HW13

May 31, 2020

1 HW13

1.1 Alireza Darvishi 96109674

[1]:
```
import numpy as np
import cvxpy as cp
import matplotlib.pyplot as plt
```

1.1.1 A14.31. b

[2]:
```
b = np.array([400, 80, 400, 200, 400, 400, 80, 400, 100, 500])
v = np.array([500, 100, 500, 200, 700, 300, 120, 300, 150, 600])
n = 10
L = 4
rho_l = 0.2
rho_s = 0.3
C = 2300

s = cp.Variable(n)
g = cp.multiply( s , (v - b) / v )
N_l = cp.sum(g[:L])
N_s = cp.sum(g[L:])
base_constraints = [s >= 0,s <= v,cp.sum(s) == C ]
objective1 = cp.Minimize(rho_l*N_l+rho_s*N_s)
objective2 = cp.Minimize(rho_s*cp.pos(N_s+N_l))
objective3 = cp.Minimize(rho_l*cp.pos(N_s+N_l))
objective4 = cp.Minimize(0)

problem1 = cp.Problem(objective1,base_constraints+[N_l>=0 , N_s>= 0])
problem2 = cp.Problem(objective2,base_constraints+[N_l<=0 , N_s>= 0])
problem3 = cp.Problem(objective3,base_constraints+[N_l>=0 , N_s<= 0])
problem4 = cp.Problem(objective4,base_constraints+[N_l<=0 , N_s<= 0])

(t1,t2,t3,t4)=(problem1.solve(),problem2.solve(),problem3.solve(),problem4.
↪solve())
```

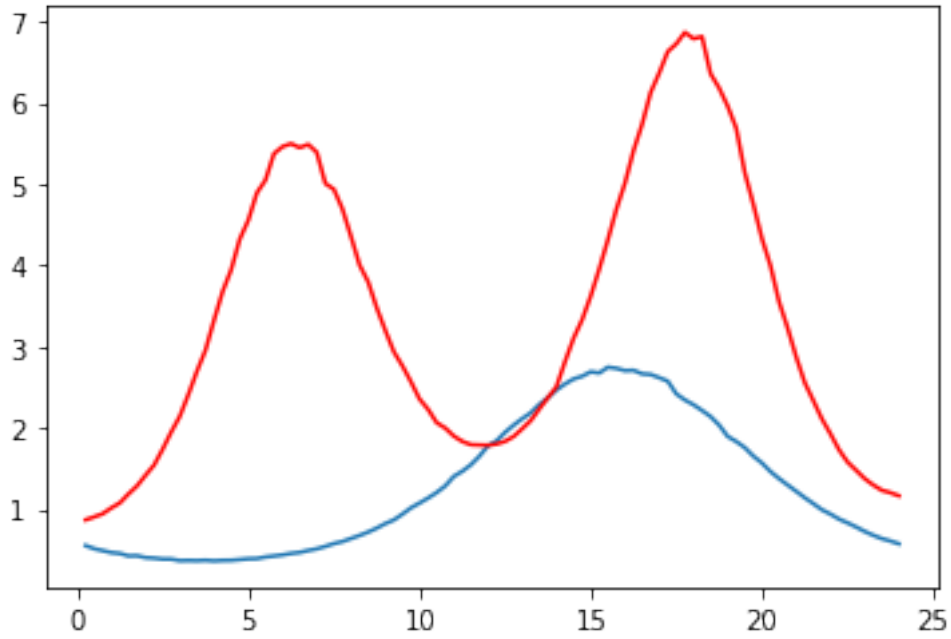
```
idx=np.argmin((t1,t2,t3,t4))
if(idx==0):
    problem1.solve()
elif(idx==1):
    problem2.solve()
elif(idx==2):
    problem3.solve()
else:
    problem4.solve()
print("best value is:",min((t1,t2,t3,t4)))
print("best s is:",np.round(s.value,2))
```

