

```

#define gen_max 1000
#define D 3
#define NP 20
/* #define F 0.8 */
#define CR 0.1
#define inibound_l 0.0
#define inibound_h 1.0

/*----Constant for rnd_uni()-----*/
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS1 1.2e-7
#define RNMIX (1.0-EPS1)

#include<stdlib.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include<conio.h>

int seed;
void DE(float sfactor);
void assignd(int , double a[], double b[]);
void assignd(int , double a[], double b[])
{
    int j;
    for(j=0;j<D;j++)
    {
        a[j]=b[j];
    }
}

double evaluate(double [],long *);
double evaluate(double tmp[],long *nfe)
{
    double cost; (*nfe)++; /* tmp[0]=v1, tmp[1]=v2 tmp[2]=y1 */

    cost=(7.5*(tmp[2]))+(5.5*(1.0-(tmp[2]))) +7.0*10*tmp[0]+6.0*10*tmp[1]+50*((1-(tmp[2]))/(0.8*(1-
exp(-0.4*tmp[1]*10))))+50*(tmp[2])/(0.9*(1-exp(-0.5*tmp[0]*10)));
    return(cost);

} /***** end of evaluate() *****/

float rnd_uni(long *);
float rnd_uni(long *idum)
{
    long j; long k;
    static long idum2=123456789;
    static long iy=0;static long iv[NTAB]; float temp;
    if(*idum<=0)
    {
        if(-(*idum)<1) *idum=1; else *idum=-(*idum); idum2=(*idum);
        for(j=NTAB+7;j>=0;j--)
        {
            k=(*idum)/IQ1;
            *idum=IA1*(*idum-k*IQ1)-k*IR1;
            if(*idum<0) *idum+=IM1;
            if(j<NTAB) iv[j]=*idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ1;
    *idum=IA1*(*idum-k*IQ1)-k*IR1;
    if(*idum<0) *idum+=IM1;
    k=idum2/IQ2;
    idum2=IA2*(idum2-k*IQ2)-k*IR2;
    if(idum2<0) idum2+=IM2;
}

```

```

        j=iy/NDIV;          iy=iv[j]-idum2;   iv[j]=*idum;
        if (iy<1)          iy+=IMM1;
        if ( (temp=AM*iy)>RNMX) return RNMX;
        else                return temp;
    }

void DE(float sfactor)
{
    int i,j,k,a,b,c,d,e,count=0,strategy=7,imin; long nfe=0;
    double x1[NP][D],x2[NP][D],cost[NP],trial[D],cost_trial,pen,costmax;
    double lhs1,lhs2,lhs3,lhs4,costmin,bestit[D],best[D];
    clock_t start, end;          FILE *fout; char ch; float F=sfactor;
    seed=(rand()%10001);
    long rnd_uni_init= -(long)seed; start = clock();

    for (i=0;i<NP;i++)
    {
        for (j=0;j<D;j++)
        {
            x1[i][j]=inibound_l + rnd_uni(&rnd_uni_init)*(inibound_h-inibound_l);
            if(x1[i][j]<=0.0) x1[i][j]=1e-7;
        }
        pen=0.0;

        if(x1[i][2]>=0.5) x1[i][2]=1.0;
        if(x1[i][2]<0.5) x1[i][2]=0.0;

        lhs1=0.9*(1-exp(-0.5*x1[i][0]*10))-2*(x1[i][2]);
        if(lhs1>0.0)
        {
            pen=lhs1*100.0;
            cost[i]=evaluate(x1[i], &nfe);
            cost[i]=cost[i]+pen;
            continue;
        }

        lhs2=0.8*(1.0-exp(-0.4*x1[i][1]*10))-2*(1-(x1[i][2]));
        if(lhs2>0.0)
        {
            pen=lhs2*100.0;
            cost[i]=evaluate(x1[i], &nfe);
            cost[i]=cost[i]+pen;
            continue;
        }
        lhs3=10*x1[i][0]-(10*x1[i][2]);
        if(lhs3>0.0)
        {
            pen=lhs3*100.0;
            cost[i]=evaluate(x1[i], &nfe);
            cost[i]=cost[i]+pen;
            continue;
        }
        lhs4=10*x1[i][1]-10*(1-(x1[i][2]));
        if(lhs4>0.0)
        {
            pen=lhs4*100.0;
            cost[i]=evaluate(x1[i], &nfe);
            cost[i]=cost[i]+pen;
            continue;
        }
        if(lhs1<=0.0 && lhs2<=0.0 && lhs3<=0.0 && lhs4<=0.0)
            cost[i]=evaluate(x1[i], &nfe);
    }

    costmin=cost[0];
    imin=0;
    for(i=1;i<NP;i++)
    {
        if(cost[i]<costmin)
        {
            costmin=cost[i];
            imin=i;
        }
    }
}

