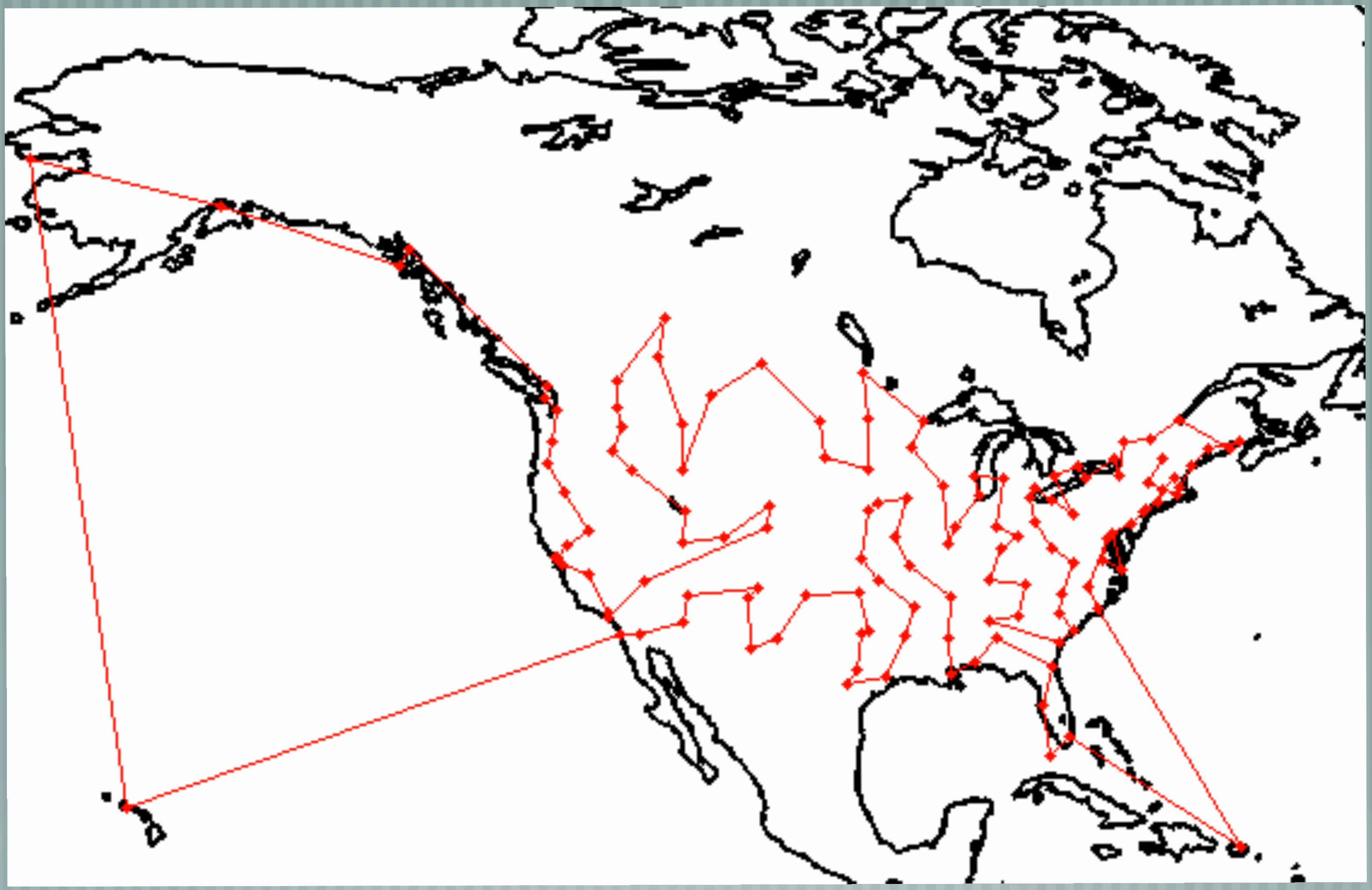


Self-Organizing maps strategy for solving the traveling salesman problem

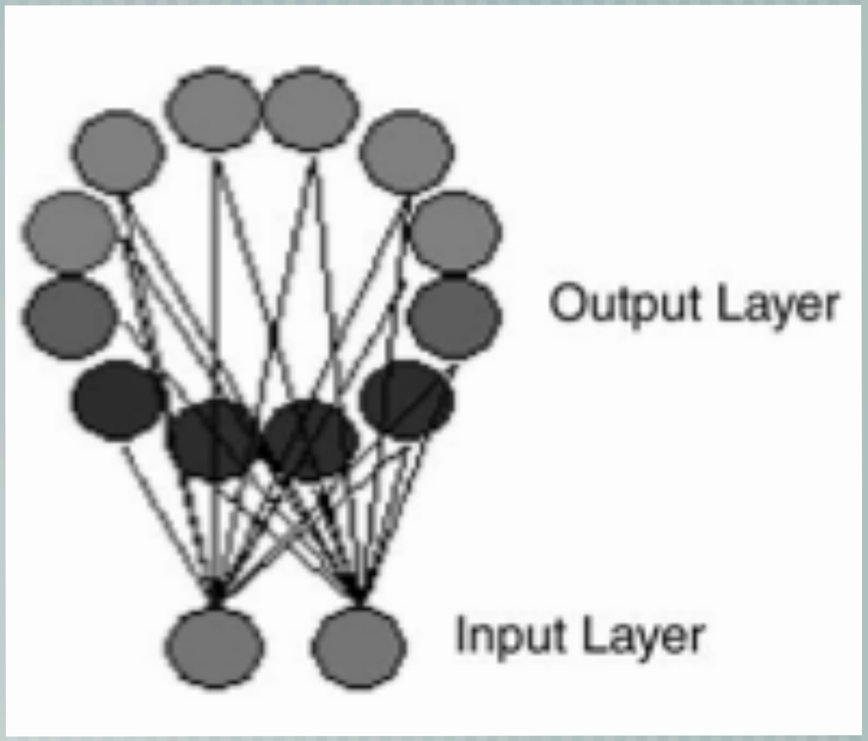
Tina Khajeh 93210761

Introduction To TSP

- ***The traveling salesman problem (TSP) is a classical optimization problem.***
- ***Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once.***
- ***The decision problem form of TSP is a NP complete (non - deterministic polynomial - time hard) problem, hence the great interest in efficient heuristics to solve it.***

SOM Applied To TSP

