A New Non-Autonomous Chaotic Neural Network for Solving TSP

Kushan Ahmadian [1], Abbas Golestani [1], M.R.Jahed Motlagh [1] and Nasser Mozayani [1]
[1]Computer faculty, Iran University of Science and Technology, Tehran, Tehran, Iran

Abstract - *In this paper we propose a new chaotic noise injection strategy into the Hopfield network which belongs to a general class of chaotic neural networks called "non-autonomous network". The objective of this method is to inject noise into the neurons based on the history of dynamics of their neighbor neurons. The measure of adjacency which is a criterion for definition of neighborhood, is based on the co-existing of neurons in a same cluster and those clusters are indicated by an ISODATA algorithm. Experimental results show efficiency of the proposed algorithm, especially in maps with separated clusters.*

Keywords: Traveling salesman problem, chaotic neural networks, Optimization.

1 Introduction

Recently many researchers have focused on chaotic neural network for minimum searching and consequently the traveling salesman problem (TSP) has been aimed to be solved by this class of neural networks. Initially, the Hopfield neural network [1][2] was found to be useful to solve TSPs, however, due to the large number of local minima in the energy function of TSP, it often trapped into of them. Consequently many methods have been proposed to overcome this problem and as a result chaotic neural networks which were proved to have the ability to escape local minima have been aimed. There are varieties of models of chaotic neural which have been used for solving optimization problems.

In [3] Aihara et al, has introduced a chaotic neural network which exhibits chaotic behavior using a negative self-coupling and gradually removing it. The model is based on the chaotic behavior of some biological neurons and has been successfully applied to several optimization problems e.g. TSP [4] and their method outperformed the Hopfield neural network in both the efficiency and solution quality and comparing to other neural networks, their approach significantly increased the probability of finding near-optimal solutions. Afterward Nozawa [5], introduced a new CNN by applying the Euler discretization in order to simulate continues and their results from the TSP method proofed the efficiency of their algorithm. Moreover, Chen and Aihara [6], introduced a chaotic simulated annealing approach which was firstly, based on a transiently chaotic phase and second, a convergence phase which tended to find the global minimums. Their method has been successfully applied to the TSP and machine maintenance problem [7]. Chen and Aihara also showed the existence of strange attractors and network stability conditions [8] suggested the dynamical phenomenon of crisis-induced intermittency to be the underlying mechanism for the chaotic switching among the minima.

The above methods belong to the "autonomous methods" in which the network searches for the global minimum by modifying the characteristic of dynamics of each neuron. However, there is another category of networks called "non-autonomous methods" in which the network avoids the local minima by adding noise to each neuron. Recently, this kind of networks is paid more attention and Hayakawa et al. [9] pointed out the intermittency chaos near the periodic-3 window of the logistic map gains the best performance for TSP. Ueta et al. [10], showed that effective intermittency chaotic noise also exists for period 5 and period 7.

Zhou et al. [11] added a chaotic noise using a chaotic time series generated by the Henon map, to the network as an external approach, in contrast to the internal approaches described previously in which the chaotic behavior is generated internally. They applied their method to the 100-city TSP problem and the obtained results showed superior optimization ability compared to the Boltzmann machine.

In this paper we introduce a new class of non-autonomous chaotic networks where the chaotic noise injection is based on the behavior of neighbor neurons. This method is especially useful for a class of optimization problems in which state changing of neurons affects neuron with limited logical distances e.g. broadcasting problem, TSP and etc. the outline of this paper is as follows: in next section we introduce the noise injection strategies for TSP and then the algorithm is introduced. In section 3 we have presented the experimental results which are based on different types of maps and finally section 4 presents the conclusion remarks.

2 Using noise in HNN for solving TSP

2.1 Chaotic noise with HNN

To overcome the shortcomings of classic Hopfield neural networks, chaotic dynamics has been applied to HNN networks by many authors. In order to fire (i,j) neuron the energy function is defined as follow:

$$E = \sum_{i,j}^{N}\sum_{k,l}^{N} w_{ijkl} x_{ij} x_{kl} - \sum_{ij}^{N} \theta_{ij} x_{ij} \quad (1)$$

In which w_{ijkl} is the coefficient between neurons (i,j) and (k,l), θ_{ij} is the threshold value and the weight coefficients are defined as follow :

$$w_{ikjl} = -A\{\delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij})\} - Bd_{ij}(\delta_{l,k+1} + \delta_{l,k-1}) \quad (2)$$