```

```

    assignd(D,best,x1[imin]);
    assignd(D,bestit,x1[imin]);
while (count<gen_max)
{
    for (i=0;i<NP;i++)
    {
        do a=int ((rnd_uni(&rnd_uni_init))*NP); while (a==i);
        do b=int (rnd_uni(&rnd_uni_init)*NP); while (b==i || b==a);
        do c=int (rnd_uni(&rnd_uni_init)*NP); while (c==i || c==a || c==b);
        do d=int (rnd_uni(&rnd_uni_init)*NP); while (d==i || d==a || d==b || d==c);
        do e=int (rnd_uni(&rnd_uni_init)*NP); while (e==i || e==a || e==b || e==c || e==d);

        /* F=rnd_uni(&rnd_uni_init); */
/*-----de/rand/1/bin-----*/
        if(strategy==1)
        { j=int (rnd_uni(&rnd_uni_init)*D);

            for (k=1;k<=D;k++)
            {
                if(rnd_uni(&rnd_uni_init)<CR || k==D)
                {
                    trial[j]=x1[c][j]+F*(x1[a][j]-x1[b][j]);
                }
                else trial[j]=x1[i][j];

                if(trial[j]<=0.0)          trial[j]=1e-7;
                if(trial[2]>=0.5)        trial[2]=1.0;
                if(trial[2]<0.5)        trial[2]=0.0;
                j=(j+1)%D;
            }
        }
/*-----DE/best/1/bin-----*/
        else if (strategy==2)
        {
            j=int (rnd_uni(&rnd_uni_init)*D);

            for (k=1;k<=D;k++)
            {
                if ((rnd_uni(&rnd_uni_init))<CR || k==D)
                {
                    trial[j]=bestit[j]+F*(x1[a][j]-x1[b][j]);
                }
                else trial[j]=x1[i][j];
                if(trial[j]<=0.0)          trial[j]=1e-7;
                if(trial[2]>=0.5)        trial[2]=1.0;
                if(trial[2]<0.5)        trial[2]=0.0;
                j=(j+1)%D;
            }
        }
/*-----de/best/2/bin-----*/
        else if (strategy==3)
        {
            /* assignd(D,trial,x1[i]); */

            j=int (rnd_uni(&rnd_uni_init)*D);

            for (k=1;k<=D;k++)
            {
                if ((rnd_uni(&rnd_uni_init))<CR || k==D)
                {
                    trial[j]=bestit[j]+F*(x1[a][j]+x1[b][j]-x1[c][j]-x1[d][j]);
                }
                else trial[j]=x1[i][j];
                if(trial[j]<=0.0)          trial[j]=1e-7;

                if(trial[2]>=0.5)          trial[2]=1.0;
            }
        }
    }
}

