Lab08 - optimalization exercises, Pawel Drapiewski 18.04.2018 r.

Table of Contents

1st excercise from zadanieLP1.pdf 1
2nd excercise from zadanieLP1.pdf 3
3rd excercise from zadanieLP1.pdf 5
Funkcja is_linear 8

1st excercise from zadanieLP1.pdf

```
% How to mix wheat, soy and fishmeal to gain
% forage with minimal sufficent composition
%  x1 - wheat
%  x2 - soy
%  x3 - fishmeal


% [PL] Mamy doczynienienia z optymalizacj# liniow# poniewa# funkcja
 celu
% jak i funcke ogranicze# s# liniowe, czyli innymi s#owy zmienne
% optymalizacyjne wyst#puj# jedynie pierwszej pot#dze, a zadane
% wspó#czynniki przy tych zmiennych mog# jedynie regulowa# pochylenie
 tej#e
% p#aszczyzny nie zaburzaj#c jej liniowo#ci. Dodtakowo jest to problem
% liniowy, a nie affiniczny poniewa# uk#ady te przechodz# przez #rodek
% uk#adu wspó#rz#dnych (punkt O(0, 0, 0, 0)). We wszystkich
 przyk#adach
% b#dzie sprawdzana ta zale#no#c przy u#yciu napisanej funkcji
 is_lienar.
% Funkcja ta sprawdza 2 za#o#enia liniowo#ci:
%   za#1. f(ax) = a * f(x)
%   za#2. f(x+y) = f(x) + f(y)

% Sprawd# liniowo## funkcji celu oraz ograniacze#
fprintf('#### Check linearity ###########\n');
fprintf('f. celu jest liniowa? - %d \n', is_linear(@(x1, x2, x3, x4)
(300 * x1 + 500 * x2 + 800 * x3)));
fprintf('ogranicznie 0.8 * x1 + 0.3 * x2 + 0.1 * x3 jest liniowe? - %d
 \n', is_linear(@(x1, x2, x3, x4)(0.8 * x1 + 0.3 * x2 + 0.1 * x3)));
fprintf('ogranicznie 00.1 * x1 + 0.4 * x2 + 0.7 * x3 jest liniowe?
 - %d  \n', is_linear(@(x1, x2, x3, x4)(00.1 * x1 + 0.4 * x2 + 0.7 *
 x3)));
fprintf('ogranicznie 0.15 * x1 + 0.1 * x2 + 0.2 * x3 jest liniowe? -
 %d \n  ', is_linear(@(x1, x2, x3, x4)(0.15 * x1 + 0.1 * x2 + 0.2 *
 x3)));
fprintf('\n\n');
```

```
% Rozwi## równanie
cvx_begin
variables x1 x2 x3
minimize 300 * x1 + 500 * x2 + 800 * x3
subject to
    0.8 * x1 + 0.3 * x2 + 0.1 * x3 >= 0.3
    00.1 * x1 + 0.4 * x2 + 0.7 * x3 >= 0.7
    0.15 * x1 + 0.1 * x2 + 0.2 * x3 >= 0.1
    % the igredients quantity can't be negative
    x1 >= 0
    x2 >= 0
    x3 >= 0
cvx_end
fprintf('Optimal values is: wheat = %2.10f, soy = %2.4f, fishmeal =
 %2.4f \n', x1, x2, x3)

#### Check linearity ###########
f. celu jest liniowa? - 1
ogranicznie 0.8 * x1 + 0.3 * x2 + 0.1 * x3 jest liniowe? - 1
ogranicznie 00.1 * x1 + 0.4 * x2 + 0.7 * x3 jest liniowe?  - 1
ogranicznie 0.15 * x1 + 0.1 * x2 + 0.2 * x3 jest liniowe? - 1



Calling SDPT3 4.0: 6 variables, 3 equality constraints
------------------------------------------------------------

 num. of constraints =  3
 dim. of linear var  =  6
*******************************************************************
   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|3.4e+00|1.9e+00|5.9e+04| 1.600000e+04  0.000000e+00|
