Laboratorium #9. Teoria optymalizacji.

Adam Gryczka

08.05.2017

Contents

Poniższy program służy do rozwiązywania problemów geometrycznych.

Dokument zawiera rozwiązanie 11 zadań na dwa sposoby

```
cvx_gp = zeros(11,1);
cvx_log = zeros(11,1);
WynikiZgodne = strings(11,1);
```

Zadanie 1

```
total_len = 2400;
cvx_begin gp quiet
    variables x y
    maximize x*y
    subject to
        2*x + y <= total_len;
cvx_end

cvx_begin quiet
    variables logx logy
    maximize(logx + logy)
    subject to
        log(2*exp(logx) + exp(logy)) <= log(total_len);
cvx_end
proba = 1;
cvx_gp(proba,1) = x;
cvx_gp(proba,2) = y;

cvx_log(proba,1) = exp(logx);
cvx_log(proba,2) = exp(logy);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 2

```
total_len = 500;
cvx_begin gp quiet
    variables x y
    maximize x*y
    subject to
        x + 2 * y <= total_len;
cvx_end

cvx_begin quiet
    variables logx logy
    maximize(logx + logy)
    subject to
        log(2*exp(logy) + exp(logx)) <= log(total_len);
cvx_end
proba = 2;
cvx_gp(proba,1) = x;
cvx_gp(proba,2) = y;
cvx_log(proba,1) = exp(logx);
cvx_log(proba,2) = exp(logy);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 3

```
total_vol = 50;
cvx_begin gp quiet
    variables h w
    minimize( 60*w*w + 48 * w * h )
    subject to
        3*w*w*h == total_vol;
cvx_end

% 2 część
total_vol = 50;
cvx_begin quiet
    variables logh logw
    minimize( 60*exp(logw)*exp(logw) + 48*exp(logw)*exp(logh) )
    subject to
         log(3)+2*logw + logh == log(total_vol) ;
cvx_end

proba = 3;
cvx_gp(proba,1) = w;
cvx_gp(proba,2) = h;

cvx_log(proba,1) = exp(logw);
cvx_log(proba,2) = exp(logh);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 4

```
% 1 wersja
total_material = 10;

cvx_begin gp quiet
    variables w h
    maximize w^2*h
    subject to
        2*w^2+4*w*h <= total_material;
cvx_end
% 2 wersja
cvx_begin quiet
    variables logw logh
    maximize log(exp(logw)^2*exp(logh))
    subject to
        log(2*exp(logw)^2+4*exp(logw)*exp(logh)) <= log(total_material);
cvx_end

proba = 4;
cvx_gp(proba,1) = w;
cvx_gp(proba,2) = h;

cvx_log(proba,1) = exp(logw);
cvx_log(proba,2) = exp(logh);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 5

1 wersja

```
total_amount = 1500;

cvx_begin gp quiet
    variables r h
    minimize 2*pi()*r^2 + 2*pi()*r*h
    subject to
        pi()*r^2*h == total_amount;
cvx_end

cvx_begin quiet
    variables logr logh
    minimize (2*pi()*exp(logr)^2+2*pi()*exp(logr)*exp(logh))
    subject to
        log(pi()*exp(logr)^2*exp(logh)) == log(total_amount)
cvx_end

proba = 5;
cvx_gp(proba,1) = r;
cvx_gp(proba,2) = h;

cvx_log(proba,1) = exp(logr);
cvx_log(proba,2) = exp(logh);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 6 - NIE DZIAŁA!!

cvx_begin gp quiet variables h % maximize(140*h-48*h^2+4*h^3) maximize(140*h^0-96*h+12*h^2) subject to % h >= 0 % h <= 5 % 140-96*h+12*h^2 <= 0 24*h-96 == 0 cvx_end h

Zadanie 7

```
surface = 200;
cvx_begin gp quiet
    variables w h
    maximize( 207-3.5*w-400/w )
    subject to
        w*h == surface
        w >= 2;
        w <= 200/3.5
cvx_end

