TSP and Sudoku

Hopfield Neural Neworks

- TSP

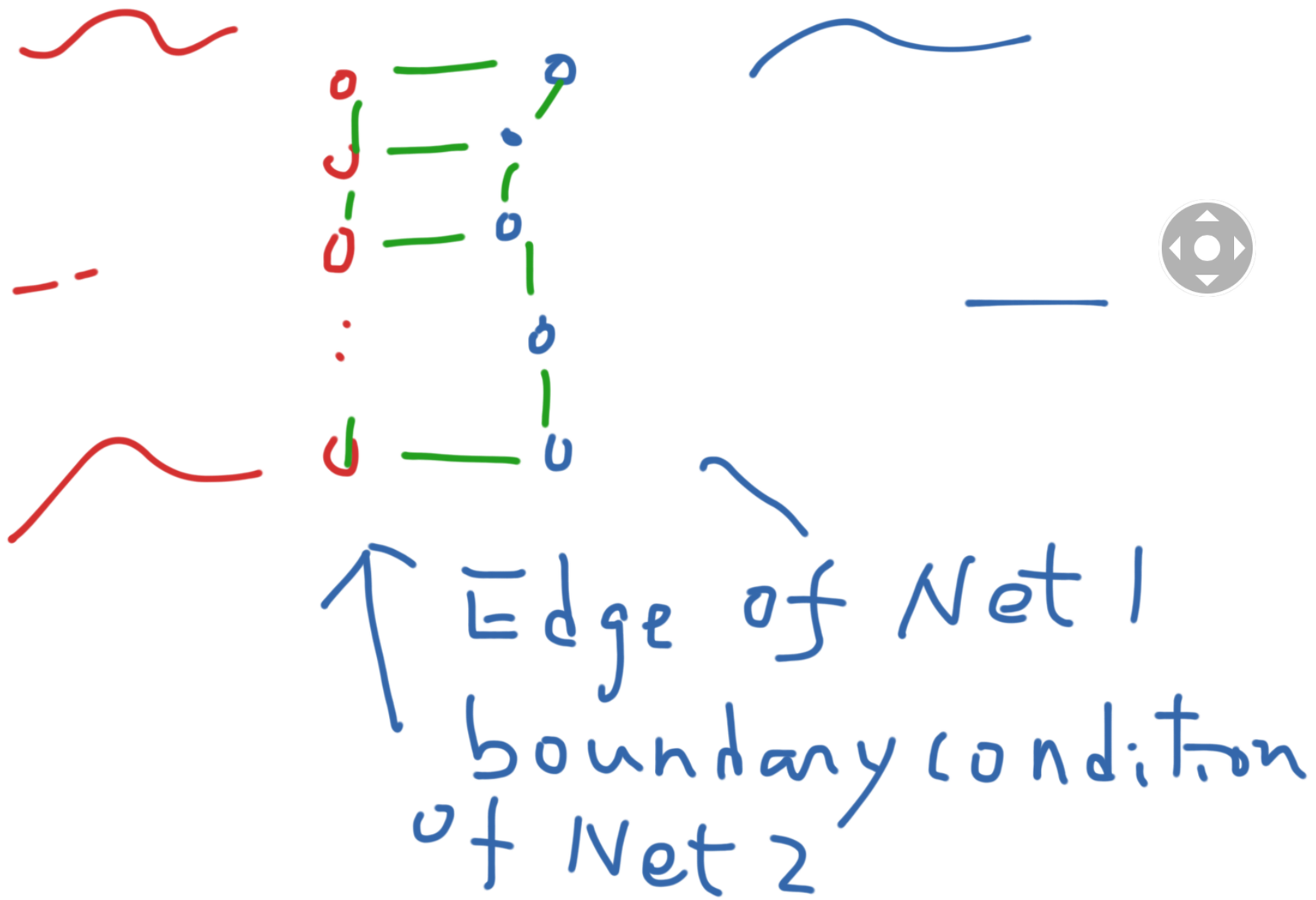
Current SoM or TSP

- partition by annealed k-means

- Concurrent sub-tasks

- Concurrent SoMs

Ex 2 - concurrent SOMs



TSP and Potts model

Kirkpatick Simulated annealing

Durbin & Willshaw Elastic Nets

Hopfield & TANK SOM, Mixed Integer Programming

. Set β to a sufficiently low value, $v_{ia} \approx \frac{1}{N}$, for all i, a

HC

EXIT

. Use mean field equation to update all v_{ia}

Increase β by an annealing process

[15] R.A. Nobakht, S.H. Ardalan, D.E. Van de
memory by quantized mean field annealing,
[16] M. Padberg, G. Rinaldi, Optimization of a 53
and cut, Oper. Res. Lett. 6 (1) (1987) 1–7.
[17] C. Peterson, B. Söderberg, A new method for
Int. J. Neural Syst. 1 (1989) 3.
[18] salesman problem, Neural Network 1 (1988)
[19] [4] D.J. Burr, Elastic matching of line drawings
6 (1980) 708–713.
[20] [5] D.W. Dong, J.J. Hopfield, Dynamic properti
3 (1992) 267–283.
[6] R. Durbin, G. Willshaw, An analogue appro
net method, Nature 326 (1987).
[7] G.J. Goodhill, D.J. Willshaw, Application c
dominance stripes, Network 1 (1990) 41–59.

