The Traveling Salesman Problem

Problem statement: A travelling salesman must visit every city in his territory exactly once and then return to his starting point. Given the cost of travel between all pairs of cities, find the minimum cost tour.

- NP-Complete Problem
- Exhaustive Enumeration:

  n nodes, $n!$ enumerations,

  $(n-1)!$ distinct enumerations

  $\frac{(n-1)!}{2}$ distinct undirected enumerations

  Example:

  n = 10, 19!/2 = 1.2 x 10^{18}

The Traveling Salesman Problem

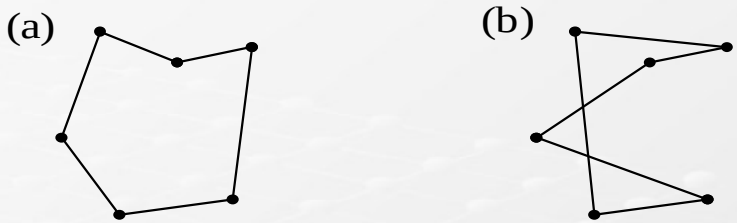
TSP: find the shortest tour connecting a set of cities.

Following Hopfield & Tank (1985) a tour can be represented by a permutation matrix:

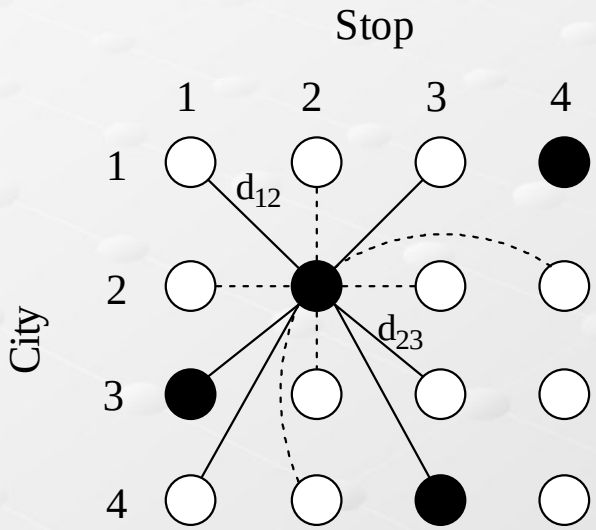
	1	2	3	4
A	0	1	0	0
B	1	0	0	0
C	0	0	1	0
D	0	0	0	1

$\longleftrightarrow$ Tour: BACD

The Traveling Salesman Problem



The TSP, showing a good (a) and a bad (b) solution to the same problem



Network to solve a four-city TSP. Solid and open circles denote units that are on and off respectively when the net 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.

Artificial Neural Network Solution

Solution to *n*-city problem is presented in an *n* x *n* permutation matrix V

X = city

i = stop at wich the city is visited

Voltage output: $V_{X,i}$

Connection weights: $T_{Xi,Yj}$

n^2 neurons

$V_{X,i}$ = 1 if city *X* is visited at stop *i*

d_{XY} = distance between city *X* and city *Y*

Artificial Neural Network Solution

- Data term:

  We want to minimize the total distance

$$E_1 = \frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{X,i} \left(V_{Y,i+1} + V_{Y,i-1} \right)$$

- Constraint terms:

  Each city must be visited once

$$E_2 = \frac{A}{2} \sum_X \sum_i \sum_{j \neq i} V_{X,i} V_{X,j}$$

  Each stop must contain one city

$$E_3 = \frac{B}{2} \sum_i \sum_X \sum_{Y \neq X} V_{X,i} V_{Y,i}$$

  The matrix must contain *n* entries

$$E_4 = \frac{C}{2} \left(\sum_X \sum_i V_{X,i} - n \right)^2$$

Artificial Neural Network Solution

- A, B, C, and D are positive constants
- Indici modulo *n*

Total cost function

$$E = \frac{A}{2} \sum_X \sum_i \sum_{j \neq i} V_{X,i} V_{X,j} + \frac{B}{2} \sum_i \sum_X \sum_{Y \neq X} V_{X,i} V_{Y,i} + \frac{C}{2} \left(\sum_X \sum_i V_{X,i} - n \right)^2 + \frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{X,i} \left(V_{Y,i+1} + V_{Y,i-1} \right)$$

La funzione energia della rete di Hopfield è:

$$E = -\frac{1}{2} \sum_{X\,Y} \sum_{i\,j} T_{Xi,Yj} V_{Xi} V_{Yj} - \sum_{X\,i} I_{Xi} V_{Xi}$$