- using a two-layer network, which consists of a two-dimensional input unit and m output units.
- The evolution of the network may be geometrically imagined as stretching of a ring toward the coordinates of the cities.
- Output neurons place on ring.
- The input neurons receive the coordinates of cities.

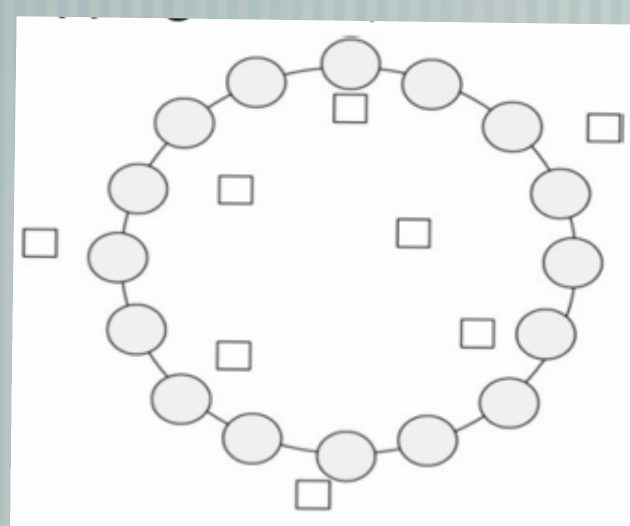


SOM Applied To TSP (con't)

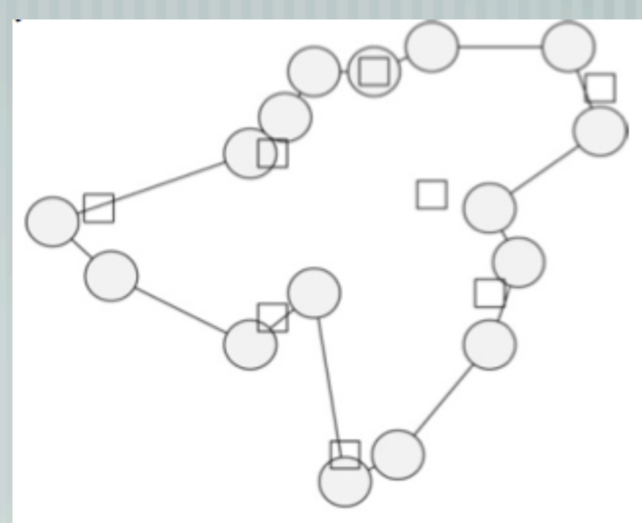
- **competition based on Euclidian distance**
- **use neighborhood function f as below:**

$$f(\sigma,d)=e^{\left(\frac{-d^2}{\sigma^2}\right)} \qquad d=\min\{\|j-J\|,\, m-\|j-J\|\}$$

