The main code of SLTLBO algorithm for sloving dynamic economic load dispatch.

1. Setting parameters of dynamic economic load dispatch, such as number of nodes, line loss and economic cost coefficient .etc.

```
%%%%%%%%%%%%%%%% setting parameters %%%%%%%%%%%%%%%%%
global TestNode LinelossRate Powerdemand Dimension Lowerbound Upperbound UR
DR Np T EnviF ProhibitZone LinelossRate15i0 LinelossRate1500
%%%%%%%%%%%%% 5 nodes and linelossRate %%%%%%%%%%%%%%%%%
TestNode5=[
0.0080   2.0   25    10   75    100   0.042   30   30;
0.0030   1.8   60    20   125   140   0.040   30   30;
0.0012   2.1   100   30   175   160   0.038   40   40;
0.0010   2.0   120   40   250   180   0.037   50   50;
0.0015   1.8   40    50   300   200   0.035   50   50];

LinelossRate5=1e-4*[0.49 0.14 0.15 0.15 0.20;
0.14 0.45 0.16 0.2 0.18;
0.15 0.16 0.39 0.1 0.12;
0.15 0.20 0.1 0.4 0.14;
0.2 0.18 0.12 0.14 0.35];
%%%%%%%%%%%%% 15 nodes and linelossrate %%%%%%%%%%%%%%%%%
TestNode15=[
0.000299   10.1   671   150 455 0 0 80 120;
0.000183   10.2   574   150 455 0 0 80 120;
0.001126    8.8   374    20 130 0 0 130 130;
0.001126    8.8   374    20 130 0 0 130 130;
0.000205   10.4   461   150 470 0 0 80 120;
0.000301   10.1   630   135 460 0 0 80 120;
0.000364    9.8   548   135 465 0 0 80 120;
0.000338   11.2   227    60 300 0 0 65 100;
0.000807   11.2   173    25 162 0 0 60 100;
0.001203   10.7   175    25 160 0 0 60 100;
0.003586   10.2   186    20  80 0 0 80 80;
0.005513    9.9   230    20  80 0 0 80 80;
0.000371   13.1   225    25  85 0 0 80 80;
0.001929   12.1   309    15  55 0 0 55 55;
0.004447   12.4   323    15  55 0 0 55 55];

LinelossRate15=1e-5*[
  1.4   1.2   0.7 -0.1 -0.3 -0.1 -0.1 -0.1 -0.3 -0.5 -0.3 -0.2   0.4   0.3 -0.1;
  1.2   1.5   1.3   0.0 -0.5 -0.2   0.0   0.1 -0.2 -0.4 -0.4   0.0   0.4   1.0 -0.2;
  0.7   1.3   7.6 -0.1 -1.3 -0.9 -0.1   0.0 -0.8 -1.2 -1.7   0.0 -2.6 11.1 -2.8;
-0.1   0.0 -0.1   3.4 -0.7 -0.4   1.1   5.0   2.9   3.2 -1.1   0.0   0.1   0.1 -2.6;
```

```
-0.3 -0.5 -1.3 -0.7    9.0    1.4 -0.3 -1.2 -1.0 -1.3    0.7 -0.2 -0.2 -2.4 -0.3;
-0.1 -0.2 -0.9 -0.4    1.4    1.6    0.0 -0.6 -0.5 -0.8    1.1 -0.1 -0.2 -1.7    0.3;
-0.1    0.0 -0.1    1.1 -0.3    0.0    1.5    1.7    1.5    0.9 -0.5    0.7    0.0 -0.2 -0.8;
-0.1    0.1    0.0    5.0 -1.2 -0.6    1.7 16.8    8.2    7.9 -2.3 -3.6    0.1    0.5 -7.8;
-0.3 -0.2 -0.8    2.9 -1.0 -0.5    1.5    8.2 12.9 11.6 -2.1 -2.5    0.7 -1.2 -7.2;
-0.5 -0.4 -1.2    3.2 -1.3 -0.8    0.9    7.9 11.6 20.0 -2.7 -3.4    0.9 -1.1 -8.8;
-0.3 -0.4 -1.7 -1.1    0.7    1.1 -0.5 -2.3 -2.1 -2.7 14.0    0.1    0.4 -3.8 16.8;
-0.2    0.0    0.0    0.0 -0.2 -0.1    0.7 -3.6 -2.5    -3.4    0.1    5.4 -0.1 -0.4    2.8;
  0.4    0.4 -2.6    0.1 -0.2 -0.2    0.0    0.1    0.7    0.9    0.4 -0.1 10.3 -10.1 2.8;
  0.3    1.0 11.1    0.1 -2.4 -1.7 -0.2 0.5 -1.2 -1.1 -3.8 -0.4 -10.1 57.8 -9.4;
  -0.1 -0.2 -2.8 -2.6 -0.3    0.3 -0.8 -7.8 -7.2 -8.8 16.8 2.8 2.8 -9.4 128.3];

LinelossRate15i0=1e-4*[-1 -2 28 -1 1 -3 -2 -2 6 39 -17 -00 -32 67 -64];
LinelossRate1500=0.0055;

EnviF=[
0.0180 -0.805 80 0.6550 0.02846
0.0150 -0.555 50 0.5773 0.02446
0.0105 -1.355 60 0.4968 0.02270
0.0080 -0.600 45 0.4860 0.01948
0.0120 -0.555 30 0.5035 0.02075
];
%%%%%%%%%%%%%%%PEV %%%%%%%%%%%%%%%%%%%%%%%%
PEV=1125;
Powerdemand5=[410 435 475 530 558 608 626 654 690 704 720 740 704 690 654 580
558 608 654 704 680 605 527 463];
Powerdemand10=[1036 1110 1258 1406 1480 1628 1702 1776 1924 2072 2146 2220
2072 1924 1776 1554 1480 1628 1776 2072 1924 1628 1332 1184];
Powerdemand15=[2236 2240 2226 2236 2298 2316 2331 2443 2630 2728 2783 2785
2780 2830 2970 2950 2902 2803 2651 2584 2432 2312 2261 2254];
PEV_EPRI=PEV*[0.1 0.1 0.095 0.07 0.05 0.03 0.01 0.003 0.003 0.013 0.021 0.021
0.021 0.021 0.021 0.010 0.005 0.005 0.018 0.036 0.052 0.095 0.1 0.1];
PEV_OFFPEAK=PEV*[0.185 0.185 0.09 0.09 0.04 0.04 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.185 0.185];
PEV_PEAK=PEV*[0 0 0 0 0 0 0 0 0 0 0 0 0.185 0.185 0.185 0.185 0.09 0.09 0.04 0.04 0
0 0 0];
PEV_STO=PEV*[0.057 0.049 0.048 0.024 0.026 0.097 0.087 0.048 0.011 0.032 0.021
0.057 0.038 0.022 0.021 0.061 0.032 0.022 0.028 0.022 0.055 0.025 0.035 0.082];
Wind=[1.25 1.96 2.02 1.44 1.3 1.84 2.92 2.41 3.07 3.24 2.5 2.91 3.3 2.5 2.8 2.31 1.91
1.31 0.64    1.65 6.54 7.41 5.85 3.61];


ProhibitZone=[25 30 55 60;
45 50 80 90;
60 70 125 140;
95 110 160 180;
```

```
85 100 175 200];
T=24;

TestNode=TestNode15;
LinelossRate=LinelossRate15;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
```