Where *A, B>0* are arbitrary coefficients. Considering the above equation the dynamics of single neuron can be defined as follow:

$$x_{ik}(t+1) = f\left(\sum_{j=1}^{N}\sum_{l=1}^{N} w_{ikjl} x_{jl}(t) + \theta_{ij} + \varepsilon\right) \quad (3)$$

Where ε, is the additional noise and *f* is a sigmoid function defined as follows:

$$f(x) = \left(1 + \tanh\frac{x}{\mu}\right) \quad (4)$$

Here μ is a variable parameter of SA noise. To indicate which neuron is to be fired, the state of neuron $x_{ik}(t)$ at time t is described as follows:

$$\bar{x}_{ik}(t) = \frac{1}{\tau}\sum_{l=t-\tau+1}^{t} x_{ik}(l)$$
$$\hat{x}_{ik}(t) = \begin{cases} 1 & ,if \quad \bar{x}_{ij}(y) \ge \bar{x}(t) \\ 0 & , \quad otherwise \end{cases} \quad (5)$$

Here $\hat{x}_{ik}(t)$ is new state of $x_{ik}(t)$, where $\bar{x}_{ik}(t)$ is the *(i,j)*th neuron's average value of output from $t-\tau$ to *t*. Furthermore, $\bar{x}(t)$ is *N-th* value in the $\bar{x}_{ij}(t)$ of $N \times N$ neurons and $\tau = N$.

In order to insert a chaotic noise, a time series is generated by a logistic map:

$$z_{ik}(t+1) = az_{ik}(t)(1 - z_{ik}(t)) \quad (6)$$

Here *a*, is a bifurcation parameter of the logistic map and dynamic of HNN with chaotic noise is given as:

$$x_{ik}(t+1) = f\left(\sum_{j=1}^{N}\sum_{l=1}^{N} w_{ikjl} x_{jl}(t) + \theta_{ij} + \varepsilon + \beta z_{ik}(t)\right) \quad (7)$$

Where, β is amplitude of noise.

2.2 conditional chaotic noise

One important drawback of these methods is their *blind* noise operations. Here the chaotic or stochastic noises are injected into the network regardless of neuron's previous behavior. In the proposed method, and in order to rationalize the process of noise injection, we first introduce a matrix of adjacency as follows:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} \quad (8)$$

Where, $a_{ij} = \begin{cases} 1, \text{ if the i-th and j-th cities are adjacent} \\ 0\text{ , else} \end{cases}$

The meaning of adjacency is not restricted to the physical distance of objects. This criterion can be redefined based on the scope of the optimization problem. For instance in an N-queen problem, since changing position of one queen may influence any other queen on the board, all entries of the matrix should be one in order to take effect by each other.

The adjacency matrix has been used to evaluate the chances of a neuron to receive chaotic noise. When a sufficient number of neighbors –which are indicated by the adjacency matrix - change their state and afterward a chaotic noise is injected to the current neuron to escape local minima. This is due to the fact that changing a route in a group of local cities may not need to change a global modification of all routs. In TSP, we have defined neighbors based on the ISODATA algorithm. All cities that are in the same cluster are neighbors and in the adjacency matrix will receive the value of "one".

To detect state changing of other neurons, we also define another matrix, C, which holds the state information of last timestamp.

$$C_{\tau} = \begin{pmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NN} \end{pmatrix} \quad (9)$$

When $c_{11} = 1$, indicates that neuron c_{11} in timestamp $\tau - 1$ has changed its state. Now we redefine Eq.8 as follows:

$$x_{ik}(t+1) = f\left(\sum_{j=1}^{N}\sum_{l=1}^{N} w_{ikjl}\, x_{jl}(t) + \theta_{ij} + \varepsilon + P(t)\right) \quad (10)$$

Where $P(t)$ is responsible to inject the proportionate chaotic noise. In order to generate the chaotic noise with respect to the adjacent neurons we define it as follows:

$$P(t) = k(t)x_{ik}(t) + \alpha I_i - z(t)\left[\frac{1}{1+e^{-\frac{x_{ik}(t)}{\varepsilon}}} - I_0\right] \quad (11)$$