Weights of the Network

The coefficients of the quadratic terms in the cost function define the weights of the connections in the network

$$T_{Xi,Yj} = -Ad_{XY}\left(1-d_{ij}\right) \quad \text{\{Inhibitory connection in each row\}}$$

$$-Bd_{ij}\left(1-d_{XY}\right) \quad \text{\{Inhibitory connection in each column\}}$$

$$-C \quad \text{\{Global inhibition\}}$$

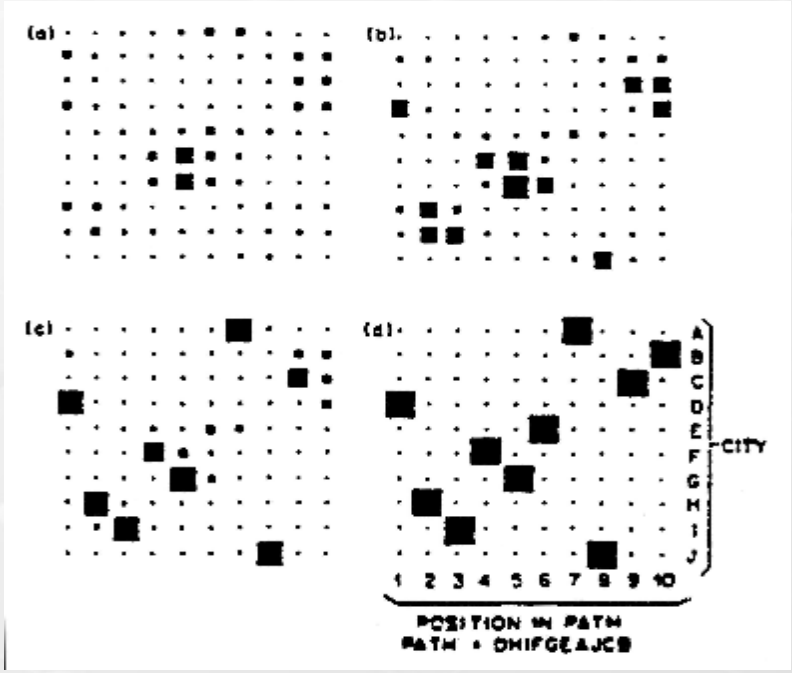
$$-Dd_{XY}\left(d_{j,i+1}+d_{j,i-1}\right) \quad \text{\{Data term\}}$$

$$d_{ij} = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases}$$

$$I_{Xi} = C_n \quad \text{\{Corrente esterna eccitatoria\}}$$

Experiments

- 10-city problem, 100 neurons
- Locations of the 10 cities are chosen randomly with uniform p.d.f. in unit square
- Parameters: A = B = 500, C = 200, D = 500



- The size of the squares correspond to the value of the voltage output at the corresponding neurons.
- Path: D-H-I-F-G-E-A-J-C-B

TSP – Un'altra formulazione

Un altro modo per esprimere i vincoli del TSP (cioè matrice di permutazione) è il seguente :

$$E_2 = \frac{A}{2} \sum_X \left(\sum_i V_{X\,i} - 1 \right)^2$$ vincolo sulle righe

$$E_3 = \frac{B}{2} \sum_i \left(\sum_X V_{X\,i} - 1 \right)^2$$ vincolo sulle colonne

La funzione energia diventa :

$$E = \frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{X\,Y}\, V_{X\,i} \left(V_{Y\,i+1} + V_{Y\,i-1} \right) + E_2 + E_3$$

Vantaggio : minor numero di parametri (A,B,D)

Problema delle n regine

Si può costruire una rete n x n in cui il neurone (i, j) è attivo se e solo se una regina occupa la posizione (i, j)

Ci sono 4 vincoli :

1. solo una regina in ciascuna riga
2. solo una regina in ciascuna colonna
3. solo una regina in ciascuna diagonale
4. esattamente n regine sulla scacchiera

Problema delle n regine

La matrice dei pesi si determina così :

$$
\begin{aligned}
-T_{ij,kl} = {} & A\,(1-d_{jl})\,d_{ik} + \\
& B\,d_{jl}\,(1-d_{ik}) + \\
& C + \\
& D\,(d_{i+j,k+l} + d_{i-j,k-l})\,(1-d_{ik})
\end{aligned}
$$

A ≡ peso inibitorio sulle righe

B ≡ peso inibitorio sulle colonne

C ≡ peso inibitorio “globale”

D ≡ peso inibitorio sulle diagonali

Reti di Hopfield per ottimizzazione

Problemi associati con il modello di Hopfield originale :

1) numero di connessioni $O(n^4)$ e numero di unità $O(n^2)$

2) difficoltà per la determinazione dei parametri A, B, C, D

3) le soluzioni ottenute sono raramente “matrici di permutazione”

4) difficoltà nell’evitare i minimi locali