2. Setting the parameters and matrix of optimization

```
  %%%%%%%%%%%%%%% bounds of the parameters%%%%%%%%%%%%%%%%%%%
Lowerbound=TestNode(:,4)';
Upperbound=TestNode(:,5)';
Dimension=length(TestNode(:,1));
UR=TestNode(:,8)';
DR=TestNode(:,9)';
%%%%%%%%%%%%%% matrix setting %%%%%%%%%%%%%%%%%%%%%%%%
Np=20;
Gm=100;
pop=zeros(Np,T*Dimension);
AN=1;
N_Itr=1;
ge=zeros(AN,Gm);
optimal=zeros(AN,Dimension*T);
optimal_all=zeros(N_Itr,Dimension*T);
Fit_value=zeros(AN,1);
Fit_value_all=zeros(1,N_Itr);
Testoptimal=zeros(AN,30);
sum=zeros(1,AN);
temD=zeros(1,AN);
mean=zeros(1,AN);
Std=zeros(1,AN);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3. import the load data and population initialize

```
%%%%%%%%%%%%%%%%%% load data %%%%%%%%%%%%%%%%%%%%%%%%%%%
for opt=5:5
    Loadoption=opt;
    switch Loadoption
        case 1
            Filename='datapureload15u.mat';%emi
            Powerdemand=Powerdemand15;
        case 2
            Filename='dataepriload15u.mat';
            Powerdemand=Powerdemand15+PEV_EPRI;
```

```
        case 3
            Filename='dataoffpload15u.mat';
            Powerdemand=Powerdemand15+PEV_OFFPEAK;
        case 4
            Filename='datapeakload15u.mat';
            Powerdemand=Powerdemand15+PEV_PEAK;
        case 5
            Filename='datastoload15u.mat';
            Powerdemand=Powerdemand15+PEV_STO;
    End
    %%%%%%%%%%% population initialize %%%%%%%%%%%%%%%%%%
            for k=1:N_Itr
                for t=1:T
                    pop(:,((t-
                        1)*Dimension+1):(t*Dimension))=rand(Np,1)*(Upperbound
                        -Lowerbound)+ones(Np,1)*Lowerbound;
                end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

4. Carry out constraints used the “ConstrainedELD.m” file, applied the SLTLBO algorithm to optimize the economic load used the “SLTLBOdynamic.m” file and compared with other algorithms.

```
    %%%%%%%%%%%%% optimization %%%%%%%%%%%%%%%%%%%%
        [pop,Lowerbound_d,Upperbound_d]=ChenkRampRateLimit(pop,Np);

         pop=CheckDemandConstraints(pop,Lowerbound_d,Upperbound_d,Np);

        disp('                    slTLBO                    ')
            [ge(1,:),optimal(1,:),Fit_value(1,:)]=SLTLBOdynamic(Gm,pop,@f);

            optimal_all(k,:)=optimal(2          ,:);
            Fit_value_all(k)=Fit_value(2,:);
            for j=1:AN
                Testoptimal(j,k)=ge(j,Gm);
                sum(j)=sum(j)+Testoptimal(j,k);
            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

5. Calculate the mean value, std and save result.

```
    %%%%%%%%%%%%%% mean and std %%%%%%%%%%%%%%%%%%
            for i=1:AN
                mean(i)=sum(i)/N_Itr;
```

```
                for j=1:N_Itr
                temD(i)=temD(i)+(Testoptimal(i,j)-mean(i))^2;
                end
                Std(i)=sqrt(temD(i));
            End
%%%%%%%%%%%%%%%%%% save %%%%%%%%%%%%%%%%%%%%%%
            save(Filename)

%%%%%%%%%%%%%% constraint function %%%%%%%%%%%%%%%%%%
function y=f(p)
        y=ConstrainedELD(p);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Paper: Yang Z, Kang L l, Niu Q, et al. A self-learning TLBO based dynamic economic/environmental dispatch considering multiple plug-in electric vehicle loads[J]. Journal of Modern Power Systems & Clean Energy, 2014, 2(4):298-307.