EE 571 Introduction to Convex Optimization

Homework 7

Madhura Kulkarni

Problem 1: Optimal Activity Levels

Method 1: Formulating the problem as a Linear Program

maximize $1^T u$

subject to $x \geq 0$

$$Ax \leq c^{max}$$

$$p_j x_j \geq u_j$$

$$p_j q_j + p_j^{disc} \left(x_j - q_j\right) \geq u_j$$

The following script solves the aforementioned problem:

```
%Given:

%Resources consumed
A = [1 2 0 1;0 0 3 1;0 3 1 1;2 1 2 5;1 0 3 2];

%Maximum allowable limit for resource consumption
cmax = repmat(100,5,1);

%Basic price
p = [3;2;7;6];

%Discount price per quantity
pdisc = [2;1;4;2];

%Quantity discount level
q = [4;10;5;10];

%% Method 1: Formulating the given problem as a Linear Program

echo on

cvx_begin

    variable u(4)
    variable x(4)

    maximize (sum(1.'*u))
    subject to
            x >= 0;
            (A * x <= cmax);
            (p.* x) >= u;
            ((p.*q)+(pdisc.*(x - q))) >= u;
```

```
cvx_end

%Display the activity levels
x

%Total revenue
r = min((p.*x),(p.*q)+(pdisc.*(x - q)))
total = sum(r)

%Revenue associated with each level
avg_price = r./x

echo off
```

Output for this script is as follows:

```
------------------------------------------------------------------

-----------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +192.5


x =

    4.0000
   22.5000
   31.0000
    1.5000


r =

   12.0000
   32.5000
  139.0000
    9.0000


total =

  192.5000
```

```
avg_price =

    3.0000
    1.4444
    4.4839
    6.0000
```

The maximum total revenue = 192.499999943975

Method 2: Formulating the given problem as a Convex problem

maximum $\sum_{j=1}^{n} r_j x_j$

subject to $Ax \leq c^{max}$

$x \geq 0$

The following script solves the aforementioned problem:

```
%Given:

%Resources consumed
A = [1 2 0 1;0 0 3 1;0 3 1 1;2 1 2 5;1 0 3 2];

%Maximum allowable limit for resource consumption
cmax = repmat(100,5,1);

%Basic price
p = [3;2;7;6];

%Discount price per quantity
pdisc = [2;1;4;2];

%Quantity discount level
q = [4;10;5;10];

%% Method 2 : Formulating the given problem as a Convex Problem

echo on

cvx_begin
    variable x(4)

    maximize (sum(min(p.*x,(p.*q + pdisc.*(x - q)))))
    subject to
            (A*x) <= cmax
```

```
            x >= 0
cvx_end

%Display the activity levels
x

%Total revenue
r = min((p.*x),(p.*q)+(pdisc.*(x - q)))
total = sum(r)

%Revenue associated with each level
avg_price = r./x

echo off
```

Output for this script is as follows:

```
------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +192.5


x =

    4.0000
   22.5000
   31.0000
    1.5000


r =

   12.0000
   32.5000
  139.0000
    9.0000


total =

  192.5000
```

```
avg_price =

    3.0000
    1.4444
    4.4839
    6.0000
```

The maximum total revenue = 192.499999943975

Comparing the outputs of Method 1 and Method 2, we can conclude that both the formulations are equivalent.

Problem 2: Reformulating constraints in CVX

The script for reformulating and solving the problem is as follows:

```
%Problem 2

%%(a) norm([x + 2y, x - y]) = = 0

%The equivalent reformulation is as follows:
        x == 0;
        y == 0;

%% (b) square(square(x + y)) <= x - y

%The equivalent reformulation is as follows:
        (x + y)^4 <= x - y;

%% (c) 1/x + 1/y <= 1; x >= 0; y >= 0

%The equivalent reformulation is as follows:
            inv_pos(x) + inv_pos(y) <= 1;

%% (d) norm([max(x,1), max(y,2)]) <= 3*x + y

%The equivalent reformulation is as follows:
            norm([u;v]) <= 3*x + y;
            max(x,1) <= u;
            max(y,2) <= v;

%% (e) x*y >= 1; x>= 0; y>= 0

%The equivalent reformulation is as follows:
            x >= inv_pos(y);
            x >= 0;
            y >= 0;

%% (f) (x + y)^2/sqrt(y) <= x - y + 5

%The equivalent reformulation is as follows:

            quad_over_lin(x + y,sqrt(y)) <= x - y + 5;

%% (g) x^3 + y^3 <= 1; x >= 0; y >= 0

%The equivalent reformulation is as follows:

            pow_pos(x,3) + pow_pos(y,3) <= 1;
```

```
%% (h) x+z <= 1 + sqrt(x*y - z^2); x>= 0; y>= 0

%The equivalent reformulation is as follows:

            x + z <= 1 + geo_mean([x - quad_over_lin(z,y),y]);

%% A convex problem solving these constraints:

echo on;

      cvx_begin
            variables x y u v z
            x == 0;
            y == 0;
            (x + y)^4 <= x - y;
            inv_pos(x) + inv_pos(y) <= 1;
            norm([u;v]) <= 3*x + y;
            max(x,1) <= u;
            max(y,2) <= v;
            x >= inv_pos(y);
            x >= 0;
            y >= 0;
            quad_over_lin(x + y,sqrt(y)) <= x - y + 5;
            pow_pos(x,3) + pow_pos(y,3) <= 1;
            x + z <= 1 + geo_mean([x-quad_over_lin(z,y),y]);
       cvx_end

echo off
```