 0:0:00| chol  1  1
 1|1.000|1.000|1.8e-06|2.5e-04|7.3e+03| 7.648938e+03  3.345575e+02|
 0:0:00| chol  1  1
 2|0.899|1.000|2.7e-07|2.5e-05|7.6e+02| 1.141956e+03  3.825555e+02|
 0:0:00| chol  1  1
 3|1.000|0.755|4.5e-08|8.1e-06|3.6e+02| 1.130796e+03  7.720316e+02|
 0:0:00| chol  1  1
 4|0.880|0.983|1.6e-08|3.9e-07|4.4e+01| 8.650061e+02  8.215011e+02|
 0:0:00| chol  1  1
 5|0.991|0.965|4.1e-09|4.1e-08|3.9e+00| 8.375979e+02  8.336854e+02|
 0:0:00| chol  1  1
 6|0.985|0.984|3.3e-10|3.9e-09|6.3e-02| 8.353307e+02  8.352675e+02|
 0:0:00| chol  1  1
 7|0.989|0.989|8.4e-11|1.1e-10|7.0e-04| 8.352945e+02  8.352938e+02|
 0:0:00| chol  1  1
 8|0.989|0.989|2.8e-11|1.8e-11|7.7e-06| 8.352941e+02  8.352941e+02|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   =  8
 primal objective value =  8.35294122e+02
 dual   objective value =  8.35294114e+02
 gap := trace(XZ)       = 7.72e-06
 relative gap           = 4.62e-09
 actual relative gap    = 4.60e-09
 rel. primal infeas (scaled problem)   = 2.78e-11
 rel. dual     "        "       "      = 1.81e-11
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 9.8e-01, 1.1e+03, 1.1e+03
 norm(A), norm(b), norm(C) = 3.1e+00, 1.8e+00, 9.9e+02
 Total CPU time (secs)  = 0.09
 CPU time per iteration = 0.01
 termination code       =  0
 DIMACS: 2.9e-11  0.0e+00  2.2e-11  0.0e+00  4.6e-09  4.6e-09
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +835.294

Optimal values is: wheat = 0.0000000274, soy = 0.8235, fishmeal =
 0.5294
```

2nd excercise from zadanieLP1.pdf

```
% An optimal breakfest
%
% x1 - corn
% x2 - milk
% x3 - bread
%


% Sprawd# liniowo## funkcji celu oraz ogranicze#
fprintf('#### Check linearity ###########\n');
fprintf('f. celu jest liniowa? - %d \n', is_linear(@(x1, x2, x3, x4)
(0.15 * x1 + 0.25 * x2 + 0.05 * x3)));
fprintf('ogranicznie  70 * x1 + 121 * x2 + 65 * x3 jest liniowe? - %d
 \n', is_linear(@(x1, x2, x3, x4)( 70 * x1 + 121 * x2 + 65 * x3)));
fprintf('ogranicznie 107 * x1 + 500 * x2 jest liniowe?  - %d  \n',
 is_linear(@(x1, x2, x3, x4)(107 * x1 + 500 * x2)));
fprintf('ogranicznie 45 * x1 + 40 * x2 + 60 * x3 jest liniowe? - %d
 \n', is_linear(@(x1, x2, x3, x4)(45 * x1 + 40 * x2 + 60 * x3)));
fprintf('\n\n');


cvx_begin
```

```
variables x1 x2 x3
minimize 0.15 * x1 + 0.25 * x2 + 0.05 * x3 % minimize the cost of the
 complete set
subject to
    % calories level
    2000 <= 70 * x1 + 121 * x2 + 65 * x3 <= 2250
    % vitamin level
    5000 <= 107 * x1 + 500 * x2 <= 10000
    % sugar level
    0 <= 45 * x1 + 40 * x2 + 60 * x3 <= 1000

    % not more than 10 portion of each meal
    0 <= x1 <= 10
    0 <= x2 <= 10
