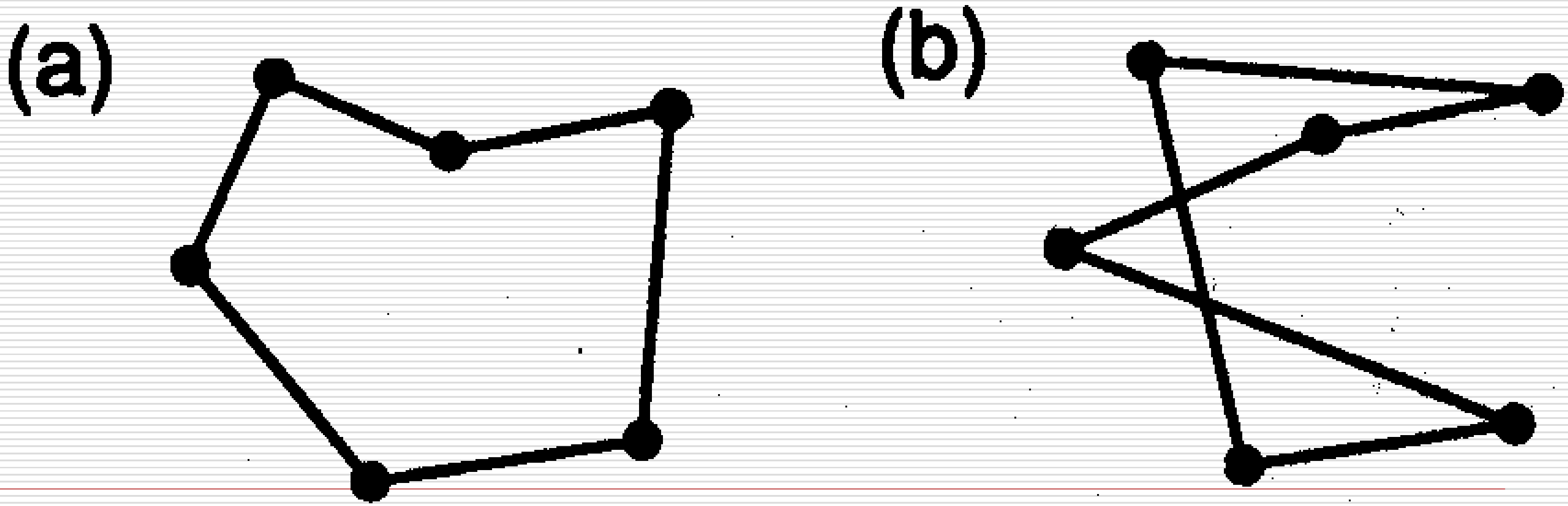
TSP Problem

Given N cities or points in a space with distances $\{d_{ij}\}$ btw cities. The task is to find the minimum-length closed tour that visits each city once and returns to its starting city.



A problem in class P or class NP

If there exists an algorithm that solves the problem in a time that grows only polynomially(or slower) with the size N of the problem, the problem is in class P

NP problems are those for which one can test in polynomial time whether any ``guess'' of the solution is right.

NP complete

if one could find a deterministic algorithm that solves one NP-complete problem in polynomial time, then all other NP problems could be solved in polynomial time.

The TSP problem is an NP-complete problem

Hopfield neural network for TSP
Biological Cybernetics, 1985

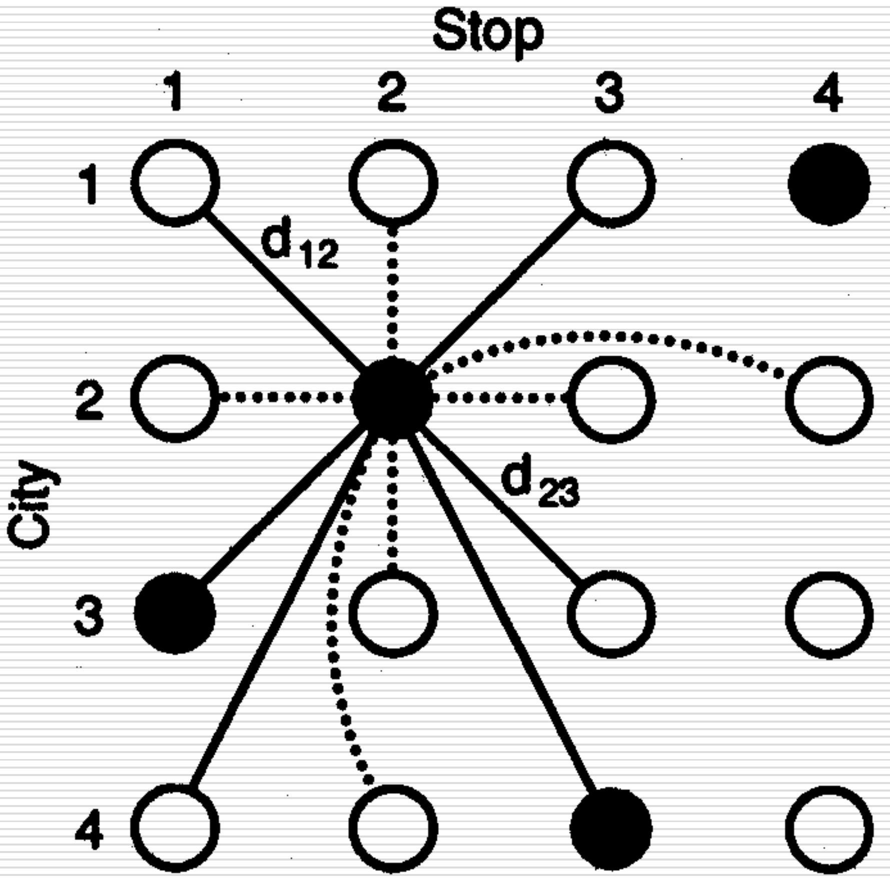


FIGURE 4.5 Network to solve a four-city travelling salesman problem. Solid and open circles denote units that are on and off respectively when the network is representing the tour 3–2–4–1. The connections are shown only for unit n_{22}; solid lines are inhibitory connections of strength $-d_{ij}$, and dotted lines are uniform inhibitory connections of strength $-\gamma$. All connections are symmetric. Thresholds are not shown.

Internal representations

$n_{ia} = 1$ if and only if city i is the ath stop on the tour

The total length of the tour is

$$L = \frac{1}{2} \sum_{ij,a} d_{ij} n_{ia} (n_{j,a+1} + n_{j,a-1})$$

Constraints

$$\sum_a n_{ia} = 1 \qquad \text{(for every city } i\text{)}$$

$$\sum_i n_{ia} = 1 \qquad \text{(for every stop } a\text{)}$$

Energy function

$$H = \frac{1}{2}\sum_{ij,a} d_{ij} n_{ia}(n_{j,a+1} + n_{j,a-1})$$
$$+ \frac{\gamma}{2}\left[\sum_a \left(1 - \sum_i n_{ia}\right)^2 + \sum_i \left(1 - \sum_a n_{ia}\right)^2\right].$$