The output for the test problem is as follows:

```
------------------------------------------------------------------------

----------------------------------------------------------------
Status: Infeasible
Optimal value (cvx_optval): +Inf
```

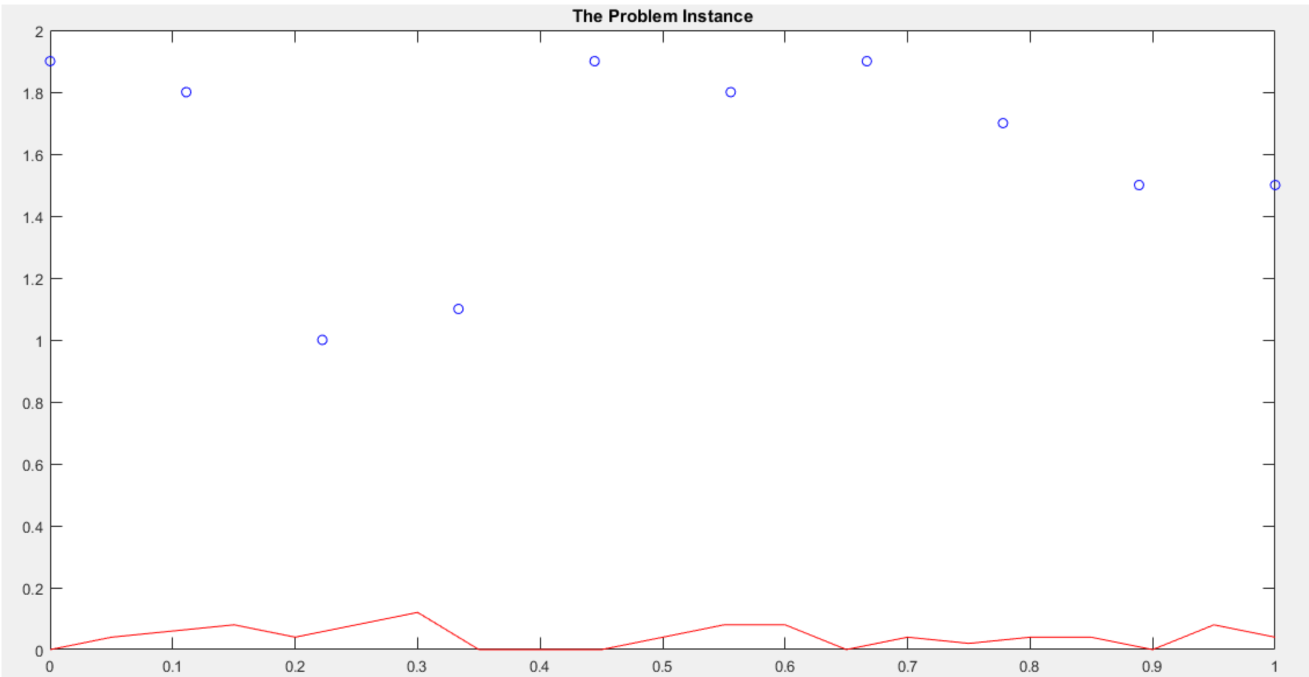
This shows that the above reformulation is acceptable to CVX and is processed without any errors.

Problem 3: The illumination problem

Given:

1. Number of lamps = 10
2. Number of patches = 20
3. Lamp power = p

Using the 'illum_data' file we observe the actual problem instance of randomly placing 'm' lamps and the illumination intensity for 'n' patches.