    0 <= x3 <= 10
cvx_end

fprintf('Optimal values is: corn = %2.10f, milk = %2.4f, bread = %2.4f
 \n', x1, x2, x3)
```

```
#### Check linearity ###########
f. celu jest liniowa? - 1
ogranicznie  70 * x1 + 121 * x2 + 65 * x3 jest liniowe? - 1
ogranicznie 107 * x1 + 500 * x2 jest liniowe?  - 1
ogranicznie 45 * x1 + 40 * x2 + 60 * x3 jest liniowe? - 1
```

```
Calling SDPT3 4.0: 12 variables, 3 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------


 num. of constraints =  3
 dim. of linear var  = 12
*******************************************************************
   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|2.3e-01|3.5e+00|1.4e+06| 6.280000e+04  0.000000e+00|
 0:0:00| chol  1  1
 1|1.000|0.983|7.0e-04|5.9e-02|6.4e+04| 4.267608e+04 -8.717104e+00|
 0:0:00| chol  1  1
 2|0.914|0.841|4.4e-04|9.6e-03|8.4e+03| 5.589230e+03 -6.314266e+00|
 0:0:00| chol  1  1
 3|0.998|0.858|9.3e-05|1.4e-03|7.5e+02| 4.599770e+02 -4.655449e+00|
 0:0:00| chol  1  1
 4|1.000|0.648|8.2e-06|5.3e-04|3.5e+02| 2.408391e+02 -4.311891e+00|
 0:0:00| chol  1  1
 5|0.777|1.000|1.9e-06|1.6e-06|1.1e+02| 1.084931e+02 -3.960953e+00|
 0:0:00| chol  1  1
 6|0.981|1.000|3.6e-08|3.8e-07|2.1e+00|-1.829059e+00 -3.958900e+00|
 0:0:00| chol  1  1
 7|0.894|1.000|3.9e-09|7.2e-09|5.0e-01|-3.371168e+00 -3.867723e+00|
 0:0:00| chol  1  1
 8|1.000|0.981|2.6e-11|9.2e-10|1.6e-01|-3.600743e+00 -3.758795e+00|
 0:0:00| chol  1  1
 9|0.980|0.985|5.6e-12|1.9e-11|3.2e-03|-3.738358e+00 -3.741569e+00|
 0:0:00| chol  1  1
10|0.989|0.989|7.0e-14|1.4e-12|3.6e-05|-3.741145e+00 -3.741181e+00|
 0:0:00| chol  1  1
11|0.989|0.989|1.1e-14|1.0e-12|4.0e-07|-3.741176e+00 -3.741177e+00|
 0:0:00| chol  1  1
12|0.995|1.000|8.3e-15|9.0e-13|6.8e-09|-3.741176e+00 -3.741176e+00|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 12
 primal objective value = -3.74117647e+00
 dual   objective value = -3.74117647e+00
 gap := trace(XZ)       = 6.77e-09
 relative gap           = 7.99e-10
 actual relative gap    = 7.26e-10
 rel. primal infeas (scaled problem)   = 8.27e-15
 rel. dual     "        "       "      = 8.97e-13
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 1.9e-01, 1.3e+01, 4.5e+03
 norm(A), norm(b), norm(C) = 7.7e+02, 1.3e+00, 1.2e+04
 Total CPU time (secs)  = 0.09
 CPU time per iteration = 0.01
 termination code       =  0
 DIMACS: 8.6e-15  0.0e+00  1.0e-12  0.0e+00  7.3e-10  8.0e-10
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +3.74118

Optimal values is: corn = 6.5882353079, milk = 10.0000, bread =
 5.0588
```

3rd excercise from zadanieLP1.pdf

```
% Sprawd# liniowo## funkcji celu oraz ogranicze#
fprintf('#### Check linearity ###########\n');
