Data Mining
TP9 - Optimisation sous contraintes

Rémi MUSSARD - Thomas ROBERT

1 Premier essai avec CVX

CVX est une bibliothèque de calcul Matlab permettant de résoudre des problèmes d'optimisation sous contraintes par une méthode itérative.

On teste d'abord CVX sur un problème simple de minimisation sous contraintes.

On donne a CVX la variable à optimiser ($\theta \in \mathbb{R}^2$) en minimisant la fonction objectif ($\theta^\top P\theta + \theta^\top q$) et en respectant les contraintes ($\theta \leq l$ et $\theta \geq u$).

CVX calcule ensuite la solution optimale par une méthode itérative.

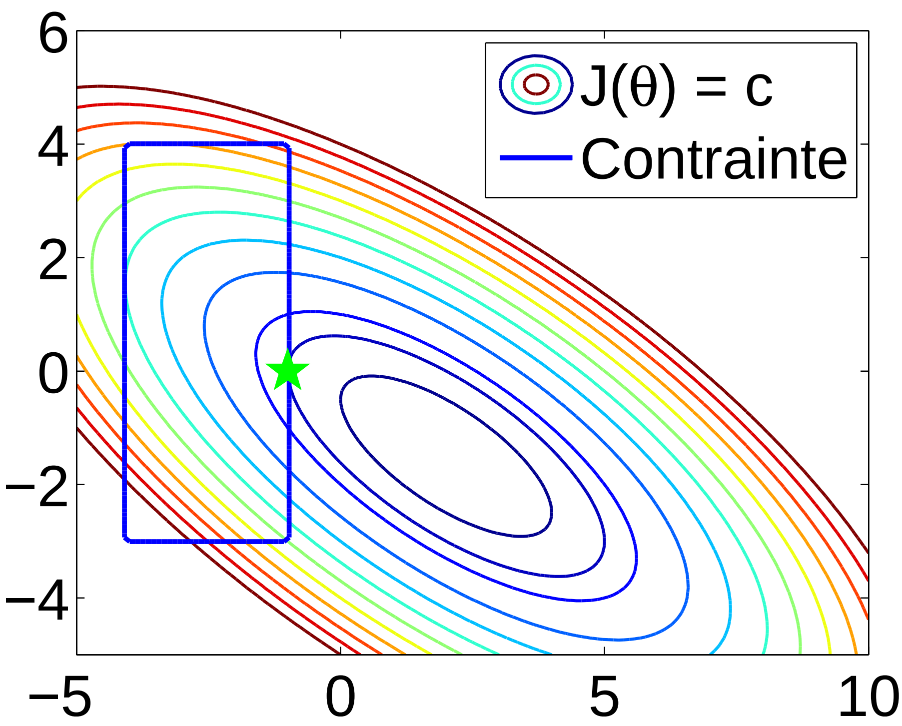
La solution obtenue semble bien être le minimum delimité par les contraintes.

```
clear all
close all
clc

% Tracé de la fonction objectif et de
N = 150;
x = linspace(-5, 10, N);
y = linspace(-5, 6, N);
[X,Y] = meshgrid(x, y);
x = reshape(X,N*N,1);
y = reshape(Y,N*N,1);
J = x.^2 + 2*y.^2 +2*x.*y - x + 2*y;
[c, h]=contour(X, Y, reshape(J, N,N), [-0.5  2 4:4:40], 'linewidth', 1.25);
%clabel(c,h);
hold on

ineq1 = (-4 <= x) & (x <=-1);
ineq2 = (-3 <= y) & (y <= 4);
ineq = ineq1 & ineq2;
hold on
[c,h]=contour(X, Y, reshape(ineq, N,N), [0 0], 'b', 'linewidth', 2);
set(gca,'fontsize', 24)
legend('J(\theta) = c', 'Contrainte', 'fontsize', 14)

% Resolution du probleme par cvx
% paramètre du problème
P=[1 1;1 2];
q = [-1; 2];
l = [-4;-3];
u = [-1; 4];

% Probleme
% minimize   1/2 theta'*P*theta + q'*theta
%              s.t.   l_i <= theta_i <= u_i

n = size(P, 1); % nombre de variables
fprintf( 'Calcul de la solution par CVX ... \n\n');

cvx_begin
    cvx_precision best
```

```
    variable theta(n); % declarer que theta est la variable du prob, vecteur de taille n

    minimize ( quad_form(theta,P) + q'*theta) % fonction objectif

    subject to

        theta >= l; % contraintes 1 (bornes inf)
        theta <= u; % contraintes 2 (bornes sup)
cvx_end

fprintf('\n\n Fait ! \n');

% affichage resultats
fprintf('\n\n Solution obtenue : \n');
disp(theta);

% trace de la solution du probleme
plot(theta(1), theta(2), 'p', 'markersize', 18, 'markeredgecolor', 'g', 'markerfacecolor', 'g')
```

```
Warning: Ignoring extra legend entries.
Calcul de la solution par CVX ...


Calling SDPT3 4.0: 8 variables, 3 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------


 num. of constraints =  3
 dim. of socp   var  =  4,   num. of socp blk  =  1