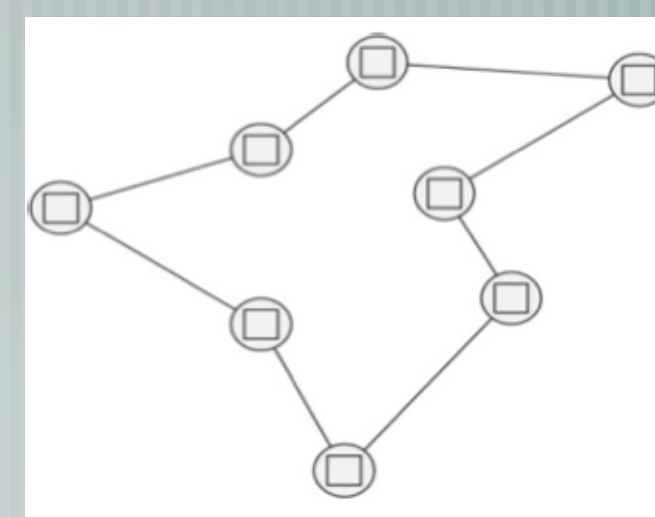
- ***evolution of the algorithm is presented :***



a) Initial stage



b) Intermediate stage



c) Final stage

Parameter Adaption

- The SOM algorithm has two adaptive parameters, the learning rate and the neighborhood function variance .

- Kohonen proposed an exponential evolution of this parameters of form :

$$\alpha_k = \alpha_0 \times \exp\left(-\frac{k}{\tau_1}\right),$$

$$\sigma_k = \sigma_0 \times \exp\left(-\frac{k}{\tau_2}\right),$$

- During experiments considering the SOM applied to the TSP there were identified new parameter adaptation heuristics, which have led to a faster convergence

$$\alpha_k = \frac{1}{\sqrt[4]{k}},$$

$$\sigma_k = \sigma_{k-1} \times (1 - 0.01 \times k), \quad \sigma_0 = 10.$$

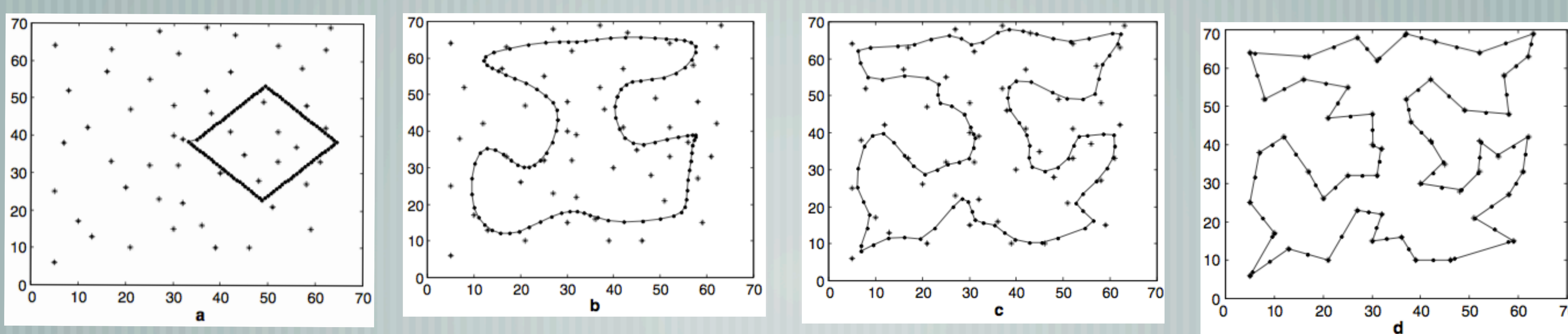
Algorithm Initialization

- **Selecting the number of nodes of neurons and neighbor length**
  - selecting the number of nodes as twice of the number of cities (m = 2n)
  - effective neighbor length, which refers to the number of nodes that should be updated when a new winner is selected, is limited to 40% of the nodes. And the neighbor length also decreases gradually
- **Index of winner neurons and effect of the initial ordering of the cities**
  - In order to prevent nodes from being selected as the winner for more than one city in each complete cycle of iterations, an inhibitory index is defined for each node.

Algorithm Initialization(con’t)

The initialization method of synaptic weights of neurons

- what kind of cures is best to use for a TSP with a given number of cites, based on a well-defined performance such as the minimum execution time or minimum tour length?
- random initialization of neurons
- random initialization of neurons on a circular ring
- initialization of neurons on a rhombic frame located in the right of the n cities’s centroid



Computational experiments

The modified growing ring SOM approach for TSP (MGSOM) is briefly described as follow:

- **Initialization: n is the number of cites algorithm inputs are the cartesian coordinate of the set of n cites**
  - **m(t) is the number of nodes on the rhombic frame located in the right of the n cities' s centroid (m(0) = n, initially).**
  - **initial value of gain parameter, r_0 = 10**
  - **t = 1 and t_{min} be the value of the desired minimum number of iteration**
- **Randomization: Starting calculation with different initial ordering of the cities**
  - **reset the inhabitation status of all nodes to false**

