博弈论简单讲就是，A采取措施影响B的行为，B的行为影响A的决策，两者来回博弈，最终达到一个动态平衡。博弈论严格来讲只是一种解题方式。

以电动出租车与换电站为例，假设电动出租车及换电站均属于同一家公司，公司想通过换电站价格定价措施去控制目标区域内的出租车数量达到预期分布。

对于司机而言，有两个成本，一个是距离成本d，一个是支付成本p，支付成本即是换电池所支付的电价，我们可以设立权重因子a将两者合并构建为一个效用函数，司机会选择该函数最小的换电站更换电池，更换电池后司机一般会在周围开始接单

$$U_n = a_1 \cdot d_n^i + a_2 \cdot p_i$$

对于公司而言，目标函数则是不同地区的出租车实际分布e与期望分布E的绝对差之和，公司通过调整价格去影响司机的选择，从而调整司机在不同区域的分布

$$\min_{\vec{P}} \sum_{i=1}^{m} |e_i - E_i|$$

双层博弈论的模型分析

①第一阶段，充电站统计出各电动出租车的换电请求后，根据优化目标，制定价格策略
②第二阶段，电动出租车根据自身效用函数从所有换电站中选择出目标换电站进行跟换电池
③第一阶段和第二阶段交替往复进行，直到达到纳什均衡

算法设计步骤

算法：价格策略制定及充电站选择算法
Input：参数初始化
1. tage1：价格策略制定阶段
2. while $\left(\sum_{i=1}^{i=m}\lvert e_i - E_i \rvert < \varepsilon \quad or \quad t > T\right)$ do
3. t 为当前迭代次数，T 为总的迭代次数
4. $i \in \{1,2,...,m\}$，计算θ和$\delta_i(t)$
5. $p_i(t+1) = [p_i(t) + \theta \cdot \delta_i(t)]_{p\min}^{p\max}$
6. Stage2：充电站选择博弈阶段
7. for 对于每一个$n \in \{1,2,...,N\}$ do
8. 计算ET_n到每一个换电站的总成本，为每一个ET找到换电成本最小的换电站
9. end for
10. 如果没有ET改变目标充电站，那么
11. end while
Output：输出换电站价格和出租车的选择决策