```

```

        if(trial[2]<0.5)      trial[2]=0.0;

        j=(j+1)%D;
    }
}

/*-----de/rand/2/bin-----*/
else if (strategy==4)
{
    /* assignd(D,trial,x1[i]); */
    j=int (rnd_uni(&rnd_uni_init)*D);

    for (k=1;k<=D;k++)
    {
        if ((rnd_uni(&rnd_uni_init))<CR || k==D)
        {
            trial[j]=x1[e][j]+F*(x1[a][j]+x1[b][j]-x1[c][j]-x1[d][j]);
        }
        else trial[j]=x1[i][j];
        if(trial[j]<=0.0)      trial[j]=1e-7;
        if(trial[2]>=0.5)    trial[2]=1.0;
        if(trial[2]<0.5)    trial[2]=0.0;

        j=(j+1)%D;
    }
}

/*-----de/rand-to-best/1/bin-----*/
else if (strategy==5)
{
    /* assignd(D,trial,x1[i]); */
    j=int (rnd_uni(&rnd_uni_init)*D);

    for (k=1;k<=D;k++)
    {
        if ((rnd_uni(&rnd_uni_init))<CR || k==D)
        {
            trial[j]=trial[j]+F*(bestit[j]-trial[j])+F*(x1[a][j]-x1[b][j]);
        }
        else trial[j]=x1[i][j];
        if(trial[j]<=0.0)      trial[j]=1e-7;
        if(trial[2]>=0.5)    trial[2]=1.0;
        if(trial[2]<0.5)    trial[2]=0.0;

        j=(j+1)%D;
    }
}

/*-----de/rand/1/exp-----*/
else if (strategy==6)
{
    assignd(D,trial,x1[i]);
    j=int (rnd_uni(&rnd_uni_init)*D);
    k=0;
    do
    {
        trial[j]=x1[c][j]+F*(x1[a][j]-x1[b][j]);
        if(trial[j]<=0.0)      trial[j]=1e-7;
        if(trial[2]>=0.5)    trial[2]=1.0;
        if(trial[2]<0.5)    trial[2]=0.0;
        j=(j+1)%D;
        k++;
    }
    while((rnd_uni(&rnd_uni_init))<CR && k<D);
}

/*-----de/best/1/exp-----*/
else if (strategy==7)
{
    assignd(D,trial,x1[i]);
    j=int (rnd_uni(&rnd_uni_init)*D);
    k=0;
    do
    {

```

```

        trial[j]=bestit[j]+F*(x1[a][j]-x1[b][j]);
        if(trial[j]<=0.0)        trial[j]=1e-7;
        if(trial[2]>=0.5)        trial[2]=1.0;
if(trial[2]<0.5)        trial[2]=0.0;
        j=(j+1)%D;
        k++;
    }
    while((rnd_uni(&rnd_uni_init))<CR && k<D);
}

/*-----de/best/2/exp-----*/

else if (strategy==8)
{
    assignd(D,trial,x1[i]);
    j=int (rnd_uni(&rnd_uni_init)*D);
    k=0;
do
    {
        trial[j]=bestit[j]+F*(x1[a][j]+x1[b][j]-x1[c][j]-x1[d][j]);
        if(trial[j]<=0.0)        trial[j]=1e-7;
        if(trial[2]>=0.5)        trial[2]=1.0;
        if(trial[2]<0.5)        trial[2]=0.0;
            j=(j+1)%D;
            k++;
    }
    while((rnd_uni(&rnd_uni_init))<CR && k<D);
}

/*-----de/rand/2/exp-----*/

else if (strategy==9)
{
    assignd(D,trial,x1[i]);
    j=int (rnd_uni(&rnd_uni_init)*D);
    k=0;
do
    {
        trial[j]=x1[e][j]+F*(x1[a][j]+x1[b][j]-x1[c][j]-x1[d][j]);
        if(trial[j]<=0.0)        trial[j]=1e-7;
        if(trial[2]>=0.5)        trial[2]=1.0;
        if(trial[2]<0.5)        trial[2]=0.0;
            j=(j+1)%D;
            k++;
    }
    while((rnd_uni(&rnd_uni_init))<CR && k<D);
}

/*-----de/rand-to-best/1/exp-----*/

else
{
    /* assignd(D,trial,x1[i]); */
    j=int (rnd_uni(&rnd_uni_init)*D);
    k=0;
do
    {
        trial[j]=trial[j]+F*(bestit[j]-trial[j])+F*(x1[a][j]-x1[b][j]);
        if(trial[j]<=0.0)        trial[j]=1e-7;
        if(trial[2]>=0.5)        trial[2]=1.0;
        if(trial[2]<0.5)        trial[2]=0.0;
            j=(j+1)%D;
            k++;
    }
    while((rnd_uni(&rnd_uni_init))<CR && k<D);
}

pen=0.0;

lhs1=0.9*(1-exp(-0.5*trial[0]*10))-2*(trial[2]);
if(lhs1>0.0)
{
    pen=lhs1*100.0;
}