$$z(t+1) = (1-\beta_1)z(t) \quad (12)$$

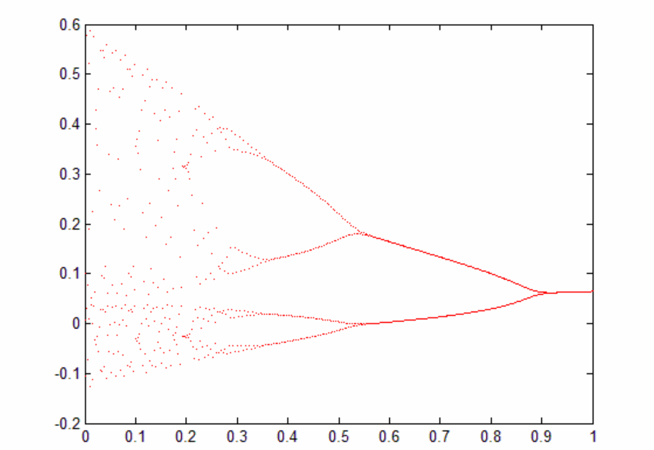
$z(t)$ is the self-feedback connection weight, β_1 is damping factor for neural self-coupling and I_0 is a positive parameter. We have assumed the following values for these parameters: $\alpha = 0.015, \beta_1 = 0.001, \varepsilon = 0.004, I_0 = 0.65$ to achieve the best results. $k(t)$, is the damping factor which is responsible to control the dynamic behavior of injected noise:

$$k(t) = v(t).\beta_2(t).g\left(\frac{\sum_j a_{ij}\, c_j}{\lambda}\right) \quad (13)$$

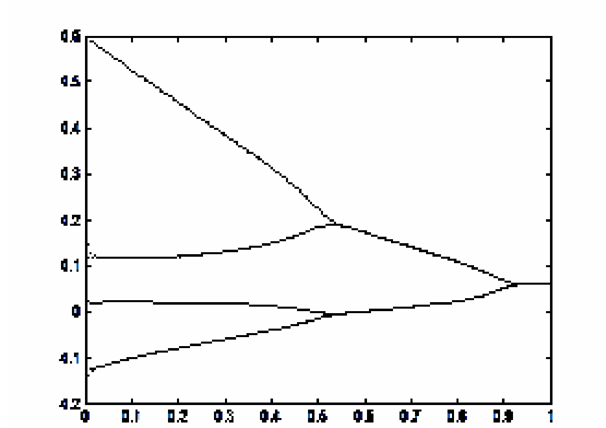
$$\beta_2(t+1) = (1-\gamma)\beta_2(t)$$

λ, is the maximum number of "ones" in rows of Matrix A. in this problem $g(.)$, has been tuned to output a value of 0.6 for an input value which is greater than $1/\lambda$. By other means, if only one of neighbors change its state, then the state of current neuron will be injected by chaotic noise. Indeed, behavior of this function may vary according to the scope of the problem. γ, which is the decaying parameter of $g(.)$, has been chosen to be 0.001 and finally $v(t)$, checks the function for previous chaotic history and if the neuron already receives the chaotic noise, it cancels reinitiating the parameters.

As it has been shown in Figure 1, by varying the parameter k, and for values greater than 4.3 we achieve the chaotic noise.



(a)



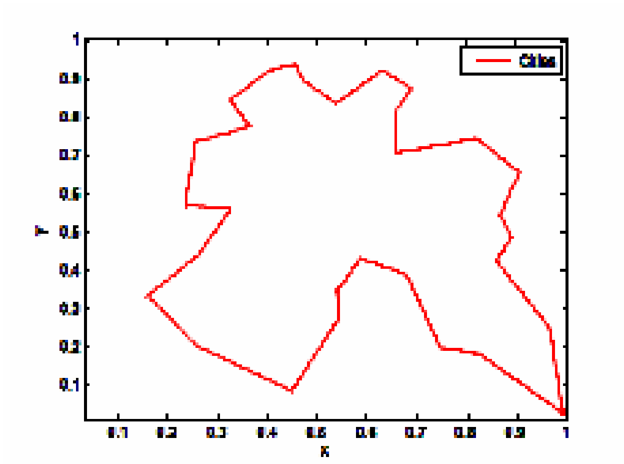
(b)

Figure 1: Chaotic and period doubling behaviors of Eq. for (a): k=4.3 and (b): k=4