 dim. of linear var  =  4
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|4.8e-01|1.7e+00|4.2e+02| 1.000000e+02  0.000000e+00| 0:0:00| chol  1  1
 1|1.000|0.909|7.4e-07|1.7e-01|7.2e+01| 4.405662e+01 -6.324685e+00| 0:0:01| chol  1  1
 2|0.968|1.000|5.1e-07|2.1e-03|1.1e+01| 1.935962e+00 -8.629965e+00| 0:0:01| chol  1  1
 3|0.960|0.789|7.6e-08|6.0e-04|1.5e+00|-4.594180e+00 -6.066299e+00| 0:0:01| chol  1  1
 4|0.829|1.000|5.0e-08|2.1e-05|3.9e-01|-5.250242e+00 -5.641430e+00| 0:0:01| chol  1  1
 5|0.987|0.981|1.2e-09|2.4e-06|6.1e-03|-5.496178e+00 -5.502196e+00| 0:0:01| chol  1  1
 6|0.982|0.937|1.7e-10|3.5e-07|2.2e-04|-5.499883e+00 -5.500090e+00| 0:0:01| chol  1  1
 7|0.991|0.925|3.5e-11|2.6e-08|1.1e-05|-5.499996e+00 -5.500006e+00| 0:0:01| chol  1  1
 8|0.984|0.960|5.0e-12|1.0e-09|3.6e-07|-5.500000e+00 -5.500000e+00| 0:0:01| chol  1  1
 9|1.000|0.993|1.1e-12|8.0e-12|1.4e-08|-5.500000e+00 -5.500000e+00| 0:0:02| chol  2  2
10|0.997|0.995|1.2e-12|7.9e-13|2.2e-10|-5.500000e+00 -5.500000e+00| 0:0:02| chol  2  2
11|1.000|0.997|7.8e-12|1.9e-14|4.9e-12|-5.500000e+00 -5.500000e+00| 0:0:02| chol  3  3
12|1.000|0.984|5.4e-11|8.8e-16|1.8e-13|-5.500000e+00 -5.500000e+00| 0:0:02| chol  3 * 3
13|1.000|0.974|2.4e-10|1.0e-16|8.4e-15|-5.500000e+00 -5.500000e+00| 0:0:02| chol * 4 * 5
  stop: primal infeas has deteriorated too much, 1.7e-08
14|1.000|0.748|2.4e-10|1.0e-16|8.4e-15|-5.500000e+00 -5.500000e+00| 0:0:02|
  lack of progress in infeas
-------------------------------------------------------------------
 number of iterations   = 14
 primal objective value = -5.50000000e+00
 dual   objective value = -5.50000000e+00
 gap := trace(XZ)       = 4.86e-12
```

```
 relative gap            = 4.05e-13
 actual relative gap     = 8.20e-12
 rel. primal infeas      = 7.81e-12
 rel. dual   infeas      = 1.86e-14
 norm(X), norm(y), norm(Z) = 4.6e+00, 4.3e+00, 5.9e+00
 norm(A), norm(b), norm(C) = 3.8e+00, 4.7e+00, 1.1e+01
 Total CPU time (secs)   = 1.83
 CPU time per iteration = 0.13
 termination code        = -7
 DIMACS: 9.3e-12  0.0e+00  2.5e-14  0.0e+00  8.2e-12  4.1e-13
-------------------------------------------------------------------
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +2

 Fait !

 Solution obtenue :
   -1.0000
   -0.0000
```



2 Résolution du problème

On résoud désormais notre problème d'optimisation grâce à CVX en passant par le problème dual, plus simple à exprimer.