Mean field equations

$$u_{ia} = -\frac{\partial H}{\partial n_{ia}}$$

$$n_{ia} = g(\beta u_{ia}) = \frac{1}{2}(1 + \tanh(\beta u_{ia}))$$

Hopfield's Result

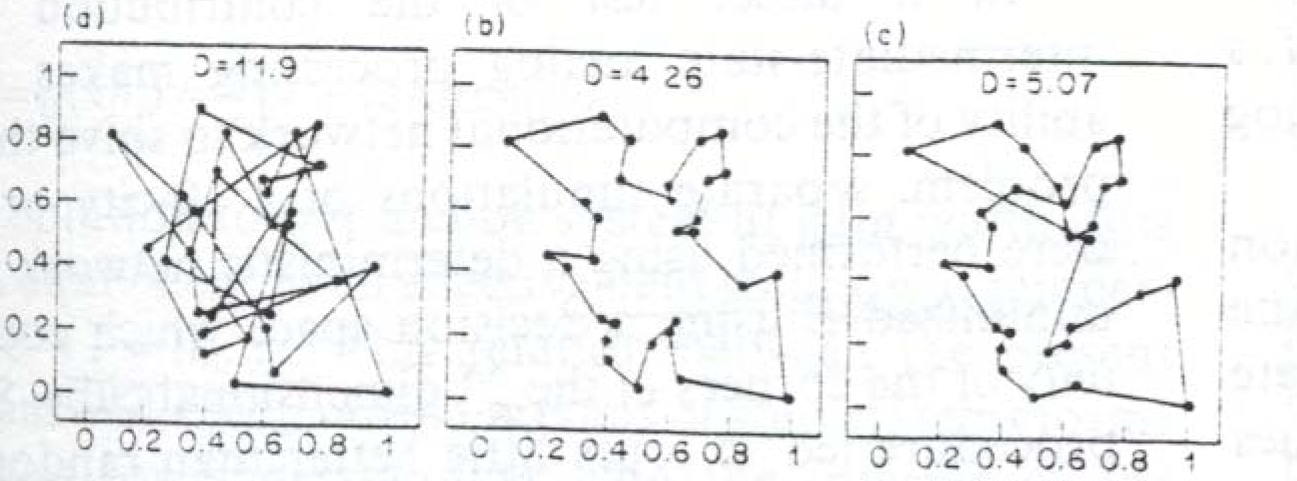


Fig. 4a–c. **a** A random tour for 30 random cities. **b** The Lin-Kernighan tour. **c** A typical tour obtained from the analog network by slowing increasing the gain

Advantages and disadvantages of Hopfield neural networks

Qualified collective decision?
Parallel and distributed architecture
Fault tolerance

vs

Validity: selection of parameters, internal representations
Quality: geometrical feature, combinatorial feature

Potts neural networks for TSP

- Internal representations
- Mathematical framework
- Mean field annealing

Carsten Peterson - Homepage

Rewrite this energy function

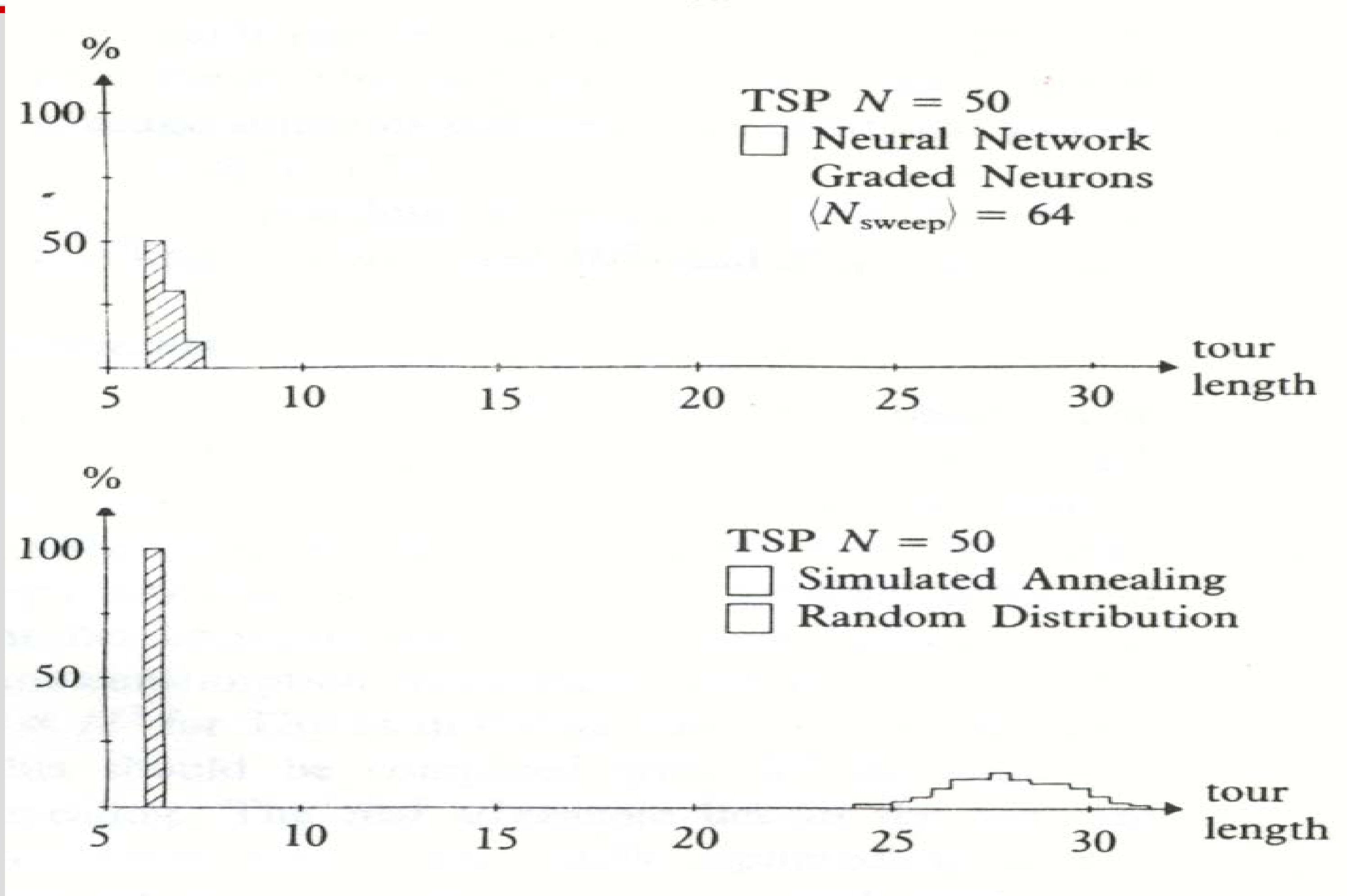
$$E = \sum_{ij} D_{ij} \sum_{a} S_{ia} S_{j(a+1)} - \frac{\beta}{2} \sum_{i} \sum_{a} S_{ia}^2 + \frac{\alpha}{2} \sum_{a} \left(\sum_{i} S_{ia}\right)^2.$$

