes Levy(\beta)$ 。这个计算式其实就是$X_{t+1} = X_t + \alpha S$，S就是服从Levy分布$Levy \sim u = t^{-\beta}, 1 \leq \beta \leq 3$的随机步长，考虑到具体怎么计算时，Yang采用的正是1994年的Mantegna方法 。

所以在布谷鸟算法中，我们可以用下面的具体计算公式来计算鸟窝的更新位置:

> $X_{t+1} = X_t + \alpha S = X_t + \alpha.* \frac{\text{服从}N(0,\sigma^2)\text{的随机数}}{|\text{服从}N(0,1)\text{的随机数}|^{\frac{1}{\beta}}}$

其中，$\sigma = \{\frac{\Gamma(1+\beta)sin(\frac{\pi\beta}{2})}{\beta\Gamma(\frac{1+\beta}{2})2^{\frac{\beta-1}{2}}}\}^{\frac{1}{\beta}}$，通常，$\beta = 1.5$

这在matlab等一些编程工具中都是可以计算的。

值得一提的是，α是步长缩放因子，通常$\alpha = 1$，在之后的布谷鸟算法发展中，针对α 有各种各样的变种，如Yang[^8]为了让算法适应不同的解，让 $\alpha = \alpha_0(X_i - Xj)$,$X_i, X_j$为任意不同的解 。

4 附：CS算法求解函数最小值代码(CSDN下载地址http:)

