粒子群寻优算法

写这篇笔记的因为，在我学习这个算法的时候，从网上看了一些资料，理论知识大致讲得都差不多，给出的一些实例简单，本来是很能说明问题，但是其中有一些错误，造成了一些误解，以下是根据我自己的理解编写的代码，经测试能够得到正确的寻优结果，简单说明如下。

算法的原理就不在这多说了，参考相关文献。本例是对一维函数f=-x(x-2)进行寻优，为方便理解，以两个粒子为例，在初始化时人为把两个粒子放在了-5和5的位置，其实这里可以随机放的。

粒子群算法关键有二点，一是位置和速度的更新，二是局部最优解和全局最优解的更新。

对于一，标准粒子群算法给出的公式是：

v[i]=w\*v[i]+c1\*r1\*(pbest[i]-x[i])+c2\*r2\*(gbest-x[i]);

x[i]+=v[i];

注意，这里的pbest[i]和gbest是局部最优解和全局最优解所对应的粒子所在位置，与x对应，而有些网上的文章说是适应度函数值，即与y对应，这是不正确的！另外，在位置和速度更新的时候，要考虑是否越界，包括上界和下界，有些网上的文章只对上界做了约束，而没有对下界做寿 约束，这样会导致负速度很大或者位置跑得很远，导致找不到最优解。

对于二，包括局部最优解和全局最优解粒子位置和所对应的适度度函数的更新。首先计算所当前所有粒子所在位置的适应度，然后对每个粒子选其历史最大适应度做为局部适应度最大最优解，所对就位置为局部位置最优解，它们是一一对应的。再对所有粒子的局部适应度最优解进行比较，最大者做为全局适应度最优解，所对应的位置为全局位置最优解，此二者也是一一对应的。这里选“最大”是在寻找极大值的前提下的，如果是寻找极小值，那就得相反了。

#include "stdafx.h"

#include <math.h>

#include <iostream>

#include <time.h>

#include <stdlib.h>

using namespace std;

#define N 2 //2个粒子

#define IMAX 30 //最大次数

#define random() (rand()/double(RAND\_MAX)) //产生0～x范围的随机数

void fitnessFunction(double \*x,double \*y,int n)//适应度函数

{

for (int i=0;i<n;i++)

y[i]=-1\*x[i]\*(x[i]-2);

}

void OutputInfo(double \*x,double \*y,double \*v,double \*pbest,int n,int k)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*K="<<k<<":\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for (int i=0;i<n;i++)

{

cout<<"X["<<i<<"]="<<x[i]<<",Y["<<i<<"]="<<y[i]<<endl;

cout<<"V["<<i<<"]="<<v[i]<<endl;

cout<<"pBest["<<i<<"]="<<pbest[i]<<endl<<endl;

}

cout<<endl;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

//定义

double \*x=new double[N]; //粒子位置

double \*y=new double[N]; //当前粒子对应的适应值

double \*v=new double[N]; //当前粒子速度

double \*pbest=new double[N]; //局部粒子局部最优位置

double \*pbestf=new double[N]; //局部粒子局部最优位置处对应的适应值

double gbest; //全局粒子最优位置

double gbestf; //全局粒子最优位置处对应的适应值

double c1=2.0;

double c2=2.0;

double vmax=0.2;

//初始化

x[0]=-5;//+random()\*10;

x[1]=5;//-random()\*10;

v[0]=0.01;

v[1]=0.015;

fitnessFunction(x,y,N);

for (int i=0;i<N;i++)

{

pbestf[i]=y[i];

pbest[i]=x[i];

if (i==0)

{

gbestf=y[i];

gbest=x[i];

}

if (y[i]>gbestf)

{

gbestf=y[i];

gbest=x[i];

}

}

//开始搜索过程

cout<<"Start:"<<endl;

srand((int)time(0));

for (int k=0;k<IMAX;k++)

{

double w=0.4;//惯性系数

for (int i=0;i<N;i++)

{

//更新位置和速度

double r1=random();

double r2=random();

v[i]=w\*v[i]+c1\*r1\*(pbest[i]-x[i])+c2\*r2\*(gbest-x[i]);

if (v[i]>vmax)v[i]=vmax;

if (v[i]<-1\*vmax)v[i]=-1\*vmax;

x[i]+=v[i];

if (x[i]>2.0)x[i]=2.0;

if (x[i]<-2.0)x[i]=-2.0;

}

fitnessFunction(x,y,N);

//更新个体极值和群体极值

for (int i=0;i<N;i++)

{

if (y[i]>pbestf[i])

{

pbest[i]=x[i];

pbestf[i]=y[i];

}

if (pbestf[i]>gbestf)

{

gbestf=pbestf[i];

gbest=pbest[i];

}

}

//输出本次寻优结果

OutputInfo(x,y,v,pbest,N,k);

cout<<"gbest:"<<gbest<<endl;

}

system("pause");

delete []x;

delete []y;

delete []v;

delete []pbest;

delete []pbestf;

return 0;

}