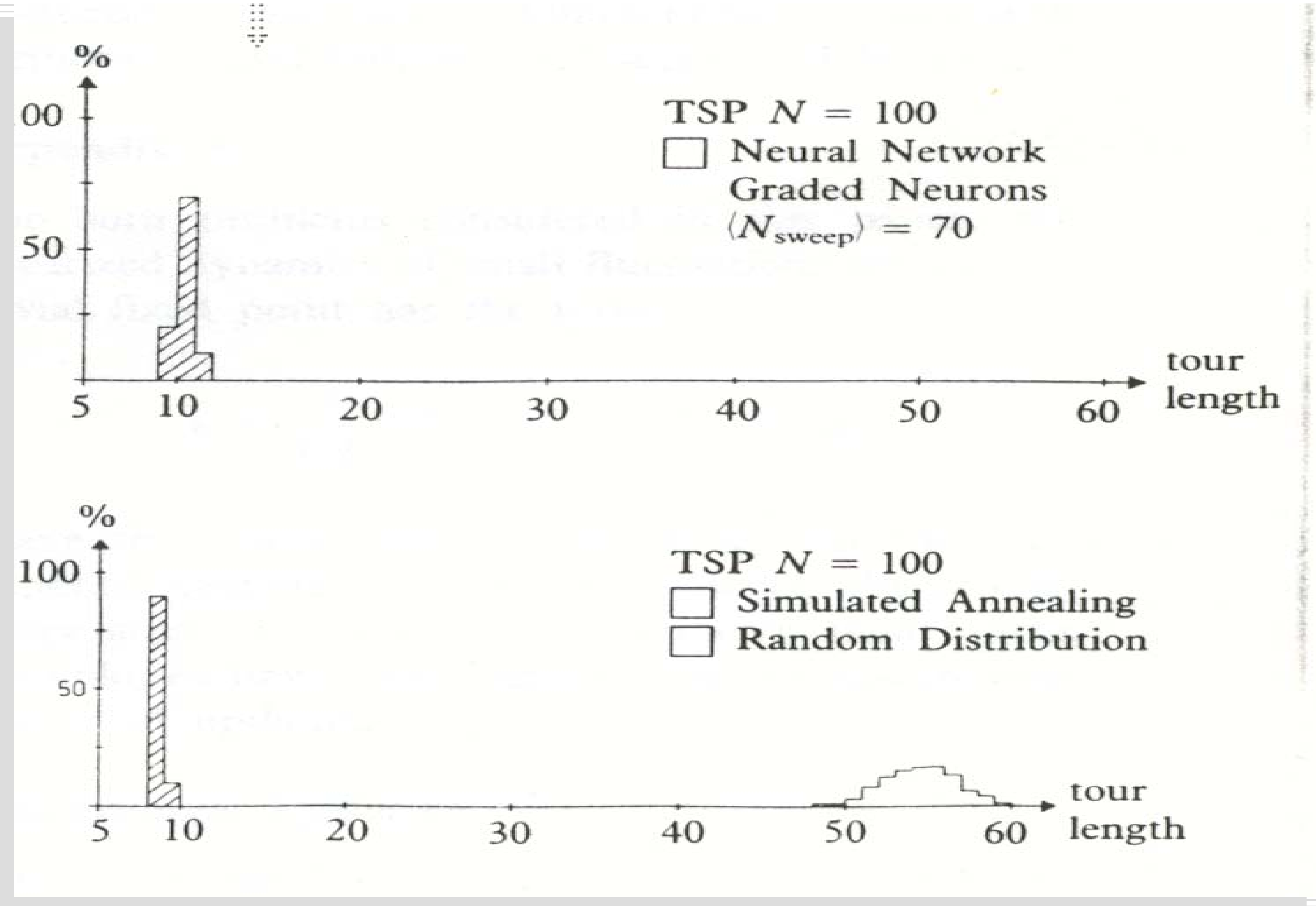
S_{ia} be on if the city labelled i is visited at the ath stop in the tour.

Energy function

$$E = \sum_{ij} D_{ij} \sum_{a} S_{ia} S_{j(a+1)} + \frac{\beta}{2} \sum_{i} \sum_{a \neq b} S_{ia} S_{jb}$$

$$+ \frac{\alpha}{2} \sum_{a} \Big(\sum_{i} S_{ia} - 1\Big)^2$$


%
100
50
TSP $N = 50$
Neural Network
Graded Neurons
$\langle N_{\text{sweep}} \rangle = 64$
5
10
15
20
25
30
tour
length
%
100
50
TSP $N = 50$
Simulated Annealing
Random Distribution
5
10
15
20
25
30
tour
length


%
00
50
TSP N = 100
Neural Network
Graded Neurons
$\langle N_{sweep} \rangle = 70$
5
10
20
30
40
50
60
tour
length
%
100
50
TSP N = 100
Simulated Annealing
Random Distribution
5
10
20
30
40
50
60
tour
length

Rewrite the energy function

$$\mathbf{S}_i = (\mathrm{S}_{\mathrm{i1}}\ \mathrm{S}_{\mathrm{i2}}\ ,\ldots,\ \mathrm{S}_{\mathrm{iN}})^{\mathrm{T}} \in \{\mathbf{e}_1,\ldots,\mathbf{e}_N\}$$

$$\mathrm{E} = \sum_{ij} D_{ij} \sum_{a} S_{ia} S_{j(a+1)} + \frac{\alpha}{2} \sum_{a} (\sum_{i} S_{ia} - 1)^2$$

$$\sum_a \left(\sum_i S_{ia} - 1\right)^2$$

$$= \sum_a \left(\left(\sum_i S_{ia}\right)^2 - 2\sum_i S_{ia} + 1\right)$$

$$= \sum_a \left(\sum_i S_{ia}\right)^2 - 2\sum_a \sum_i S_{ia}$$

$$= \sum_a \sum_i S_{ia}^2 + 2\sum_a \sum_i \sum_{j \neq i} S_{ia} S_{ja} - 2\sum_a \sum_i S_{ia}$$

$$= 2\sum_a \sum_i \sum_{j \neq i} S_{ia} S_{ja} - \sum_a \sum_i S_{ia} = 2\sum_a \sum_i \sum_{j \neq i} S_{ia} S_{ja}$$

Energy function

$$\mathbf{S}_i = (\mathrm{S}_{\mathrm{i1}}\ \mathrm{S}_{\mathrm{i2}}\ ,\ldots,\mathrm{S}_{\mathrm{iN}})^{\mathrm{T}} \in \{\mathbf{e}_1,\ldots,\mathbf{e}_N\}$$

$$\mathrm{E} = \sum_i \sum_{j \neq i} D_{ij} \sum_a S_{ia} S_{j(a+1)} + \alpha \sum_a \sum_i \sum_{j \neq i} S_{ia} S_{ja}$$

Mean energy

$$\mathbf{v}_i = (\mathrm{v}_{\mathrm{i1}}\ \mathrm{v}_{\mathrm{i2}}\ ,\ldots,\ \mathrm{v}_{\mathrm{iN}})^{\mathrm{T}}$$

$$\mathrm{E} = \sum_i \sum_{j \neq i} D_{ij} \sum_a v_{ia} v_{j(a+1)}$$

$$+ \alpha \sum_a \sum_i \sum_{j \neq i} v_{ia} v_{ja}$$

Mean field equation

$$\mathrm{E} = \sum_{i} \sum_{j \neq i} D_{ij} \sum_{a} v_{ia} v_{j(a+1)}$$

$$+ \alpha \sum_{a} \sum_{i} \sum_{j \neq i} v_{ia} v_{ja}$$

$$u_{ia} = -\frac{dE}{dv_{ia}} = -\sum_{j \neq i} D_{ij} \left(v_{j(a-1)} + v_{j(a+1)} \right) - \alpha \sum_{j \neq i} v_{ja}$$