Computational experiments(con't)

- **Parameters Adaption:** The learning rate and the neighborhood function variance are calculated using

- **Competition:** select the closed node to city i based on Euclidean distance.
  - This competition is held among only nodes, which have not been selected as a winner in this iteration

- **Adaptation:** synaptic vectors of the winner and its neighbors are updated
  - the neighbor length of the node J is limited to 40% of m(t)
  - $C_j(t)$ is a signal counter for inspection of the learning history
  - The signal counter of the winner is updated and other signal counters preserve their values at time t.

$$C_j(t+1) = \begin{cases} C_j(t) + 1, & \text{if } i = J \\ C_j(t), & \text{otherwise} \end{cases}$$

Computational experiments(con ’ t)

- **Insertion a node:** Inserts a novel node at every T_{int} learning times hereby the map can grow up to $m(t) = 2n$.
    - At $t = kT_{int}$, k is a positive integer, we determine a node p whose signal counter has the maximum value.
  - take a neighbor $f = p - 1$, novel node r is inserted between p and f .The synaptic vector of r is initialized as follow:

$$y_r = 0.5(y_p + y_f).$$

  - Counter values of p and r are re-assigned as follows:

$$C_p(t+1) = 0.5C_p(t), \qquad C_r(t+1) = 0.5C_p(t).$$

- Set i=i+1. If i<n+1, go to step 4. Otherwise, set t=t+1 and go to the next step.
- **Convergence test:** Return to step 2 while t 6 tmin.

Implementation's Results:

Map	Optimal	Implementation
Burma14	3323	3412
ulysses16	6859	6859
bays29	2020	2131
fri26	937	1102

Results :

TSP instances used for computational experiments

Instances	No. of cities	Optimum length
Bier127	127	118,282
Eil51	51	426
Eil76	76	538
kroA200	200	29,368
Lin105	105	14,379
pcb442	442	50,778
Pr107	107	44,303
Pr6	6	96,772
Pr152	152	73,682
Rat195	195	2323
ed100	100	7910
st70	70	675

Results from PKN, GN, AVL and MGSOM applied to TSP instance

Instances	PKN	GN	AVL	MGSOM
Bier127	122211.7	155163.2	122673.9	119,580
Eil51	443.9	470.7	443.5	431.9569
Eil76	571.2	614.3	571.3	556.2061
kroA200	30927.8	39370.2	30994.9	29,947
Lin105	15374.2	15469.5	15311.7	14,383
pcb442	–	–	59649.8	55,133
Pr107	44504.3	80481.3	45096.4	44,379
Pr136	103878.0	5887.7	103442.3	98,856
Pr152	74804.2	105230.1	74641.0	74,228
Rat195	–	–	2681.2	2462
rd100	87.9	8731.2	8265.8	8002.7
st70	692.8	755.7	693.3	682.9886

Results:

Results from KL, KG, SETSP and MGSOM applied to TSP instances

Instances	KL	KG	SETSP	MGSOM
Bier127	121548.7	121923.7	120470	119,580
Eil51	438.2	438.2	435.46	431.9569
Eil76	564.8	567.5	560.78	556.2061
kroA200	30200.8	30444.9	30,284	29,947
Lin105	14664.4	14564.6	14,566	14,383
pcb442	56399.9	56082.9	55,937	55,133
Pr107	44628.3	44491.1	44,484	44,379
Pr136	101156.8	101752.4	101,030	98,856
Pr152	74395.5	74629.0	74,543	74,228
Rat195	2607.3	2599.8	2583	2462
rd100	8075.7	8117.4	8115.7	8002.7
st70	685.2	690.7	685.8	682.9886

Average deviations from the optimal tour length

Instances	KL	KG	SETSP	MGSOM
Bier127	2.76	3.08	1.85	1.0936
Eil51	2.86	2.86	2.22	1.3983
Eil76	4.98	5.48	4.23	3.3840
kroA200	2.84	3.67	3.12	1.9715
Lin105	1.98	1.29	1.30	0.0278
pcb442	11.07	10.45	10.16	8.5770
Pr107	0.73	0.42	0.41	0.1719
Pr136	4.53	5.15	4.40	2.1530
Pr152	0.97	1.29	1.17	0.7412
Rat195	12.24	11.92	11.19	5.9847
rd100	2.09	2.62	2.60	1.1718
st70	1.51	2.33	1.60	1.1835
Total average	4.05	4.21	3.69	2.3215

References

- Bai, Yanping, Wendong Zhang, and Zhen Jin. "An new self-organizing maps strategy for solving the traveling salesman problem." *Chaos, Solitons & Fractals* 28.4 (2006): 1082-1089.