Matlab Coding of MFA optimization for Graph bisection

- Matlab module
  - Graph bisection data generation
  - MFA
  - v_updating
  - v2cutsize

Graph bisection data generation

function T=graph_bisection_data(density,N,A)

N: node size

Density is a parameter that is employed to control connectivity of nodes

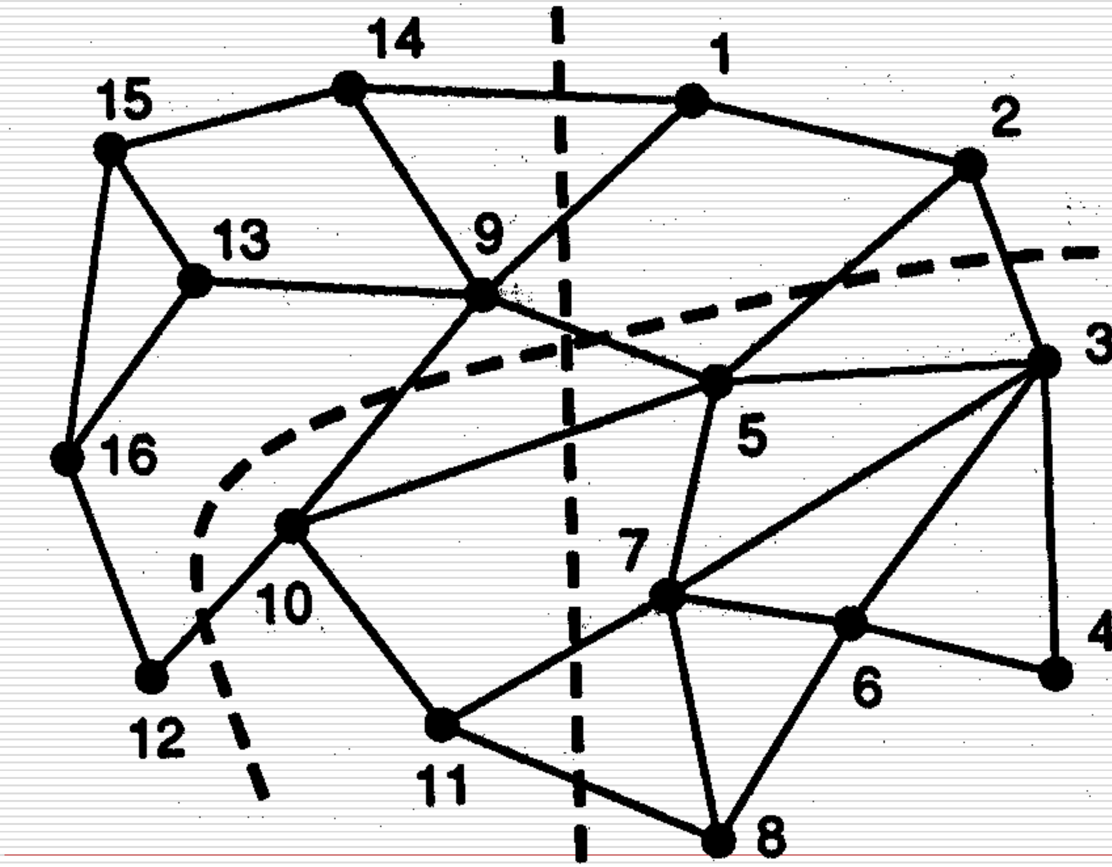
- s=rand(1,1)
- two nodes are connected if s > density

T is a matrix that collect edges

- T(i,i) = A for all i
- T(i,j) equals T(j,i)
- T(i,j) equals one if nodes i and j are connected and equals zero otherwise

Graph Bipartition

Cost 6



Cost 5

Energy function

$$\mathrm{E}(S) = -\sum_{i=1}^{N}\sum_{j\neq i}^{N} \mathrm{T}_{ij} S_i S_j + \frac{\mathrm{A}}{2}(\sum_{i} S_i)^2$$