Mean field equation

$$u_{ia} = -\frac{dE}{dv_{ia}} = -\sum_{j \neq i} D_{ij}(v_{j(a-1)} + v_{j(a+1)}) - \alpha \sum_{j \neq i} v_{ja}$$

$$v_{ia} = \frac{\exp(\beta u_{ia})}{\sum_{b} \exp(\beta u_{ib})}$$

Mean field annealing

1. Set β to a sufficiently low value, $v_{ia} \approx \frac{1}{N}$, for all i, a
2. Use mean field equation to update all v_{ia}
3. Increase β by an annealing process
4. If a halting condition holds exit otherwise goto step 2

Exercise

- Implement the mean field annealing method for solving the travelling salesman problem

Matlab Coding of MFA optimization for TSP

- Matlab module
  - TSP data generation
  - MFA
  - v_updating
  - V2tour_length

TSP data generation

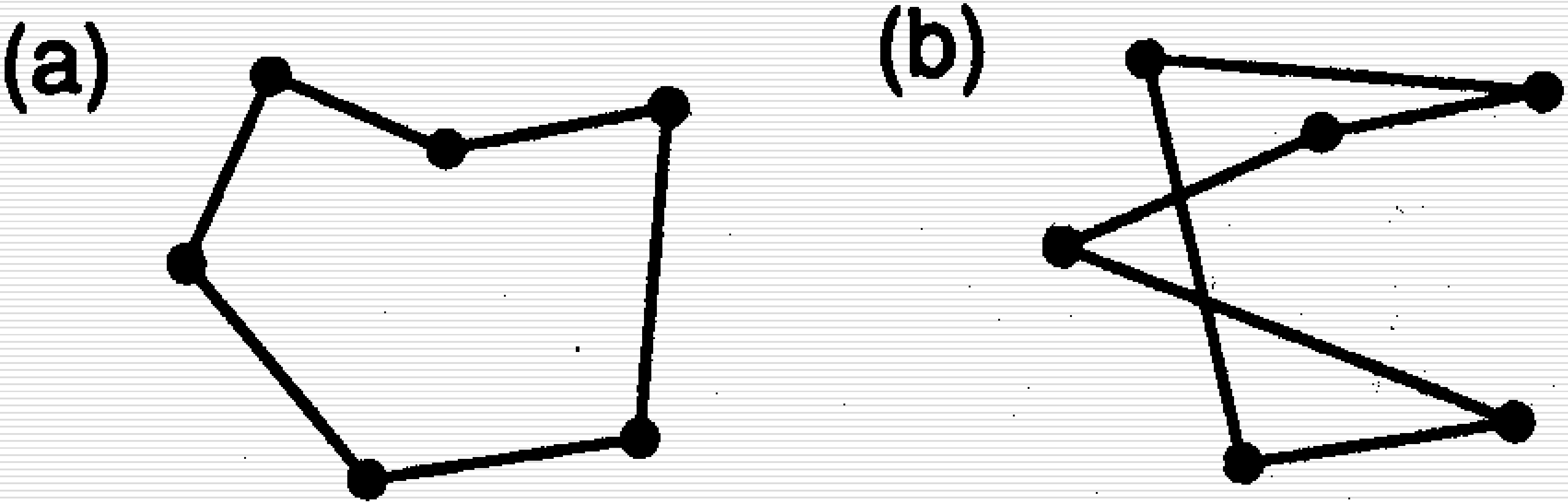
```
function D=TSP_data(N)
```

N: city size

D is a matrix that collect distances between cities

- D(i,i) = 0 for all i
- D(i,j) equals D(j,i)
- D(i,j) measures the distance between cites i and j

TSP



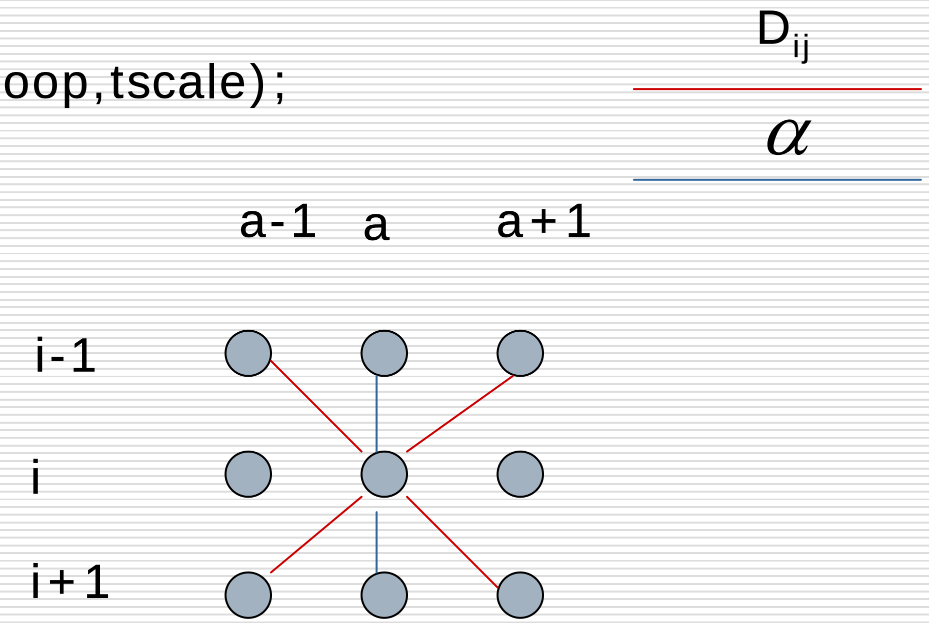
Energy function

$$E = \sum_{i} \sum_{j \neq i} D_{ij} \sum_{a} v_{ia} v_{j(a+1)}$$

$$+ \alpha \sum_{a} \sum_{i} \sum_{j \neq i} v_{ia} v_{ja}$$

Call MFA

MFA_Potts(15,D,loop,tscale);



MFA_Potts

function MFA_Potts(temp,D,loop,tscale);

- temp : temperature
- D: distance matrix
- loop: 50
- tscale: annealing factor

Annealing schedule

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

v2tour_length

function [vd,tour_length]= v2tour_length(v,D)

- v: state matrix
- D: distance matrix
- tour_length
- vd : each column of v contains only one active bit

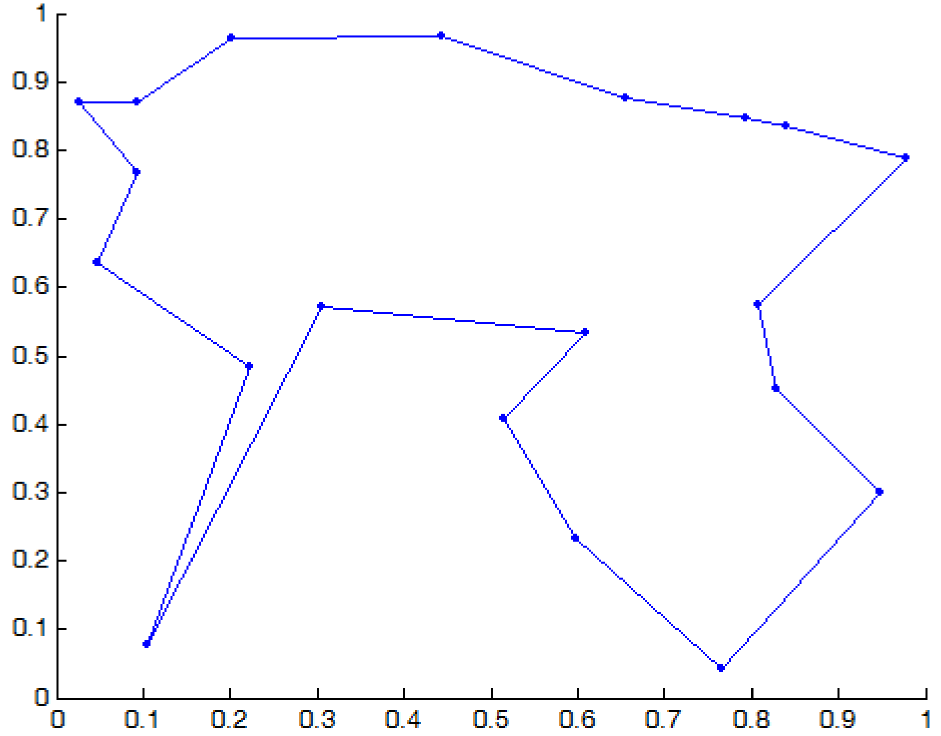
Mean field equation

$$u_{ia} = -\frac{dE}{dv_{ia}} = -\sum_{j \neq i} D_{ij}\left(v_{j(a-1)} + v_{j(a+1)}\right) - \alpha \sum_{j \neq i} v_{ja}$$