```

```

cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
    for (j=0;j<D;j++)
        x2[i][j]=trial[j];
    cost[i]=cost_trial;
    if(cost_trial<costmin)
    {
        costmin=cost_trial;
        imin=i;
        assignd(D,best,trial);
    }
}
else for (j=0;j<D;j++)
    x2[i][j]=x1[i][j];
continue;
}

lhs2=0.8*(1-exp(-0.4*trial[1]*10))-2*(1-(trial[2]));
if(lhs2>0.0)
{
    pen=lhs2*100.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {
            costmin=cost_trial;
            imin=i;
            assignd(D,best,trial);
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
    continue;
}

lhs3=10*trial[0]-(10*trial[2]);
if(lhs3>0.0)
{
    pen=lhs3*100.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {
            costmin=cost_trial;
            imin=i;
            assignd(D,best,trial);
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
    continue;
}

lhs4=10*trial[1]-(10*(1-trial[2]));
if(lhs4>0.0)
{
    pen=lhs4*100.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)

```

```

        {
            costmin=cost_trial;
            imin=i;
            assignd(D,best,trial);
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
continue;
}
if(lhs1<=0.0 && lhs2<=0.0 && lhs3<=0.0 && lhs4<=0.0)
{
    cost_trial=evaluate(trial, &nfe);
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {
            costmin=cost_trial;
            imin=i;
            assignd(D,best,trial);
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
    continue;
}
} /***** end of for loop *****/
assignd(D,bestit,best);
for (i=0;i<NP;i++)
{
    for (j=0;j<D;j++)
        x1[i][j]=x2[i][j];
}

costmax=cost[0];
for(i=1;i<NP;i++)
{ if(costmax<cost[i])
costmax=cost[i];
}
costmin=cost[0];
for(i=1;i<NP;i++)
{ if(costmin>cost[i])
costmin=cost[i];
}

if((costmax-costmin)<0.000001)
    break;
count++;
} /***** end of while loop *****/

    end = clock();
for(i=0;i<NP;i++)
{
    for(j=0;j<D;j++)
    { if(j==2) printf("u[%d]=%lf",j,x1[i][j]); else
      printf("u[%d]=%lf",j,x1[i][j]*10);
    }
    printf("cost[%d]=%lf    ",i,cost[i]);
}
printf("NFE=%ld  seed=%d\n",nfe,seed);
printf("The time was: %f\n", (end - start) / CLK_TCK);

printf("would you like to exit? press Y for exit");
/* ch=getch();
if(ch=='y' || ch=='Y') { printf("exited"); exit(1);} */

fout=fopen("\\result10.xls","a+");
/* fprintf(fout,"nThe out put is\n:");
for(i=0;i<NP;i++)
{ if(i%10==0)
{
    fprintf(fout, "x1=%f  x2=%f  x3=%f  x4=%f  ",x1[i][0],x1[i][1],x1[i][2],x1[i][3]);
}
}
}

```

```

        fprintf(fout, "cost[%d]=%f \n",i,cost[i]);
    }
} */
fprintf(fout, "%d    %ld ",strategy,nfe);
fprintf(fout, "%f    ", (end - start) / CLK_TCK);
fprintf(fout, "%d    %d    %f    %f    %f    %f\n",count,seed,F,CR,costmax,costmin);
fclose(fout);

} /***** end of DE() *****/

void main()
{
    int i,j;float sfactor=0.0;
    for(j=0;j<10;j++)
    {
        sfactor=sfactor+0.1;
        for(i=1;i<=100;i++)
        {
            DE(sfactor);
        }
    }
} /***** end of main() *****/

```