$$\mathrm{E}(S) = -\sum_{i=1}^{N}\sum_{j\neq i}^{N} w_{ij} S_i S_j$$

$$w_{ij} = \mathrm{T}_{ij} - A, \text{ for i} \neq \text{j}$$

$$w_{ii} = 0$$

Call MFA

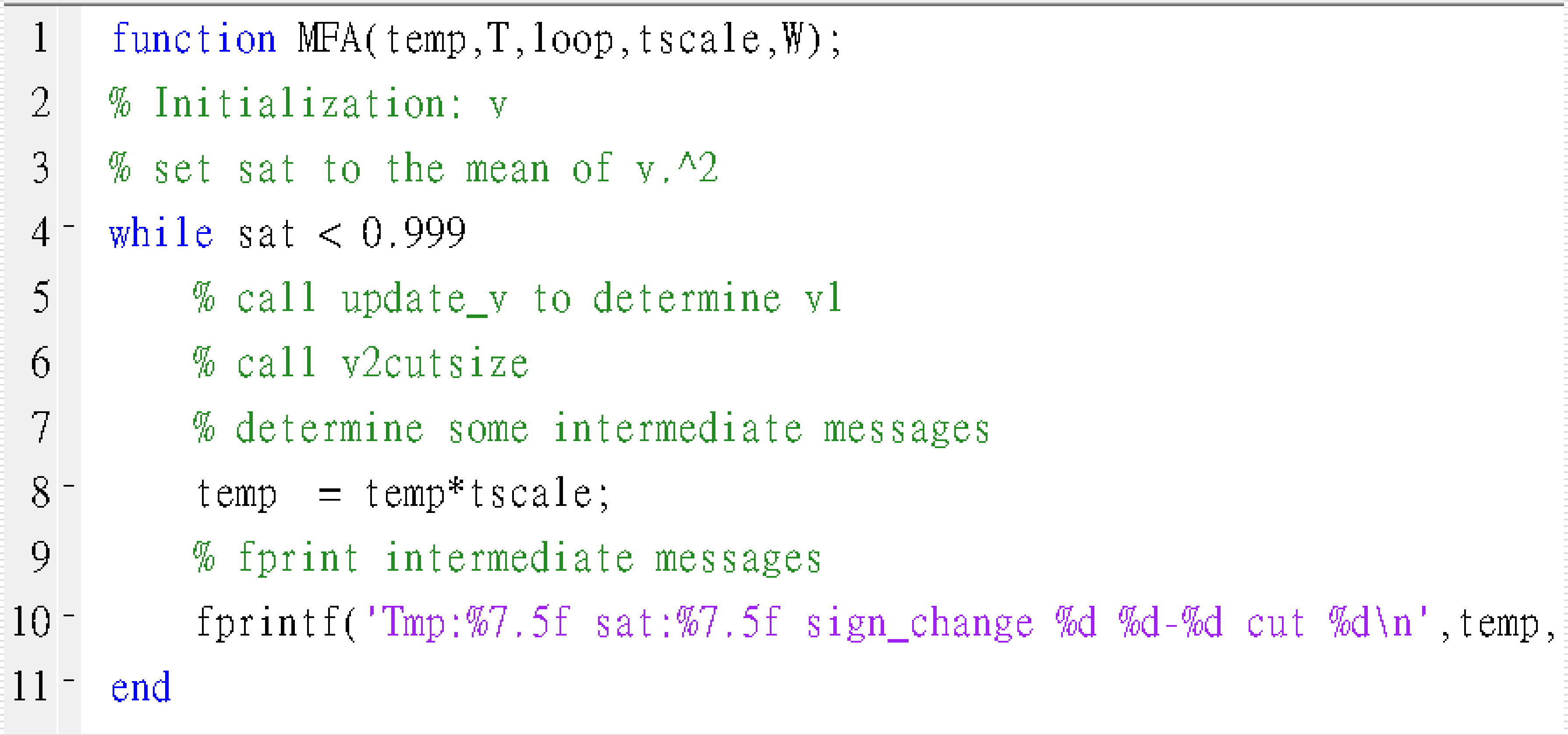
```
W=T-A;
MFA(15,T,loop,tscale,W);
```

MFA

function MFA(temp,T,loop,tscale,W);