Potts Mean Field Annealing for TSP

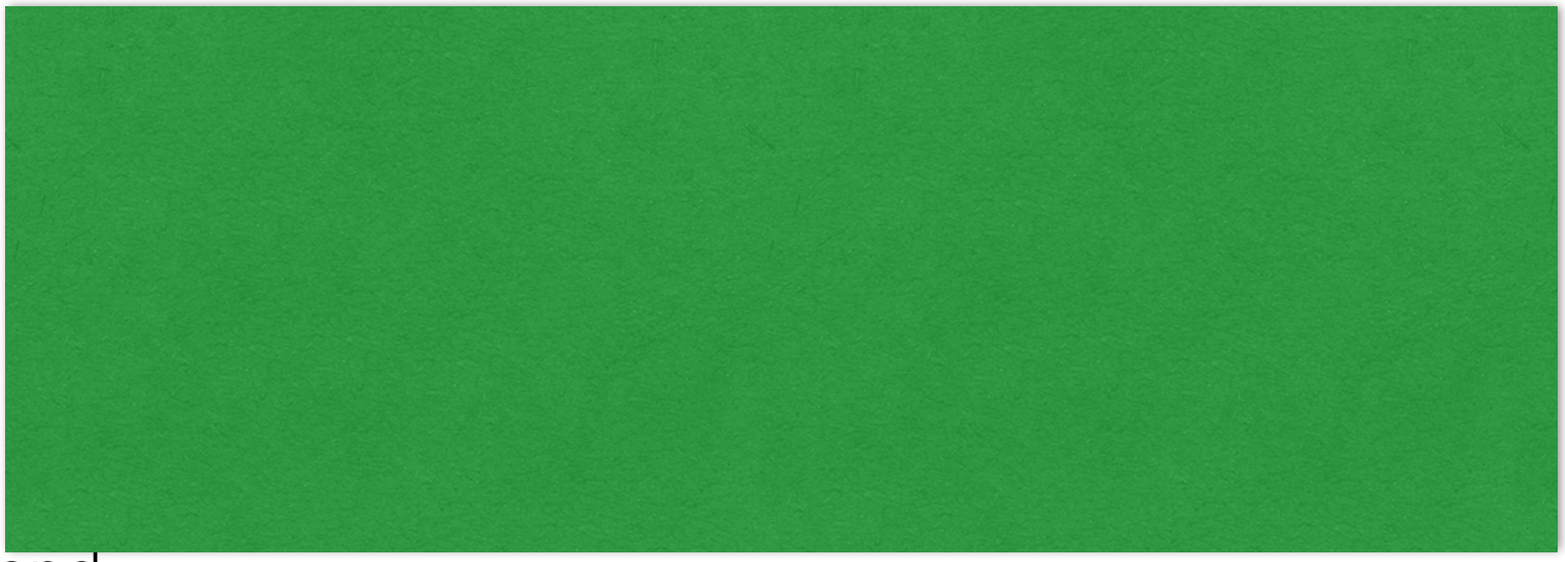
- MATLAB Coding

```
function demoMFA_TSP()
    mode=1;
    switch mode
        case 1
            T0=1;
            tscale=0.99;
        case 2
            tscale=0.9995;
            T0=0.5;
    end
    A=1;    N=20;
    [D x]=TSP_data(N);
    loop=50;
    seq=MFA_TSP(T0,D,loop,tscale,A);
    plot_tour(x,seq);

return
function plot_tour(x,seq)
    N=length(seq);figure;hold on;
    for i=1:N-1
        s=seq(i);t=seq(i+1);
        plot(x(1,[s t]),x(2,[s t]));
    end
    s=seq(N);t=seq(1);
    plot(x(1,[s t]),x(2,[s t]));
    plot(x(1,:),x(2,:),'.');
return
function [D x]=TSP_data(N)
    x=rand(2,N);
    for i=1:N
        for j=1:N
            D(i,j)=norm(x(:,i)-x(:,j));
        end
    end
```

```
function seq=MFA_TSP(temp,D,loop,tscale,A);
N=size(D,1);
K=N;
v = 1/K+(rand(N,K)-0.5)/1000;
sat = sum(sum(v.*v))/N;
vmini=N*N;ii=0;
while sat < 0.99
    % E step
        [v] = update_v_tsp(N,temp,A,v,loop,D);
    % M step
    sat = sum(sum(v.*v))/N;
    temp  = temp*tscale;
    ii=ii+1;
    if mod(ii,10)==0
    fprintf('Tmp:%7.5f sat:%7.5f \n',temp,sat);
    end
end
[vd,tour_length]= v2tour_length(v,D);
fprintf('validity %d tour_length: %f\n',vd,tour_length);
[vv seq]=max(v);
```

```
function [v]=update_v_tsp(N,temp,A,v,loop,D)
v = v+(rand(N,N)-0.5)*10.^-8;
indp=[N 1:N-1];indn=[2:N 1];
for j = 1:loop
    tempv=v;
      for i = 1:N
```



```
    end
    if sum(sum(abs(tempv-v))) < 10.^-8
        j=loop+1;
    end
end
```

```
function [vd,tour_length]= v2tour_length(v,D)
    N=size(D,1);
    [vv v_stop]=max(v');
    vd=0;tour_length=0;
    for i=1:length(v_stop)
        ip1=i+1;
        if ip1 > N
            ip1=1;
        end
    if v_stop(i)==v_stop(ip1)
        vd=vd+1;
    else
        tour_length=tour_length+D(v_stop(i),v_stop(ip1));
    end
end
```