$$v_{ia} = \frac{\exp(\beta u_{ia})}{\sum_{b} \exp(\beta u_{ib})}$$

function [v]=update_v_tsp(N,temp,A,v,loop,D)

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

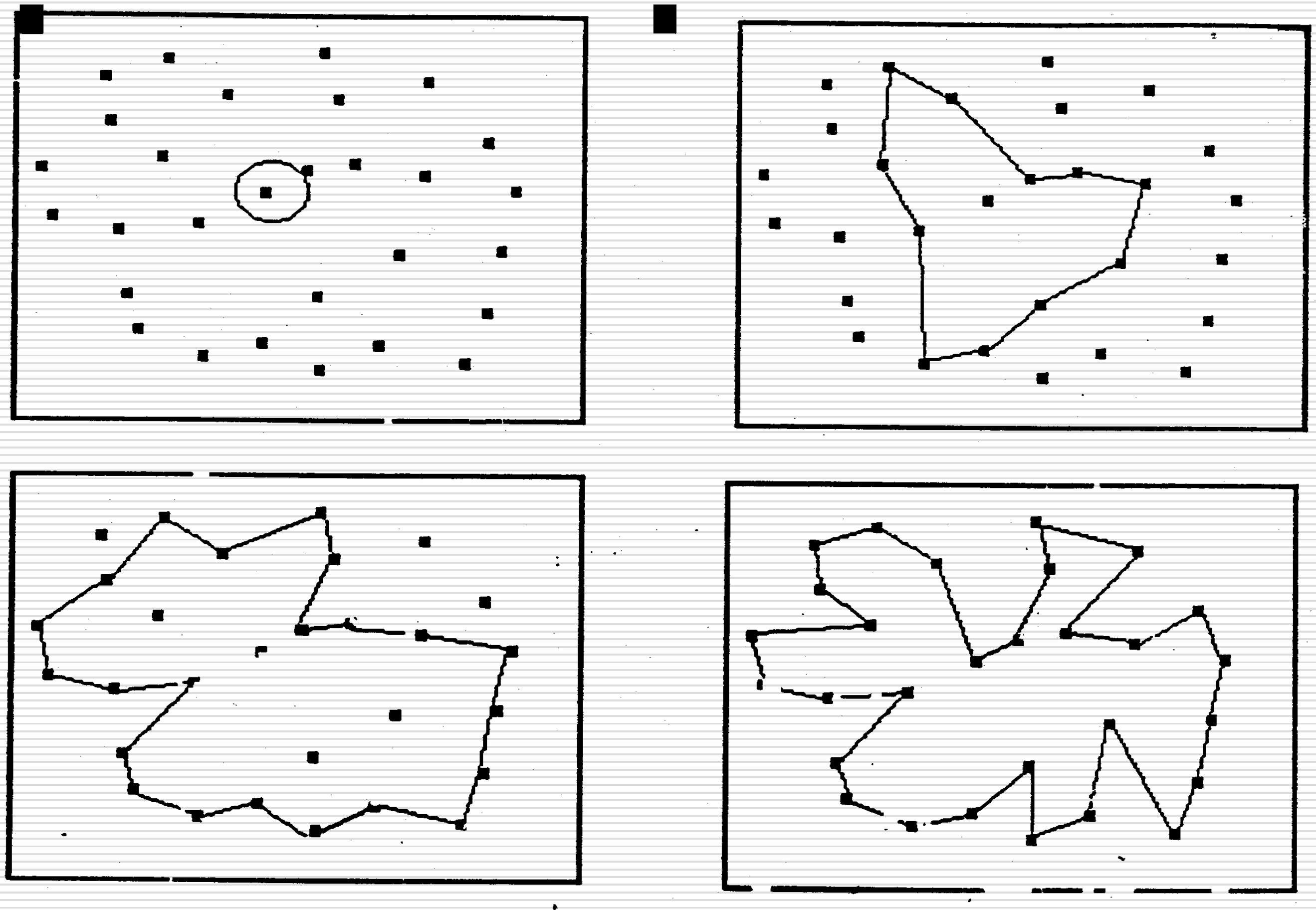

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Elastic Net

R. Durbin and G. Willshaw," A dimensional reduction framework for understanding cortical maps," Nature Vol. 343 15 Feb (1990)

Elastic Net Tutorial

Example



Elastic net algorithm

Updating rule:

$$\Delta w_i = \eta(\sum_u \Lambda^u(i)(\xi^u - w_i) + \kappa(w_{i+1} - 2w_i + w_{i-1})$$

A normalized Gaussian form

$$\Lambda^u(i) = \frac{\exp(-\left|\xi^u - w_i\right|^2 / 2\sigma^2)}{\sum_j \exp(-\left|\xi^u - w_j\right|^2 / 2\sigma^2)}$$

A cost function

$$E\{w_i\} = -\sigma^2 \sum_u \log[\sum_i \exp(-\left|\xi^u - w_i\right|^2 / 2\sigma^2)] + \frac{\kappa}{2} \sum_i \left|w_{i+1} - w_i\right|^2$$

MFA unsupervised learning

$\delta_{ia} = 1$ if *city* i is visited at the ath stop

$$E = \sum_i \sum_a \delta_{ia} \left\| x_i - w_a \right\|^2 + \frac{A}{2} \sum_a \left(\left\| w_a - w_{a+1} \right\|^2 + \left\| w_a - w_{a-1} \right\|^2 \right)$$

$$\sum_a \delta_{ia} = 1$$

$$\sum_i \delta_{ia} = 1$$

A hybrid energy function

- Discrete Potts variables
- Continuous geometrical variables

Unsupervised MFA Learning

- Step1: derive mean field equation
- Step2: derive the updating rule of continuous variables

Revised energy function

$$\delta_{ia} = 1 \text{ if } city \text{ i is visited at the ath stop}$$

$$E = \frac{1}{2}\sum_{i}\sum_{a}\delta_{ia}\left\|x_i - w_a\right\|^2 + \frac{\alpha}{2}\left(\sum_{i}\delta_{ia} - 1\right)^2$$

$$+ \frac{\mathrm{A}}{2}\sum_{a}\left(\left\|w_a - w_{a+1}\right\|^2 + \left\|w_a - w_{a-1}\right\|^2\right)$$

$$E(\boldsymbol{\delta}, \mathbf{w}) = \frac{1}{2}\sum_{i}\sum_{a}\delta_{ia}\left\|x_i - w_a\right\|^2 + \alpha\sum_{a}\sum_{i}\sum_{j\neq i}\delta_{ia}\delta_{ja}$$

$$+\frac{\mathrm{A}}{2}\sum_{a}\left(\left\|w_a - w_{a+1}\right\|^2 + \left\|w_a - w_{a-1}\right\|^2\right)$$