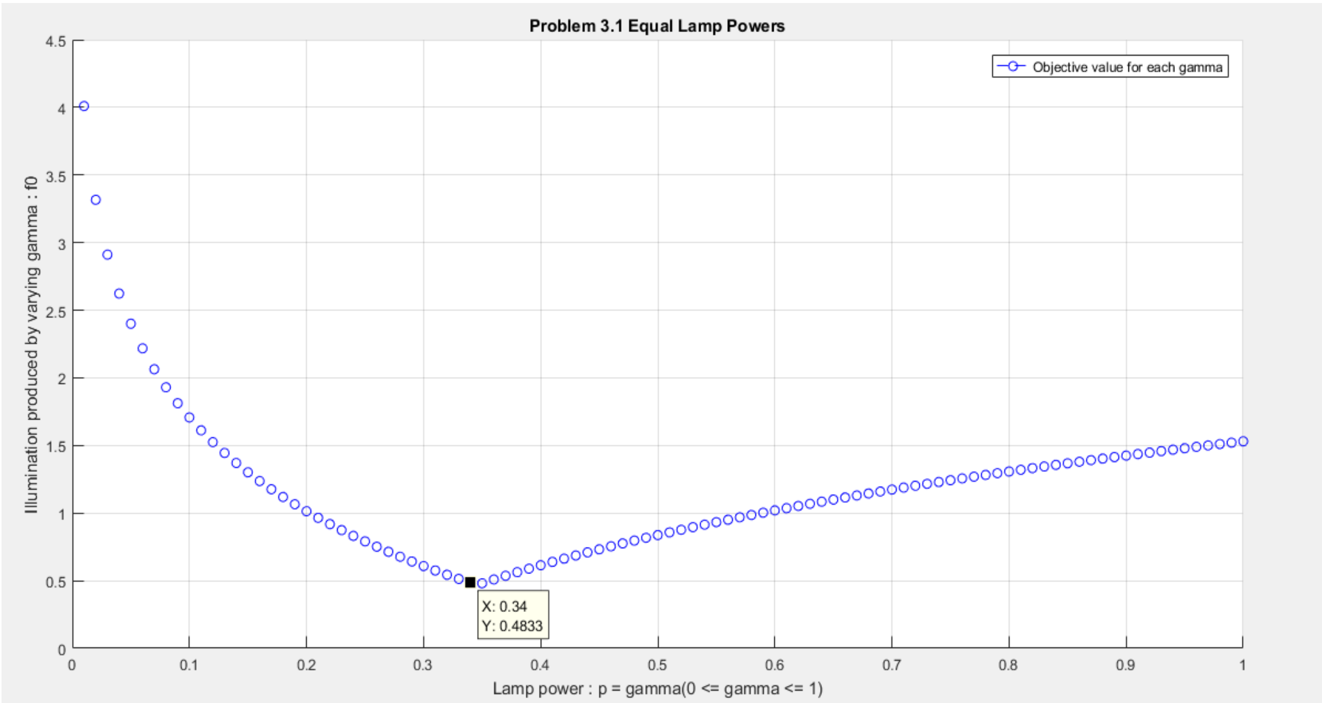
Problem 3.1: Equal Lamp Powers

For a variable 'γ' in the range [0,1] we calculate the optimal power 'p (γ)' which further determines the illumination defined by the function '$f_0(p)$'.

The script for the calculation is as follows:

```
%% Problem 3.1 Equal lamp powers

figure(2);
hold on;

%Choosing a value for gamma between 0 and 1
for gamma = 0:0.01:1

    %Lamp power
    p01 = repmat(gamma,m,1);

    %Objective value for each gamma
    f01 = max(abs(log(A*p01)));

    disp(f01);

    grid on;
    plot(gamma,f01,'--bo');
    xlabel('Lamp power : p = gamma(0 <= gamma <= 1)');
    ylabel('Illumination produced by varying gamma : f0');
    title('Problem 3.1 Equal Lamp Powers');
    legend('Objective value for each gamma');

end

hold off;
```

The plot of 'f_0 vs p' is as follows:



From the plot above, it is seen that for γ = 0.34 we find the minimum value of the objective function f_0 = 0.4833.