代码

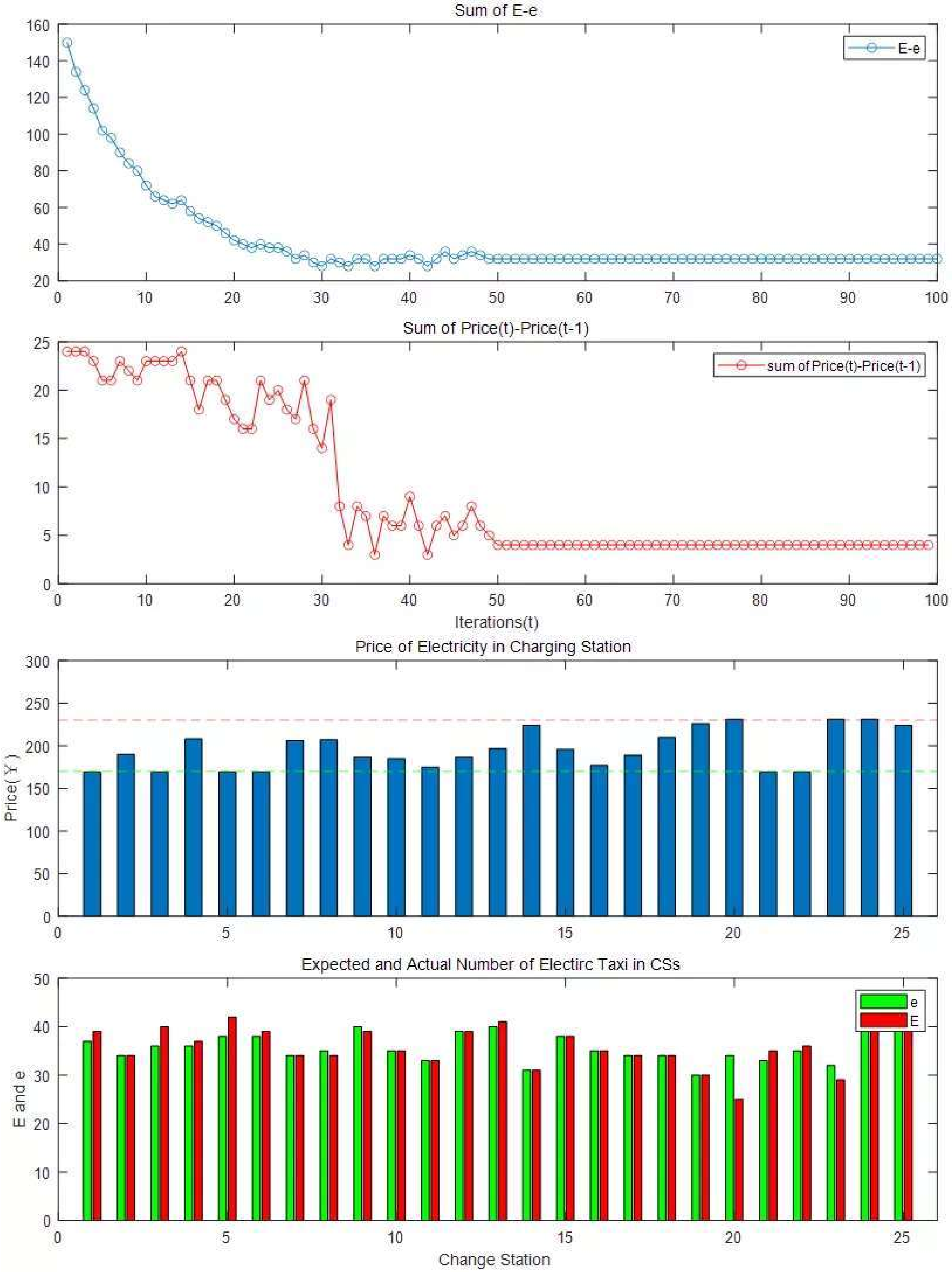
```
clear
setenv('BLAS_VERSION','')
n=900;%换电需求数
min_price=170;%换电价格范围
max_price=230;
A=normrnd(36,5,1,25);
E=fix(A);%初始期望，平均值为36，方差为5的高斯分布
a=sum(E)-n;
A=A-a/25;
E=fix(A);
b=sum(E)-n;
A=A-b/25;
E=fix(A);
a1=0.05;a2=0.95;%距离成本与换点价格权重
x=rand(n,1).*20000;%初始化需求车辆位置
y=rand(n,1).*20000;
H=[2,2;2,6;2,10;2,14;2,18%初始化换电站位置
6,2;6,6;6,10;6,14;6,18
10,2;10,6;10,10;10,14;10,18
14,2;14,6;14,10;14,14;14,18
```

```
18,2;18,6;18,10;18,14;18,18].*1000;
figure
plot(x,y,'r*')
hold on
plot(H(:,1),H(:,2),'bo')
legend('user','Change Station')
%% 计算期望与实际期望
D=[];%需求车辆到各换电站的需求比例
price=200.*ones(1,25);
for i=1:length(H)
for j=1:length(x)
        D(i,j)=a1*sqrt((H(i,1)-x(j))^2+(H(i,2)-y(j))^2)+a2*price(i);%总费
end
end
[d1,d2]=min(D);%选择最近距离换电站
C=tabulate(d2(:));%统计选择换电站次数
e=C(:,2);
err=sum(abs(E-e'));%期望差之和，即博弈对象
ER(1)=err;
J=[];
%% 博弈
for k=2:100
    j=0;
    for i=1:25
        if e(i)<E(i) && price(i)>=min_price
            price(i)=price(i)-1;
            j=j+1;
        end
        if e(i)>E(i) && price(i)<=max_price
            price(i)=price(i)+1;
            j=j+1;
        end
    end
    J=[J,j];
    DD=[];
    for i=1:length(H)
        for j=1:length(x)
            DD(i,j)=a1*sqrt((H(i,1)-x(j))^2+(H(i,2)-y(j))^2)+a2*price(i);
```

```
        end
    end
    [dd1,dd2]=min(DD);
    CC=tabulate(dd2(:));
    e=CC(:,2);
    err=sum(abs(E-e'));
    ER=[ER,err];
end

figure
plot(ER,'-o')
title('Sum of E-e')
set(gcf,'unit','normalized','position',[0.2,0.2,0.64,0.32])
legend('E-e')

figure
plot(J,'r-o')
title('Sum of Price(t)-Price(t-1)')
xlabel('Iterations(t)')
set(gcf,'unit','normalized','position',[0.2,0.2,0.64,0.32])
legend('sum of Price(t)-Price(t-1)')

figure
bar(price,0.5)
hold on
plot([0,26],[min_price,min_price],'g--')
plot([0,26],[max_price,max_price],'r--')
title('Price of Electricity in Charging Station')
ylabel('Price(￥)')
axis([0,26,0,300])
set(gcf,'unit','normalized','position',[0.2,0.2,0.64,0.32]);

figure
h=bar([e,E'],'gr');
set(h(1),'FaceColor','g'); set(h(2),'FaceColor','r');
axis([0,26,0,50])
title('Expected and Actual Number of Electirc Taxi in CSs')
ylabel('E and e')
```

```
set(gcf,'unit','normalized','position',[0.2,0.2,0.64,0.32]);
xlabel('Change Station')
legend('e','E')
```

结果


Sum of E-e
E-e
Sum of Price(t)-Price(t-1)
sum of Price(t)-Price(t-1)
Iterations(t)
Price of Electricity in Charging Station
Price(Y)
Expected and Actual Number of Electirc Taxi in CSs
e
E
E and e
Change Station