Les calculs préparatoires ont permis de trouver le lien direct entre la solution du problème dual et la solution du problème primal. (En l'occurrence, $z = \mu^\top X$.

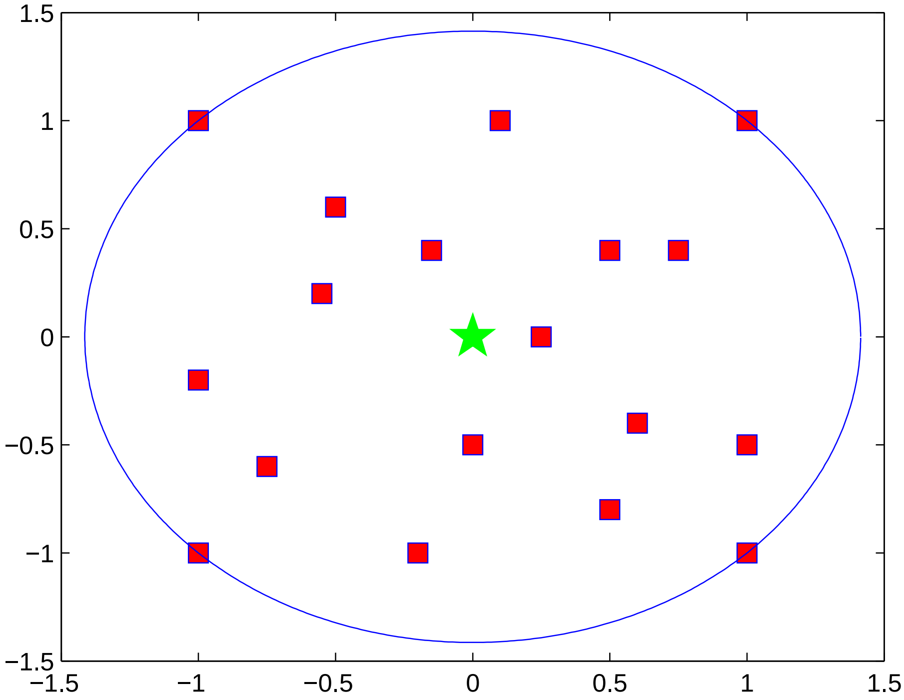
Notons que grâce aux variables duales du problème dual, on peut retomber sur le problème primal et trouver directement le rayon maximum, que l'on peut vérifier en traçant un cercle de ce rayon autour de la caserne.