Problem 3.2 Least Squares with saturation

The given least squares problem:

minimize $\sum_{k=1}^{n}(a_k^T p - 1)^2 = ||Ap - 1||_2^2$

The following script solves the aforementioned problem:

```
%% Problem 3.2 Least Squares with saturation

    b02 = ones(n,1);

    %The optimal point for the Saturated Least Squares problem
    p02 = A \ b02;

 %Comparing the values of the optimal solutions and setting them to
zero or one
 for i = 1: length(p02)

     %If the value of the least square solution is negative, set it
to zero
     if(p02(i) < 0)
        p02(i) = 0;

     %If the value of the least square solution is greater than one,
set it to one
     else if(p02(i) > 1)
             p02(i) = 1;
        end
    end
 end

%The optimal value for the Saturated Least Squares problem
f02 = max(abs(log(A*p02)))
```

The condition is such –

i. For any negative value of 'p_i' the value must be set to zero.
ii. For any value of 'p_i' greater than 1, must be set to one.

The optimal values of p:

p02

10x1 double

	1
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	1
9	0
10	1

The optimal value of the objective function ‘f_0’ = 0.862783558711942

Problem 3.3 Regularized least squares

Given least squares problem:

minimize $\sum_{k=1}^{n}(a_k^T p - 1)^2 + \rho\sum_{j=1}^{m}(p_j - 0.5)^2 = ||Ap - 1||_2^2 + \rho||p - (1/2)1||_2^2$

The following script solves the aforementioned problem:

```
%% Problem 3.3 Regularized Least Squares

 b31 = ones(n,1);

 b32 = ones(m,1);

 %Varying the value of 'rho' such that the values in 'A' are in the
range
 %of [0,1]
for rho = 0.1:0.01:10

   p31 = [A; sqrt(rho)*eye(m)]\[b31;sqrt(rho)*0.5*b32];

   t31 = zeros(m,1);
   t32 = ones(m,1);

   if(t31 <= p31 <= t32)

       break;

   end
end

%The optimal point for the Regularised Least Squares problem
p03 = p31;

%The optimal value for the Regularised Least Squares problem
f03 = max(abs(log(A*p03)));
```

The condition is such –

i. Vary the value of ‘ρ’ such that the values of A lie between 0 and 1.

The optimal value of ‘p’:

p03

10x1 double

	1
1	0.6093
2	0.5762
3	0.1131
4	-0.1663
5	0.5064
6	0.4683
7	0.4997
8	0.4477
9	0.3926
10	0.4746

For such an optimal point, the objective function f_0 = 0.457930831501868

Problem 3.4 Chebyshev approximation

Given problem:

minimize $max_{k=1,\ldots,n} \, |a_k^T p - 1 = \, ||Ap - 1||_\infty$
subject to $0 \le p_j \le 1, \; j = 1, \ldots, m$

The following script solves the aforementioned problem:

```
%% Problem 3.4 Chebyshev approximation

b04 = ones(n,1);
t41 = zeros(m,1);
t42 = ones(m,1);

cvx_begin

    variable p41(10)

        minimize (norm((A*p41) - b04,inf))
        subject to
                t41 <= p41 <= t42

cvx_end

%The optimal point for the Chebyshev Approximation problem
p4 = p41

%The optimal value for the Chebyshev Approximation problem
f04 = max(abs(log(A*p4)))
```

The optimal values of 'p':

p4

10x1 double

	1
1	1.0000
2	0.1165
3	1.6902e-08
4	1.1228e-08
5	1.0000
6	6.5023e-07
7	1.0000
8	0.0249
9	1.2349e-07
10	1.0000

The optimal point p = 0.342838

The optimal value of the objective function 'f_0' = 0.419824202931886

Problem 3.5 Exact Solution

Given convex problem:

minimize $max_{k=1,..,n}\ max(a_k^T p, 1/a_k^T)$
subject to $0 \leq\ p_j\ \leq 1\ \ j = 1, . . , m$

The following script solves the aforementioned problem:

```
%% Problem 3.5 Exact solution

t51 = zeros(m,1);
t52 = ones(m,1);

cvx_begin

    variable p51(10)

        minimize (max([(A*p51);inv_pos(A*p51)]))
            subject to
                    t51 <= p51 <= t52

cvx_end

%The optimal point for the Exact solution
p5 = p51

%The optimal value for the Exact solution
f05 = max(abs(log(A*p5)))
```

The optimal points 'p':

p5

10x1 double

	1
1	1.0000
2	0.2023
3	6.8254e-09
4	4.5504e-09
5	1.0000
6	2.6841e-07
7	1.0000
8	0.1882
9	4.9365e-08
10	1.0000

The optimal point p = 1.42971
The optimal value 'f_0' = 0.357474323576440