Mean Energy

$$E(\mathbf{v},\mathbf{w}) = \frac{1}{2}\sum_{i}\sum_{a} v_{ia}\left\| x_i - w_a \right\|^2 + \alpha \sum_{a}\sum_{i}\sum_{j \neq i} v_{ia} v_{ja}$$

$$+ \frac{\mathrm{A}}{2}\sum_{a}\left(\left\| w_a - w_{a+1} \right\|^2 + \left\| w_a - w_{a-1} \right\|^2 \right)$$

Mean field equation

$$u_{ia} = -\frac{dE}{dv_{ia}} = -\left\| \mathbf{x}_i - w_a \right\|^2 - \alpha \sum_{j \neq i} v_{ja}$$

$$v_{ia} = \frac{\exp(\beta u_{ia})}{\sum_b \exp(\beta u_{ib})}$$

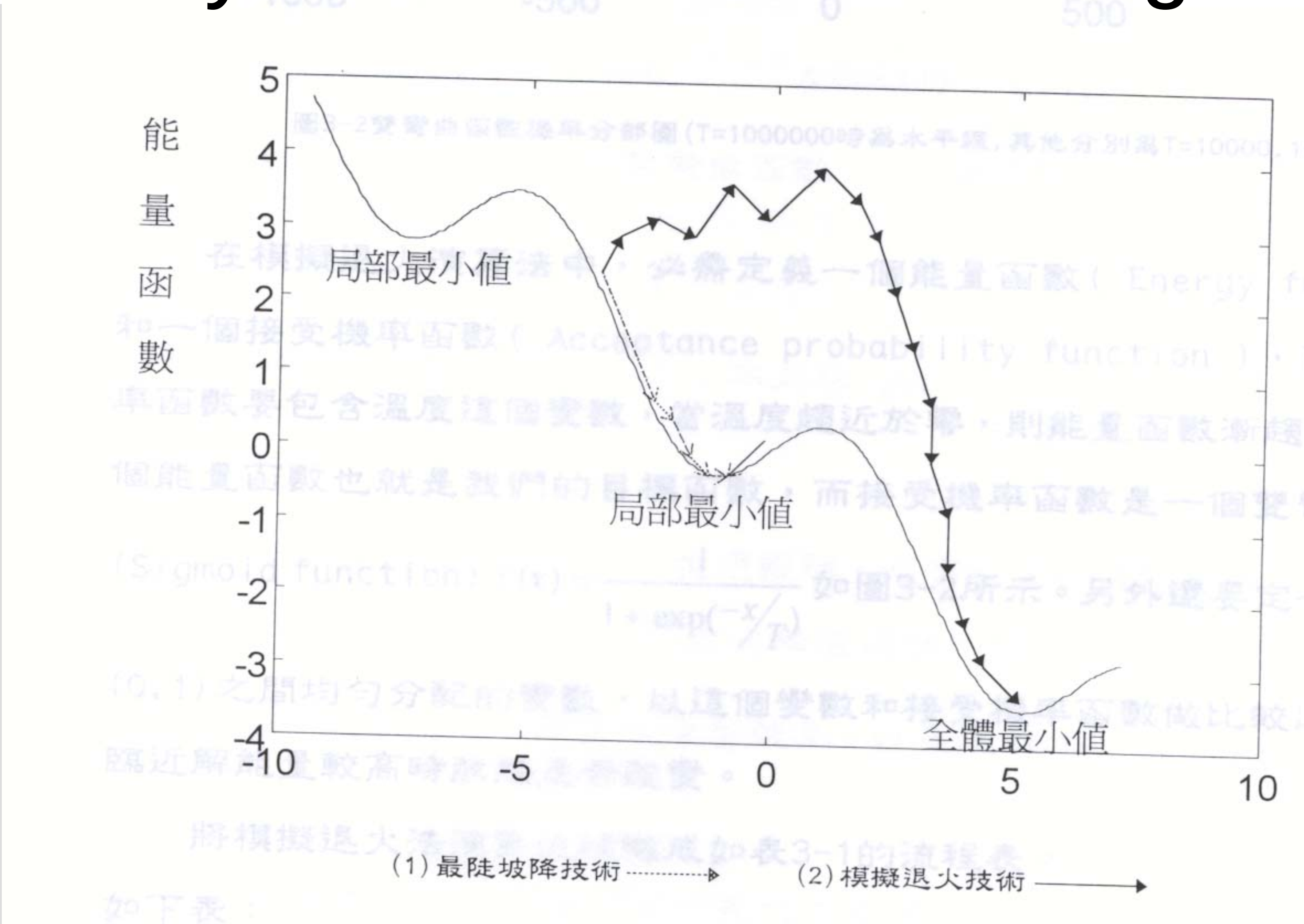
Updating rule

$$\frac{dE(\mathbf{v},\mathbf{w})}{d\mathbf{w}_a} = \sum_i v_{ia}(\mathbf{x}_i - \mathbf{w}_a)$$

$$+ \mathrm{A}(2\mathbf{w}_a - \mathbf{w}_{a+1} - \mathbf{w}_{a-1}) = 0 \text{ for all } a$$

$$solve \;\; all \; \mathbf{w}_a$$

TSP by simulated annealing



Acceptance probability function

(Sigmoid function) $f(x)=\dfrac{1}{1+\exp(-x/T)}$

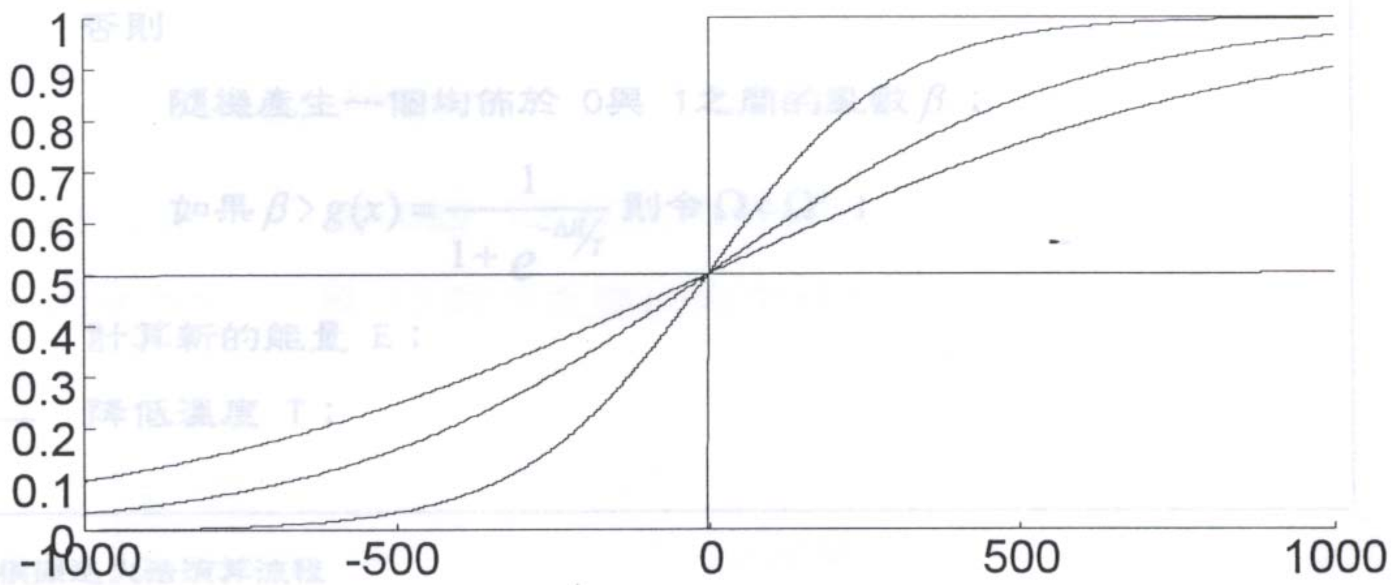


圖3-2雙彎曲函數機率分部圖(T=1000000時爲水平線,其他分別爲T=10000,100,1及0.01)