fprintf('f. celu jest liniowa? - %d \n', is_linear(@(x1, x2, x3, x4)
(1000 * x1 + 199.9 * x2 - 5800 * x3 - 6300 * x4)));
fprintf('ogranicznie  x1 + x2 jest liniowe? - %d \n', is_linear(@(x1,
 x2, x3, x4)(x1 + x2)));
fprintf('ogranicznie 90 * x3 + 100 * x4 jest liniowe?  - %d  \n',
 is_linear(@(x1, x2, x3, x4)(90 * x3 + 100 * x4)));
fprintf('ogranicznie 40 * x3 + 50 * x4 jest liniowe? - %d \n  ',
 is_linear(@(x1, x2, x3, x4)(40 * x3 + 50 * x4)));
```

```
fprintf('ogranicznie 100 * x1 + 199.9 * x2 + 700 * x3 + 800 * x4 jest
 liniowe? - %d \n', is_linear(@(x1, x2, x3, x4)(100 * x1 + 199.9 * x2
 + 700 * x3 + 800 * x4)));
fprintf('\n\n');


cvx_begin
variables drug_1 drug_2 raw_mat_1 raw_mat_2
minimize 1000 * raw_mat_1 + 199.9 * raw_mat_2 - 5800 * drug_1 - 6300 * drug_2 %
 costs - income after transformation
subject to:
    % active ingredient of the drug constrains
    0.01 * raw_mat_1 + 0.02 * raw_mat_2 - 0.5 * drug_1 - 0.6 * drug_2
 >= 0
    % warehouse storage constrians
    raw_mat_1 + raw_mat_2 <= 1000
    % human resources constrains
    90 * drug_1 + 100 * drug_2 <= 2000
    % machines resources constrains
    40 * drug_1 + 50 * drug_2 <= 800
    % budget constrains
    100 * raw_mat_1 + 199.9 * raw_mat_2 + 700 * drug_1 + 800 * drug_2
 <= 100000

    % non negative values
    drug_1 >= 0
    drug_2 >= 0
    raw_mat_1 >= 0
    raw_mat_2 >= 0
cvx_end

fprintf('Optimal values is: material 1 = %2.10f, material 2 = %2.4f,
 drug 1 = %2.4f, drug 2 = %2.4f \n', raw_mat_1, raw_mat_2, drug_1,
 drug_2)
```

```
#### Check linearity ###########
f. celu jest liniowa? - 1
ogranicznie  x1 + x2 jest liniowe? - 1
ogranicznie 90 * x3 + 100 * x4 jest liniowe?  - 1
ogranicznie 40 * x3 + 50 * x4 jest liniowe? - 1
  ogranicznie 100 * x1 + 199.9 * x2 + 700 * x3 + 800 * x4 jest
 liniowe? - 1



Calling SDPT3 4.0: 9 variables, 4 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints =  4
 dim. of linear var  =  9
*******************************************************************
   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|3.5e+00|2.8e+00|2.7e+07| 3.085946e+06  0.000000e+00|
 0:0:00| chol  1  1
 1|0.595|0.009|1.4e+00|2.8e+00|1.2e+09| 4.702826e+06 -1.612354e+08|
 0:0:00| chol  1  1
 2|0.092|0.148|1.3e+00|2.4e+00|1.1e+09| 4.766013e+06 -1.228560e+08|
 0:0:00| chol  1  1
 3|0.126|0.199|1.1e+00|1.9e+00|9.4e+08| 4.464172e+06 -1.256388e+08|
 0:0:00| chol  1  1
 4|1.000|0.929|1.6e-08|1.4e-01|7.9e+07| 3.302774e+06 -1.587545e+07|
 0:0:00| chol  1  1
 5|1.000|0.986|3.7e-10|1.9e-03|4.3e+06| 2.820296e+06 -7.801357e+05|
 0:0:00| chol  1  1
 6|0.782|0.955|1.1e-10|8.7e-05|8.7e+05| 4.532320e+05 -4.067433e+05|
 0:0:00| chol  1  1
