

Laboratorium #11. Teoria optymalizacji.

Adam Gryczka

15.05.2017

```
% data for power flow problem
n = 12; % total number of nodes
m = 18; % number of edges (transmission lines)
k = 4; % number of generators
rand('state',0);
Pmax = 1+4*rand(m,1); % transmission line capacities
% Pmax(18,1)=0;
Gmax = [3; 2; 4; 7]; % maximum generator power
c = [4; 8; 5; 3]; % supply generator costs
d = 1+1.5*rand(n-k,1); % network power demands
% graph incidence matrix
A = [ -1 -1 0 0 0 0 0 0 0 0 0 0 ;
      0 0 -1 -1 0 0 0 0 0 0 0 0 ;
      0 0 0 0 0 0 0 0 0 -1 -1 0 ;
      0 0 0 0 0 0 -1 0 0 0 0 0 ;
      1 0 0 0 1 -1 0 0 0 0 0 0 ;
      0 1 1 0 -1 0 1 -1 0 0 0 0 ;
      0 0 0 1 0 0 0 0 -1 1 0 0 ;
      0 0 0 0 0 0 0 1 1 0 0 -1 ;
      0 0 0 0 0 0 0 0 0 1 -1 0 ;
      0 0 0 0 0 0 0 0 0 0 1 1 ;
      0 0 0 0 0 0 0 0 0 0 0 1 ;
      0 0 0 0 1 0 0 0 0 0 0 -1 ];

% the code below is not data for the problem
% it is used only to generate the network graph

% x-y coordinates
XY = [ ... % node x-y coordinates
      1.5 5.2; % node 1
      4.9 5;   % node 2
      6.9 3.5; % node 3
      1.9 3.5; % node 4
      0.2 4.4; % node 5
      3.2 4.8; % node 6
      5.9 4.5; % node 7
      3.9 3.6; % node 8
      5.9 2.5; % node 9
      3.9 3;   % node 10
      1.4 2.5; % node 11
      0 3];    % node 12

% node adjacency matrix
Ad = -A*A';
Ad = Ad - diag(diag(Ad));

epsx = 0.05; epsy = 0.15; % text placing offset

figure;
% connect edges
gplot(Ad,XY,'-k'); hold on;

% label generator nodes
for j = 1:k
    plot(XY(j,1),XY(j,2),'rs',...
         'MarkerFaceColor','r',...
         'MarkerSize',12);
```

```

    text(XY(j,1)-eps,XY(j,2)+epsy,int2str(j),'FontSize',10);
end

% label regular nodes
for j = k+1:n
    plot(XY(j,1),XY(j,2),'.k',...
        'MarkerSize',15);
    text(XY(j,1)-eps,XY(j,2)+epsy,int2str(j),'FontSize',10);
end
axis off; hold off;
% print('-depsc','pwr_net.eps');

% Podpunkt 1

cvx_begin quiet
    variable g(4,1) % całkowita moc WSZYSTKICH generatorów
    minimize (c'*g) % koszt generowania mocy
    subject to
        g>=0 % każdy generator generuje prąd (dodatnio) a nie go pobiera (ujemnie)
        abs(Gmax)>= g % Każdy generator produkuje moc mniejszą od swojej własnej mocy maksymalnej
        sum(g)==sum(d) % Suma mocy generowanych jest równa sumie mocy zużytych
cvx_end;
disp('Podpunkt 1:')
c'*g

% Podpunkt 2

% Tu wynik wychodzi dobry, ale chyba zła droga dojścia
cvx_begin quiet
    variables p(m, 1) g(k, 1)
    minimize(c'*g)
    subject to
        sum(g) == sum(d)
        p(3) == 0 % Odłączam trzecią linię
        abs(Pmax) <= p
        Gmax <= g
cvx_end
disp('Podpunkt 2:')
c'*g

```

Podpunkt 1:

ans =

44.5952

Podpunkt 2:

ans =

56.1865