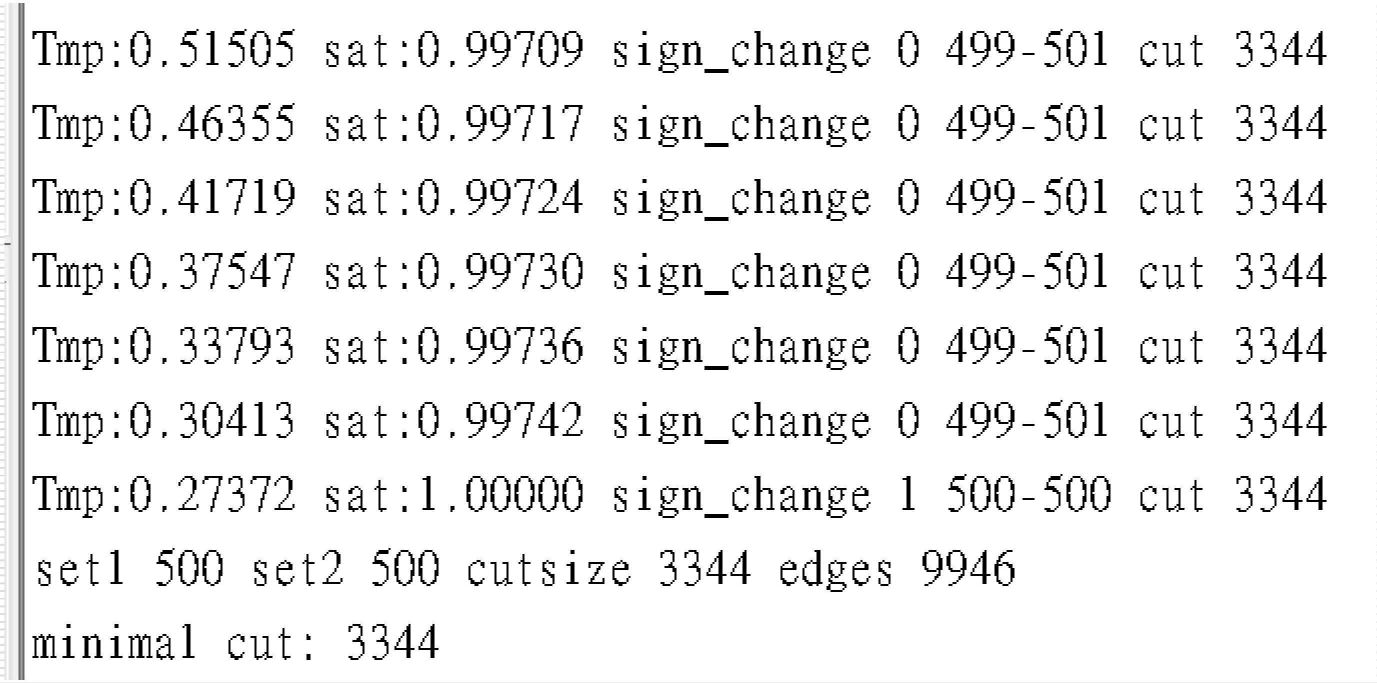
- □ temp : temperature
- □ T: connection matrix
- □ loop: 50
- □ tscale: annealing factor
- □ W: weight matrix in the energy function

MFA



```
function MFA(temp,T,loop,tscale,W);
% Initialization: v
% set sat to the mean of v.^2
while sat < 0.999
    % call update_v to determine v1
    % call v2cutsize
    % determine some intermediate messages
    temp  = temp*tscale;
    % fprint intermediate messages
    fprintf('Tmp:%7.5f sat:%7.5f sign_change %d %d-%d cut %d\n',temp,
end
```

Intermediate messages



```
Tmp:0.51505 sat:0.99709 sign_change 0 499-501 cut 3344
Tmp:0.46355 sat:0.99717 sign_change 0 499-501 cut 3344
Tmp:0.41719 sat:0.99724 sign_change 0 499-501 cut 3344
Tmp:0.37547 sat:0.99730 sign_change 0 499-501 cut 3344
Tmp:0.33793 sat:0.99736 sign_change 0 499-501 cut 3344
Tmp:0.30413 sat:0.99742 sign_change 0 499-501 cut 3344
Tmp:0.27372 sat:1.00000 sign_change 1 500-500 cut 3344
set1 500 set2 500 cutsize 3344 edges 9946
minimal cut: 3344
```

messages

- tmp: temperature
- sat : expectation of v.^2
- sign-change : number of elements of v whose signs change after calling update_v
- Numbers of nodes in two sets
- cutsize

Annealing schedule

- □ Set beta to a sufficiently small value
- □ Increase beta carefully

v2cutsize

function [count1,count2,cutsize]=oc(v,T)