On trouve une solution en $z = (0,0)^\top$) avec un rayon maximum de $R = 1,4142 = \sqrt{2}$

```
load maisons.mat

figure;
dots = plot(X(:,1), X(:,2), 'bs');
hold on;
set(dots(1),'MarkerFaceColor','red', 'markersize', 10);

% données
n = size(X,1);
H = X*X';
q = diag(H);

% minimisation
cvx_begin
    cvx_precision best

    variable('m(n)'); % declarer que theta est la variable du prob, vecteur de taille n

    dual variables de di;

    minimize ( m'*H*m - m'*q) % fonction objectif

    subject to
        di : m >= 0;
        de : sum(m) == 1;

        %theta >= l; % contraintes 1 (bornes inf)
        %theta <= u; % contraintes 2 (bornes sup)
cvx_end

% solution

z = m'*X;

plot(z(1), z(2), 'p', 'markersize', 18, 'markeredgecolor', 'g', 'markerfacecolor', 'g');

% rayon max

R = sqrt(abs(de) + z*z');
t = 0:0.01:2*pi;
plot(z(1)+cos(t)*R, z(2)+sin(t)*R, '-b');
```

```
Calling SDPT3 4.0: 26 variables, 8 equality constraints
------------------------------------------------------------

 num. of constraints =  8
 dim. of socp   var  =  8,   num. of socp blk  =  1
 dim. of linear var  = 18
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
```

```
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|7.4e+01|6.3e+00|1.8e+03|-1.562592e+02  0.000000e+00| 0:0:00| chol  1  1
 1|0.978|1.000|1.6e+00|5.9e-02|5.2e+01|-3.721184e+00 -1.263835e+01| 0:0:00| chol  1  1
 2|0.993|1.000|1.1e-02|5.9e-03|6.5e+00|-1.097104e+00 -7.369436e+00| 0:0:00| chol  1  1
 3|1.000|0.855|4.7e-07|3.6e-03|9.6e-01|-1.239573e+00 -2.192591e+00| 0:0:00| chol  1  1
 4|0.943|1.000|2.3e-08|5.9e-05|3.2e-01|-1.937989e+00 -2.259504e+00| 0:0:00| chol  1  1
 5|0.982|0.979|4.7e-09|7.0e-06|6.7e-03|-1.997557e+00 -2.004206e+00| 0:0:00| chol  1  1
 6|0.989|0.988|1.3e-09|6.7e-07|7.7e-05|-1.999972e+00 -2.000048e+00| 0:0:00| chol  1  1
 7|0.989|0.987|3.1e-10|8.6e-09|9.3e-07|-2.000000e+00 -2.000001e+00| 0:0:00| chol  1  1
 8|1.000|0.992|1.9e-12|1.3e-10|2.0e-08|-2.000000e+00 -2.000000e+00| 0:0:00| chol  1  1
 9|1.000|0.990|1.2e-14|2.3e-12|4.9e-10|-2.000000e+00 -2.000000e+00| 0:0:00| chol  1  1
10|1.000|0.999|6.1e-15|5.5e-13|1.5e-11|-2.000000e+00 -2.000000e+00| 0:0:00| chol  1  1
11|1.000|0.982|1.3e-16|2.1e-14|3.1e-13|-2.000000e+00 -2.000000e+00| 0:0:00| chol  1  1
12|1.000|0.961|2.1e-16|1.5e-15|2.0e-14|-2.000000e+00 -2.000000e+00| 0:0:00| chol  1  1
13|1.000|0.769|1.4e-17|5.3e-16|5.2e-15|-2.000000e+00 -2.000000e+00| 0:0:01| chol  1  1
14|0.519|0.756|1.9e-16|2.8e-16|2.7e-15|-2.000000e+00 -2.000000e+00| 0:0:01| chol  1  1
15|1.000|0.338|2.3e-16|2.8e-16|2.1e-15|-2.000000e+00 -2.000000e+00| 0:0:01| chol  1  1
16|0.412|0.503|1.7e-16|8.2e-17|1.4e-15|-2.000000e+00 -2.000000e+00| 0:0:01| chol
  linsysolve: Schur complement matrix not positive definite
  stop: difficulty in computing predictor directions
-------------------------------------------------------------------
 number of iterations   = 17
 primal objective value = -2.00000000e+00
 dual   objective value = -2.00000000e+00
 gap := trace(XZ)       = 1.39e-15
 relative gap           = 2.77e-16
 actual relative gap    = 2.66e-16
 rel. primal infeas     = 1.67e-16
 rel. dual   infeas     = 8.20e-17
 norm(X), norm(y), norm(Z) = 8.7e-01, 2.0e+00, 6.6e+00
 norm(A), norm(b), norm(C) = 6.7e+00, 2.4e+00, 7.4e+00
 Total CPU time (secs)  = 0.68
 CPU time per iteration = 0.04
 termination code       =  0
 DIMACS: 2.0e-16  0.0e+00  1.6e-16  0.0e+00  2.7e-16  2.8e-16
-------------------------------------------------------------------
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): -2

z =

   1.0e-10 *

    0.0546    0.1472
```


1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5