> 求函数$f(x) = \sum_{i=1}^{n} x_i^2, (-20 \le x \le 20,\ n = 10)$最小值

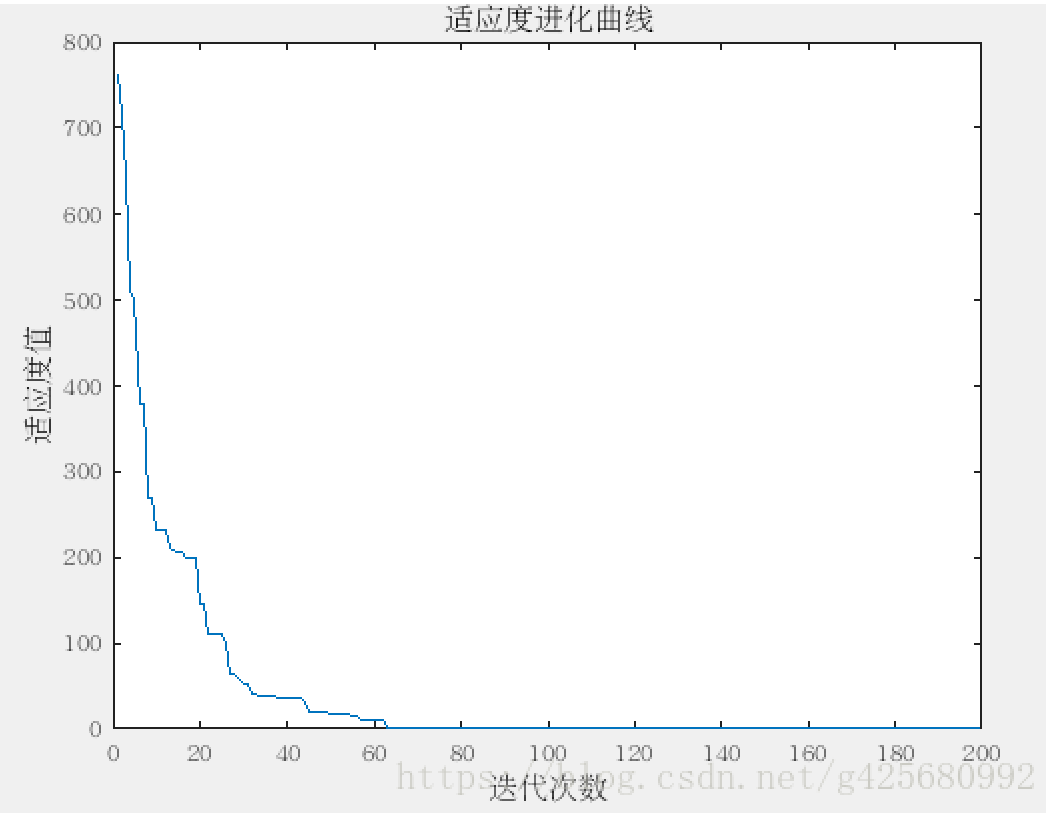
```
% Script 布谷鸟算法，求解函数最小值
% @author zhaoyuqiang
%#ok<*SAGROW> Remove hints of syntax
%#ok<*CLALL>
%#ok<*FNDSB>
clear all ;
close all ;
clc ;
N = 25; %  Number of nests(The scale of solution)
D = 10 ; %  Dimensionality of solution
T = 200 ; % Number of iterations
Xmax = 20 ;
Xmin = -20 ;
Pa = 0.25 ; % Probability of building a new nest(After host
bird find exotic bird eggs)
nestPop = rand(N,D)*(Xmax-Xmin)+Xmin ;  % Random initial
solutions
for t=1:T
    levy_nestPop =  func_levy(nestPop,Xmax,Xmin) ; % Generate
new solutions by Levy flights
```

```
    nestPop = func_bestNestPop(nestPop,levy_nestPop);  %
Choose a best nest among  new and old nests
    rand_nestPop = func_newBuildNest(nestPop,Pa,Xmax,Xmin); %
Abandon(Pa) worse nests and build new nests by (Preference
random walk )
    nestPop = func_bestNestPop(nestPop,rand_nestPop) ; %
Choose a best nest among  new and old nests
    [~,index] = max(func_fitness(nestPop)) ; % Best nests
    trace(t) = func_objValue(nestPop(index,:)) ;
end
figure
plot(trace);
xlabel('迭代次数') ;
ylabel('适应度值') ;
title('适应度进化曲线') ;
```

```
function [ result ] = func_levy( nestPop,Xmax,Xmin)
%FUNC_LEVY : Update position of nest by using Levy flights
%@author : zhaoyuqiang
[N,D] = size(nestPop) ;
% Levy flights by Mantegna's algorithm
beta = 1.5 ;
alpha = 1 ;
sigma_u =
(gamma(1+beta)*sin(pi*beta/2)/(beta*gamma((1+beta)/2)*2^((beta
-1)/2)))^(1/beta) ;
sigma_v = 1 ;
u = normrnd(0,sigma_u,N,D) ;
v = normrnd(0,sigma_v,N,D) ;
step = u./(abs(v).^(1/beta)) ;
% alpha = 0.1.*(nestPop(randperm(N),:)-
nestPop(randperm(N),:)); % Bad effect
nestPop = nestPop+alpha.*step ;
% Deal with bounds
nestPop(find(nestPop>Xmax)) = Xmax ; %#ok<*FNDSB>
nestPop(find(nestPop<Xmin)) = Xmin ;
result = nestPop ;
end
```

```
function [ nestPop ] = func_bestNestPop( nestPop,new_nestPop )
%FUNC_ 此处显示有关此函数的摘要
%@author zhaoyuqiang
index = find(func_fitness(nestPop)<func_fitness(new_nestPop))
;
nestPop(index,:) = new_nestPop(index,:) ;
end
```

```
function [ nestPop ] = func_newBuildNest( nestPop ,Pa
,Xmax,Xmin)
%FUNC_NEWBUILDNEST new solutions are generated by using the
similarity
% between the existing eggs/solutions and the host
eggs/solutions with a discovery rate pa .
%@author zhaoyuqiang
[N,D] = size(nestPop) ;
nestPop = nestPop+rand.*heaviside(rand(N,D)-Pa).*
(nestPop(randperm(N),:)-nestPop(randperm(N),:));
% Deal with bounds
nestPop(find(nestPop>Xmax)) = Xmax ; %#ok<*FNDSB>
nestPop(find(nestPop<Xmin)) = Xmin ;
end
```



5 源码下载

https://download.csdn.net/download/g425680992/10517545

6 参考文献

---

1. 布谷鸟搜索算法研究综述， 兰少峰↩
2. Cuckoo search via Lévy flights. Yang XS↩↩
3. 逐维改进的布谷鸟搜索算法，王李进↩
4. Cuckoo search for inverse problems and simulated-driven shape optimization ,Yang XS↩
5. P. Levy, Theoric de l'Addition des Variables Aleatoires↩
6. P. Levy, Theoric de l’Addition des Variables Aleatoires ↩
7. Nature-Inspired Metaheuristic Algorithms Second Edition,Yang XS↩↩