設定一個初始解Ω；

設定模擬退火的初始溫度 T；

→重覆下列步驟，直到能量函數趨於收斂

選擇一個與Ω鄰近的解Ω'；

計算能量差距ΔE=E(Ω')−E(Ω)；

如果ΔE<0 則令Ω=Ω'；

否則

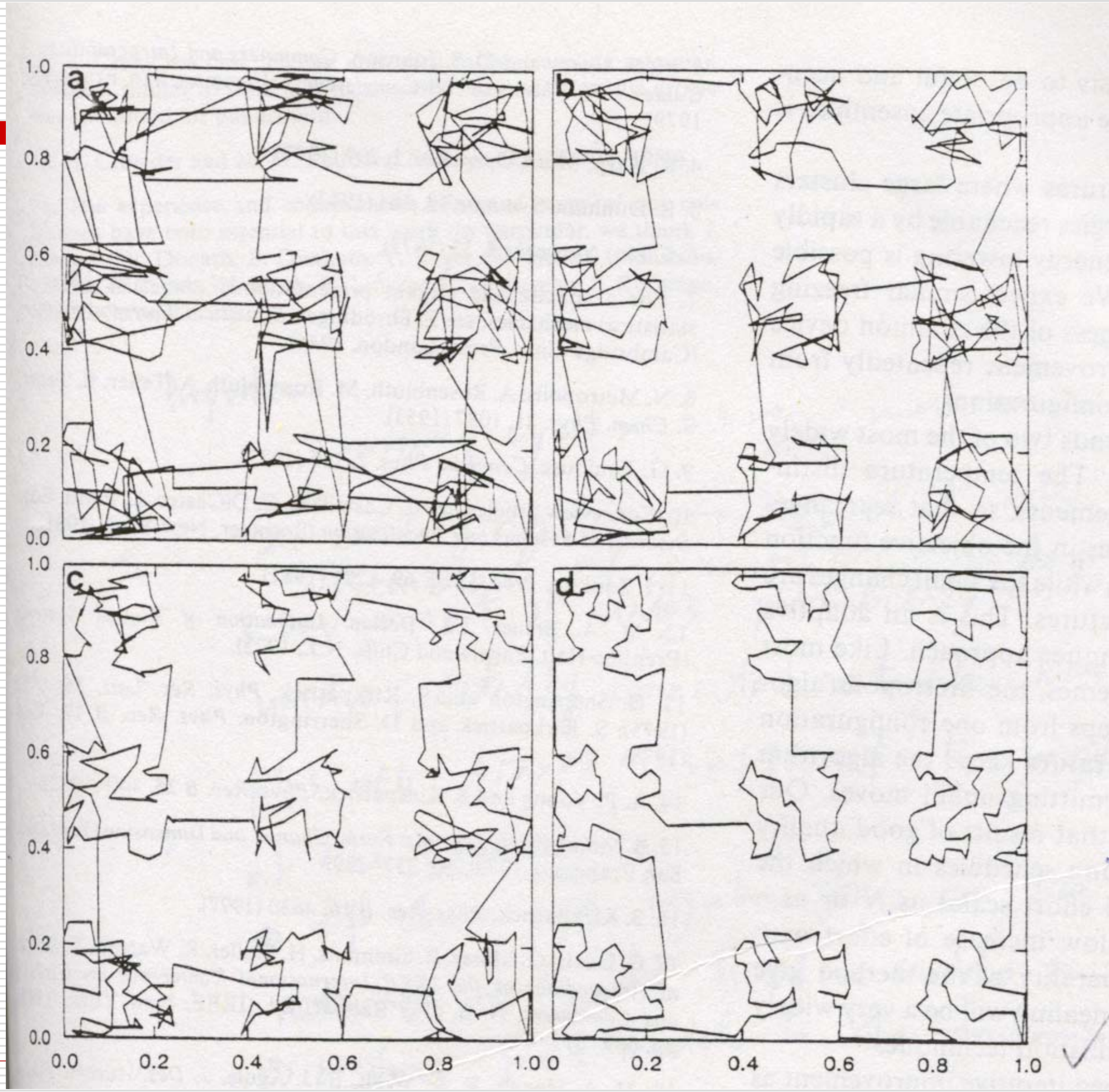
隨機產生一個均佈於 0與 1之間的亂數 β；

如果 $\beta > g(x) = \dfrac{1}{1+e^{-\Delta E/T}}$ 則令Ω=Ω'；

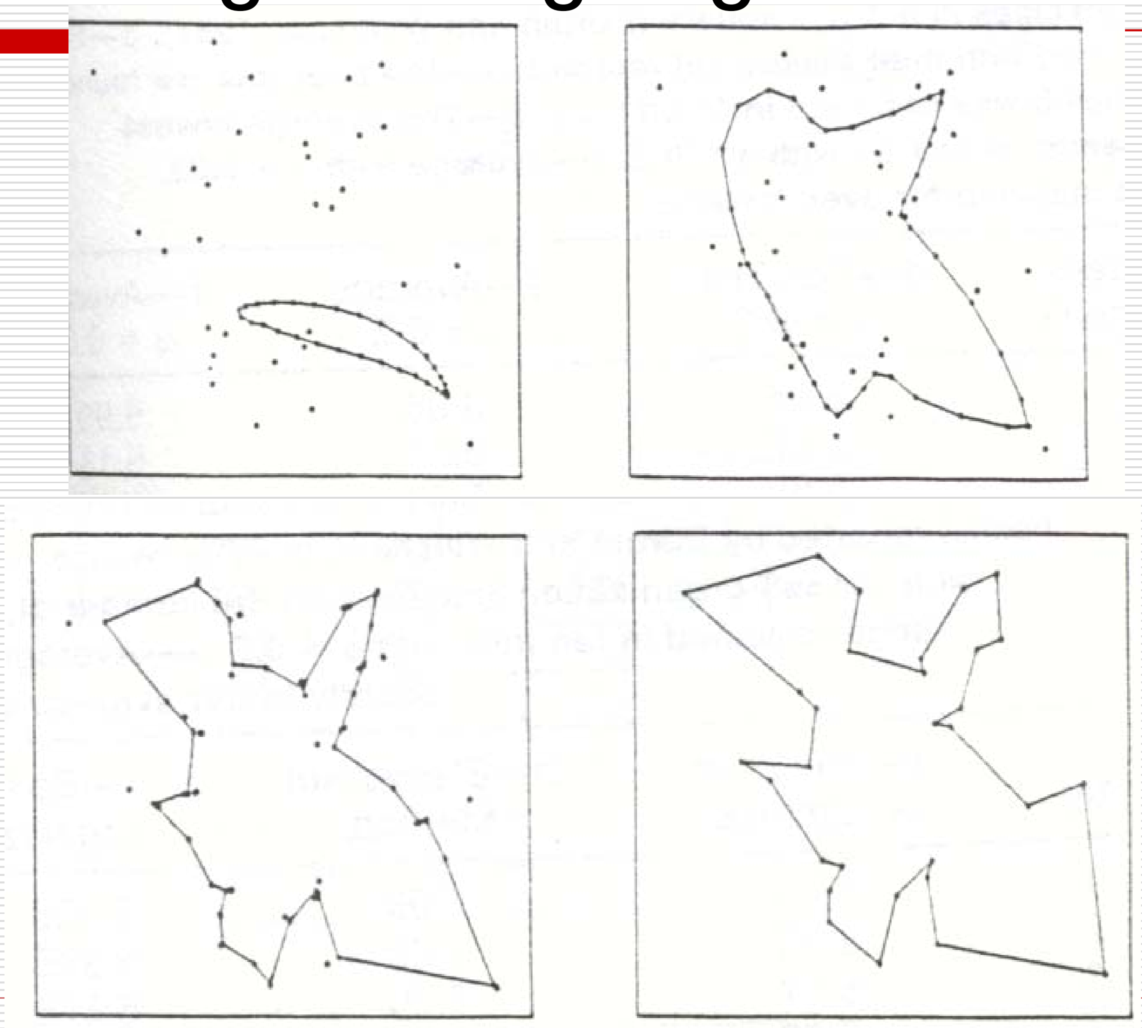
計算新的能量 E；

降低溫度 T；

Kirkpatrick's Result



Self-organizing algorithm for TSP



Step 1: Find the node j_c which is closest to city i: for each node j, compute its *potential*:

$$V_j = (x_1^i - c_1^j)^2 + (x_2^i - c_2^j)^2$$

determine j_c by competition:

$$V_{j_c} = \min V_j .$$

Step 2: Move node j_c and its neighbors

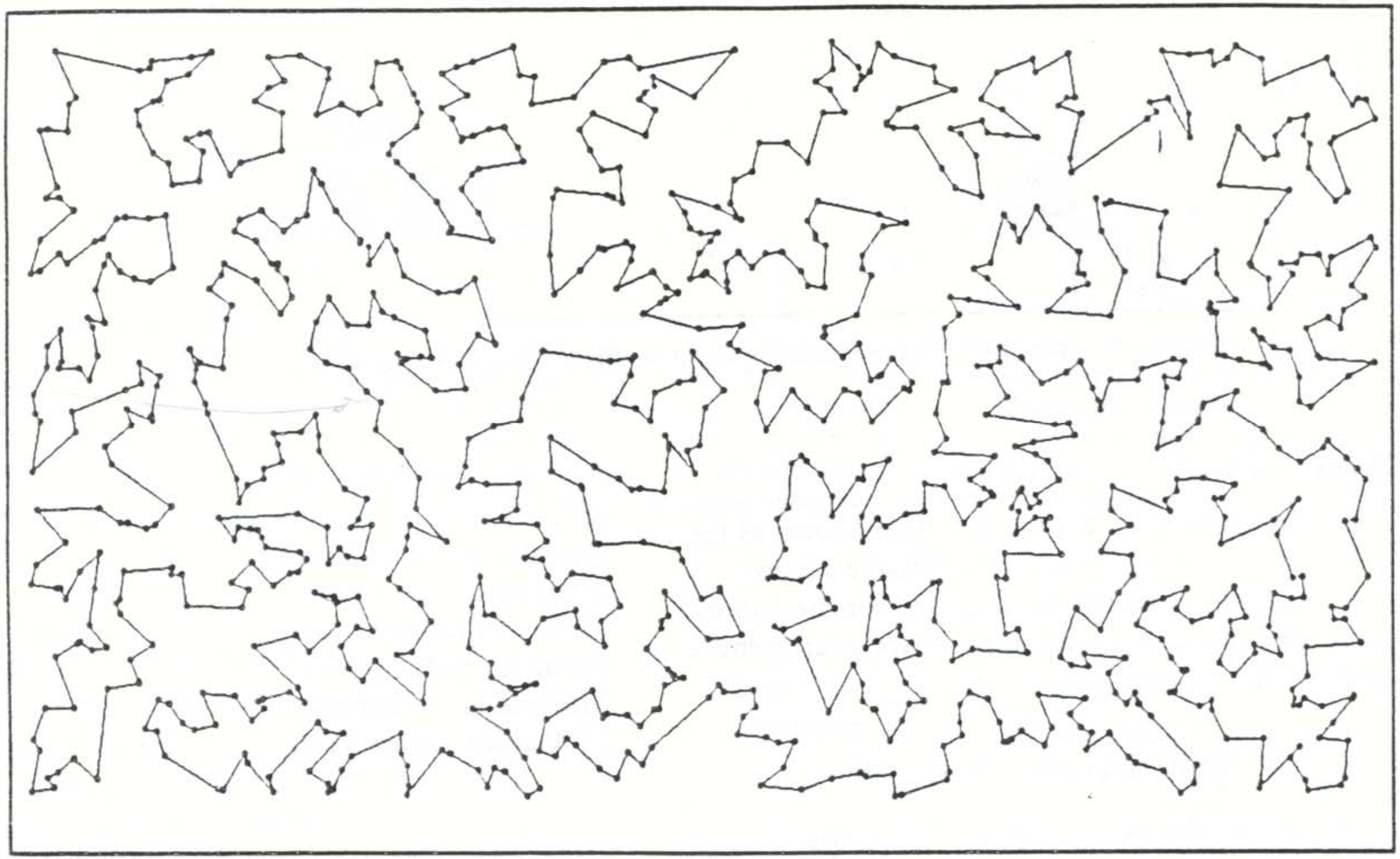
$$f(G, n) = (1/\sqrt{2}) \cdot \exp(-n^2/G^2).$$

$$c_k^j \leftarrow c_k^j + f(G, n) \cdot (x_k^i - c_k^j).$$

$$n = \inf(j - j_c \pmod N, j_c - j \pmod N).$$

Decreasing the gain $\quad G \leftarrow (1 - \alpha) \cdot G.$

Result of SOM



Natural gradient descent

(a) Amari's works for independent component analysis

(b) norm in Riemannian space

$$\|dw\|^2 = \sum_{ij} g_{ij}(w) dw_i dw_j$$

denotes the square length of a small incremental vector dw connecting w and w + dw

natural gradient descent

$$
\begin{aligned}
w_{t+1} &= w_t - \eta_t C(w_t) \nabla l(z_t, w_t) \\
C(w_t) &= G^{-1}(w) \\
G &= [g_{ij}(w)]
\end{aligned}
$$

Nondeterministic method

simulated annealing:

(a) g(x):probability density of state-space of D parameter $x = \{x^i, i = 1, D\}$

$$g(x) = (2\pi T)^{-D/2} \exp(-\Delta x^2/(2T))$$

simulated annealing

(b) h(x):probability density for acceptance of new cost function given the just previous value

$$h(x) = \frac{1}{1 + \exp(\triangle E/T)}$$

(c) T(k):schedule of annealing the temperature T in annealing-time steps

$$T(k) = \frac{T_0}{\ln k}$$