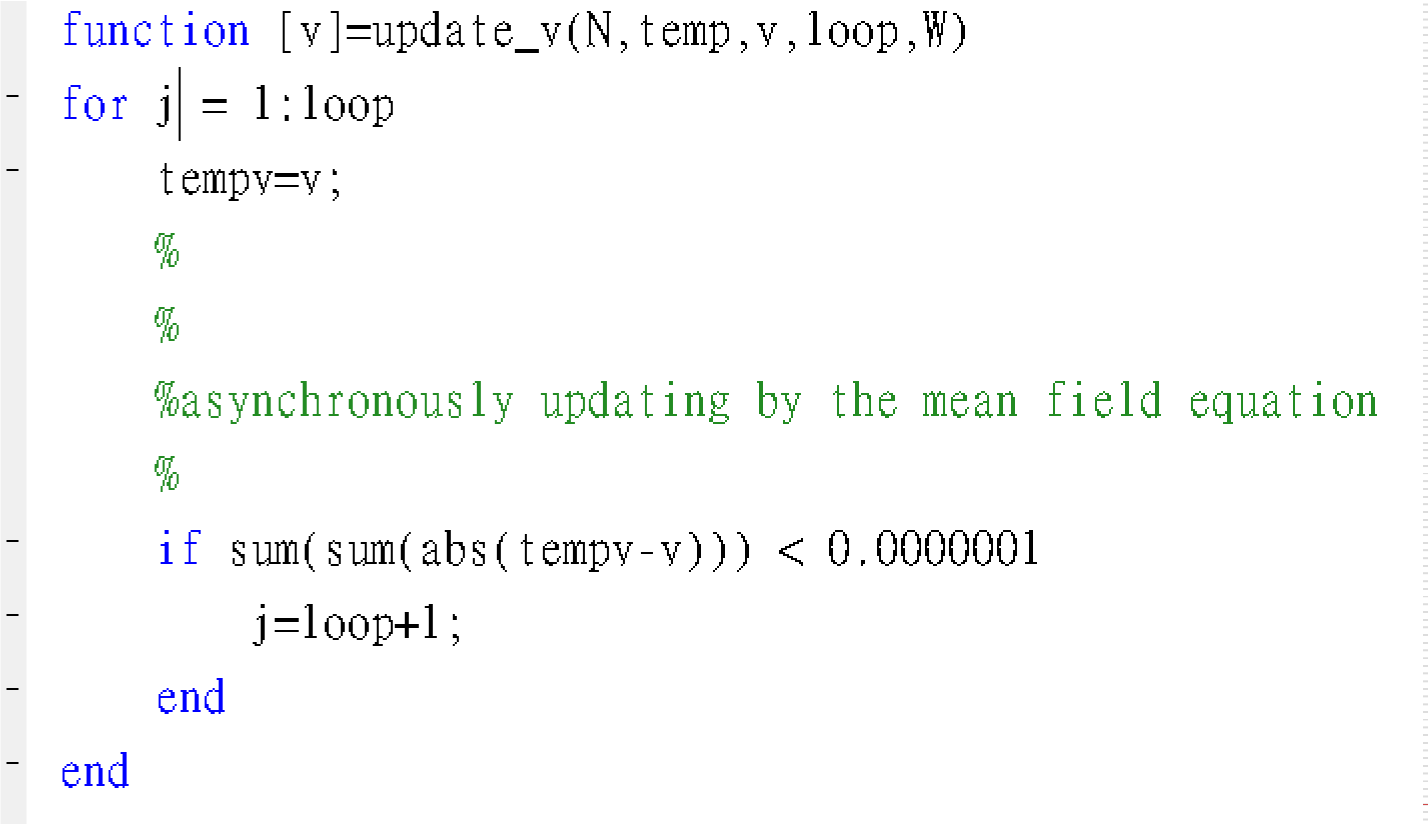
- v: state vector
- T: connection matrix
- cutsize
- count1 : number of nodes in one set
- count2 : number of nodes in the other set

Mean field equation

$$v_i = \tanh(\beta \sum_{j \neq i}^{N} w_{ij} v_j)$$

update_v

function [v]=update_v(N,temp,v,loop,W)



```
function [v]=update_v(N,temp,v,loop,W)
for j = 1:loop
    tempv=v;
    %
    %
    %asynchronously updating by the mean field equation
    %
    if sum(sum(abs(tempv-v))) < 0.0000001
        j=loop+1;
    end
end
```

Exercise, due to 12/26

- □ Implement MFA optimization for solving the graph bisection problem
- □ Give two examples to verify your matlab codes