% 2 sposób
surface = 200;
cvx_begin quiet
    variables logw logh
    maximize( 207-3.5*exp(logw)-400/exp(logw) )
    subject to
        logw+logh == log(surface)
        logw >= log(2);
        logw <= log(200/3.5)
cvx_end

proba = 7;
cvx_gp(proba,1) = w;
cvx_gp(proba,2) = h;

cvx_log(proba,1) = exp(logw);
cvx_log(proba,2) = exp(logh);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
```

Zadanie 8 - NIE DZIAŁA!!

cvx_begin gp quiet variables r h maximize(12-(4+pi)*r) %12*r - (2 + 0.5 * pi)*r^2) subject to % 2*h+2*r+pi*r 12 r=0 r<=12/(2+pi) cvx_end r h

Zadanie 9 - NIE DZIAŁA!!

```
cvx_begin quiet
    variables u v
    maximize( 256*v-16*v*v )
    subject to
        u + v == 16
cvx_end

cvx_begin quiet
    variables logu logv
%     maximize (256*exp(logv)-16*exp(logv)*exp(logv))
    minimize (16*(exp(logu) + exp(logv)))
    subject to
        log(exp(logu) + exp(logv)) <= log(16)

cvx_end
proba = 9;
cvx_gp(proba,1) = u;
cvx_gp(proba,2) = v;

cvx_log(proba,1) = exp(u);
cvx_log(proba,2) = exp(v);

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) && (round(cvx_gp(proba,2)) == round(cvx_log(proba,2)))
    WynikiZgodne(proba) = string('Tak');
else
    WynikiZgodne(proba) = string('Nie');
end
% cvx_begin gp
%     variables x y
%     maximize x*y
%     subject to
%         2*x + y <= total_len;
% cvx_end
%
% cvx_begin
%     variables logx logy
%     maximize(logx + logy)
%     subject to
%         log(2*exp(logx) + exp(logy)) <= log(total_len);
% cvx_end
```

Zadanie 10 - NIE DZIAŁA!!

cvx_begin gp quiet variables x y minimize sqrt(y^2 - 3*y + 3) % minimize(x^2+(y-2)^2) subject to 2*y - 3 == 0; % y >= x^2+1 cvx_end x y

Zadanie 11 - NIE DZIAŁA!!

cvx_begin gp quiet variables x minimize(sqrt(x^2 + 9)) subject to x/(6*sqrt(x^2+9))-1/8 == 0 cvx_end

proba = 4; cvx_gp(proba,1) = x; cvx_gp(proba,2) = NaN;

cvx_log(proba,1) = NaN; cvx_log(proba,2) = NaN;

if (round(cvx_gp(proba,1)) == round(cvx_log(proba,1))) WynikiZgodne(proba) = string('Tak'); else WynikiZgodne(proba) = string('Nie'); end

```
NrZadania = {'Zadanie 1';'Zadanie 2';'Zadanie 3';'Zadanie 4';'Zadanie 5';'Zadanie 6';'Zadanie 7';'Zadanie 8';'Zadanie 9';'Zadanie 10';'Zadanie 11'};

T = table(cvx_gp,cvx_log,WynikiZgodne,'RowNames',NrZadania)
```

```
T =
```

	cvx_gp		cvx_log		WynikiZgodne
Zadanie 1	600	1200	600	1200	"Tak"
Zadanie 2	250	125	250	125	"Tak"
Zadanie 3	1.8821	4.7051	1.8821	4.705	"Tak"
Zadanie 4	1.291	1.2909	1.291	1.2909	"Tak"
Zadanie 5	6.2036	12.407	6.2036	12.407	"Tak"
Zadanie 6	0	0	0	0	""
Zadanie 7	10.69	18.708	10.69	18.708	"Tak"
Zadanie 8	0	0	0	0	""
Zadanie 9	8	8	2981.1	2980.8	"Nie"
Zadanie 10	0	0	0	0	""
Zadanie 11	0	0	0	0	""

Published with MATLAB® R2016b