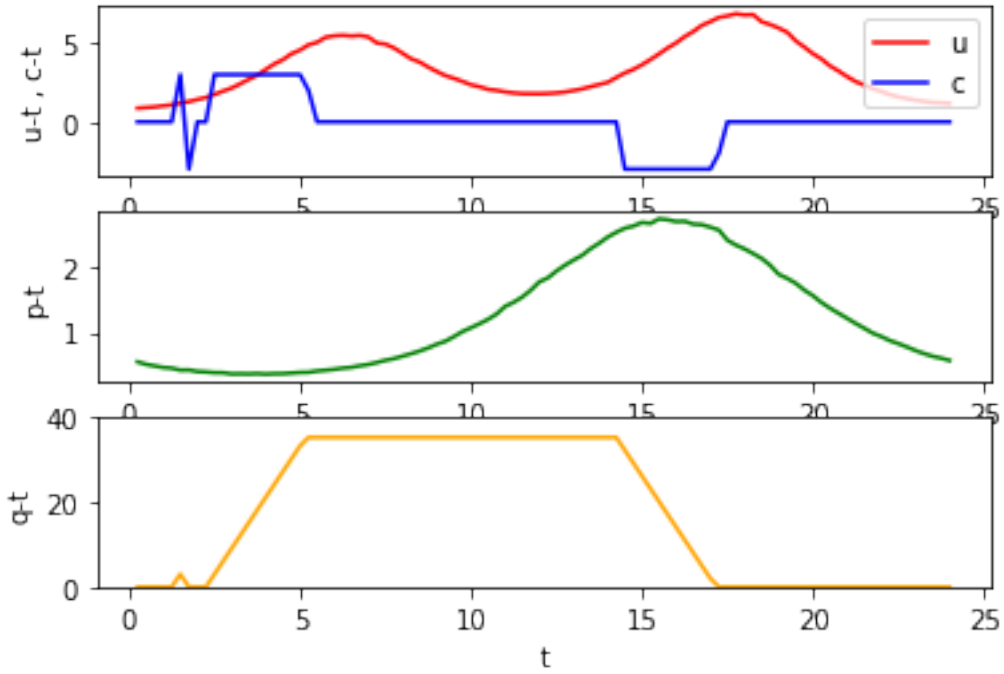
```
best value is: 16.000000002581334
best s is: [419.43  61.14 419.43 200.     0.   300.     0.   300.     0.   600.
]
```

1.1.2 A17.9. b

[3]:
```
from storage_tradeoff_data import *
c= cp.Variable(T)
objective = cp.Minimize(p.T * (u.reshape(T,)+c))
Q = cp.Parameter(value = 35)
C = cp.Parameter(value = 3)
D= cp.Parameter(value = 3)
constraint = [-D*np.ones(T)<=c , c<= C*np.ones(T), cp.cumsum(c)<=Q*np.ones(T),␣
 ↪0<=cp.cumsum(c) , cp.sum(c)==0 ]
problem = cp.Problem(objective,constraint)
problem.solve()

plt.figure(1)
ts = np.linspace(1, T, num=T).reshape(T,1)/4
plt.subplot(3,1,1)
plt.plot(ts, u, 'r');
plt.plot(ts, c.value, 'b');
plt.xlabel('t')
plt.ylabel('u-t , c-t')
plt.legend(['u','c'])
plt.subplot(3,1,2)
plt.plot(ts, p, 'g');
plt.xlabel('t')
plt.ylabel('p-t')
plt.subplot(3,1,3)
plt.plot(ts, cp.cumsum(c).value, 'orange');
plt.xlabel('t')
plt.ylabel('q-t')
plt.ylim((0, 40));
```





[4]:
```
Qs=np.linspace(0 , 150 , 151)
values = np.zeros((151 , 2))
for q in Qs:
```

```
    C.value = 3
    D.value = 3
    Q.value = q
    problem.solve()
    values[int(q) , 0]= problem.value
    C.value = 1
    D.value = 1
    problem.solve()
    values[int(q) , 1]= problem.value
plt.plot(Qs,values[:,0],Qs , values[:,1]);
plt.ylabel("cost");
plt.xlabel("Q");
```