 7|0.247|0.412|1.5e-11|5.1e-05|7.3e+05| 3.649572e+05 -3.460892e+05|
 0:0:00| chol  1  1
 8|1.000|0.558|6.7e-11|2.3e-05|4.2e+05| 2.743518e+05 -1.212676e+05|
 0:0:00| chol  1  1
 9|1.000|0.970|2.7e-11|6.9e-07|4.0e+04| 3.912043e+04 -4.970887e+02|
 0:0:00| chol  1  1
10|0.941|1.000|1.6e-12|3.5e-09|7.2e+03| 1.716057e+04  1.000010e+04|
 0:0:00| chol  1  1
11|1.000|0.916|3.0e-14|1.9e-09|1.0e+03| 1.468470e+04  1.363977e+04|
 0:0:00| chol  1  1
12|0.981|0.982|1.7e-14|9.0e-10|2.0e+01| 1.409704e+04  1.407694e+04|
 0:0:00| chol  1  1
13|0.989|0.989|2.3e-14|4.5e-10|2.3e-01| 1.408526e+04  1.408522e+04|
 0:0:00| chol  1  1
14|0.989|0.989|4.3e-15|5.5e-12|2.8e-03| 1.408513e+04  1.408513e+04|
 0:0:00| chol  1  1
15|0.989|0.989|1.3e-15|6.7e-14|3.4e-05| 1.408513e+04  1.408513e+04|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 15
 primal objective value =  1.40851251e+04
 dual   objective value =  1.40851251e+04
 gap := trace(XZ)       = 3.36e-05
 relative gap           = 1.19e-09
 actual relative gap    = 1.60e-10
 rel. primal infeas (scaled problem)   = 1.30e-15
 rel. dual     "        "       "      = 6.73e-14
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 1.1e+04, 4.4e+02, 8.3e+02
 norm(A), norm(b), norm(C) = 1.1e+03, 8.6e+03, 1.0e+05
 Total CPU time (secs)  = 0.09
 CPU time per iteration = 0.01
 termination code       =  0
 DIMACS: 1.8e-15  0.0e+00  6.7e-14  0.0e+00  1.6e-10  1.2e-09
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): -14085.1

Optimal values is: material 1 = 0.0000000017, material 2 = 438.7889,
 drug 1 = 17.5516, drug 2 = 0.0000
```

Funkcja is_linear

```
fprintf('/n');
function y = is_linear(f)
    % Aby funkcja by#a linowa to musi spe#ania# 2 za#o#enia
    % za#1. f(ax) = a * f(x)
    % za#2. f(x+y) = f(x) + f(y)

    x1 = rand(1);
    x2 =  rand(1);
    x3 =  rand(1);
    x4 =  rand(1);
    a =  rand(1);

    % Sprawd# za#o#enie 1
    f_ax = f(a * x1, a * x2, a * x3, a * x4);
    af_x = a * f(x1, x2, x3, x4);
    % Poniewa# mamy doczynienia z obliczeniami komputerowymi, to
 musimy
    % za#o#y# za jak# dok#adno#ci# porównujemy liczby. Poni#sza
 instrukcja
    % zostanie uruchomiona gdy f_ax i af_x b#d# od siebie ró#ne.
    if ~(abs(f_ax - af_x) < 1e4*eps(min(abs(f_ax),abs(af_x))))
        fprintf('Zal1. is not fullfilled. \n')
        y = 0;
        return;
    end

    % Sprawd# za#o#enie 2
    f_x_plus_y  =  f(x1 + 2, x2 + 3, x3 + 4, x4 + 5);
    fx_plus_fy =  f(x1, x2, x3, x4) + f(2, 3, 4, 5);
     if ~(abs(f_x_plus_y - fx_plus_fy) <
 1e4*eps(min(abs(f_x_plus_y),abs(fx_plus_fy))))
        fprintf('Zal2. is not fullfilled. \n')
        y = 0;
        return;
    end

    y = 1;
end
```

```
/n
```

Published with MATLAB® R2017b