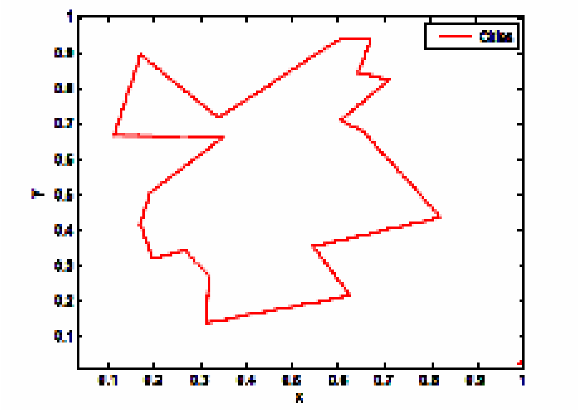
3 Experimental results

First, all values neurons have been initiated at random. In order to evaluate performance of the proposed algorithm, two sets of TSPs have been experimented. The first set which we call it "normal distribution of cities" consists of maps with nearly equivalent distances. In this set of maps, the ISODATA algorithm usually fails to cluster cities in different clusters. The adjacency matrices of these maps are similar to the matrices of N-Queen Problem and hence changing state of an individual neuron affects state of most of other neurons. However there is another set of maps in which cities are distributed in separate clusters. This kind of distribution usually occurs in vast areas where the ecological situation varies significantly from part to part of the map. Figure 2 illustrates samples of these two kinds of maps. The first three maps of this figure are samples of normally distributed cities while the forth one is an example of cities with separate clusters.

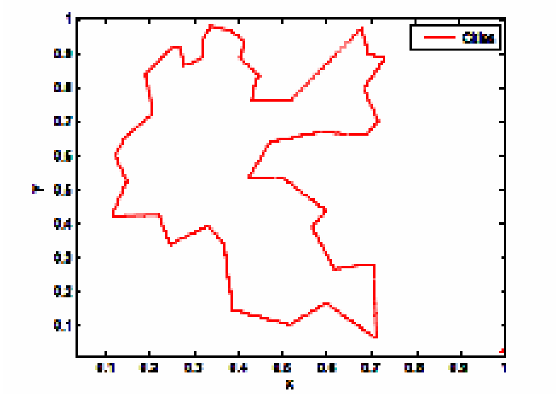
Although the of the proposed algorithm slightly outperforms the TCNN and NCNN models in the maps with normally distributed cities, however, the efficiency of the new algorithm is more obvious in the second kind of maps, separated clusters, where the rate of success for our algorithm is significantly better than TCNN and NCNN. Table 1 summarizes the experimental results for these three kinds of chaotic networks. The success rate is slightly better for samples from 19, 40 cities, whereas the NCNN algorithm is best of all for the 30-cities map. For the second kind of maps, the proposed algorithm is significantly better in all experiments.



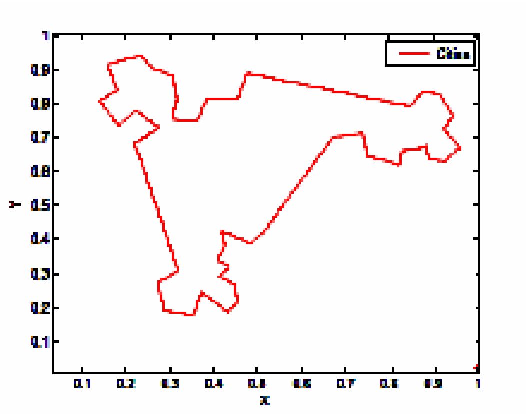
(a)



(b)



(c)



(d)

Figure 2: Maps with normally distributed cities (a,b and c) and map of cities with separated clusters(d).

Table 1: Rate of success for different models in normally distributed cities map

Number of cities		Rate of success (%)		
		TCNN	NCNN	Adjacency-based chaos injection
19		83	86	90
25		60	71	68
30		22	28	28
40		6	6	8

Table 2: Rate of success for different models in separated clusters map

Number of cities		Rate of success (%)		
		TCNN	NCNN	Adjacency-based chaos injection
20		74	72	84
30		29	32	43
45		4	4	11

4 Conclusions

In this paper we proposed a new chaotic noise injection strategy for TSP problem based on an adjacency matrix. The new noise injection strategy caused a better performance in most of maps used for experiment; however, the performance of the algorithm is more obvious for maps with separated clusters. This was due to the fact that instead of a blind noise injection strategy, neurons receive chaotic noise based on their neighbor's state and this is an advantage over all recently chaotic based driven networks for solving optimization problems. For future we plan to extend our work to a greater class of optimization problems called 0-1 optimization. From the nature of the algorithm it can be deduced that this kind of noise injection strategy can be applied to a wide class of optimization problems.

5 References

[1] J.J. Hopfield, “Neurons with Graded Response Have Collective Computational Properties like Those of Two-state Neurons”, *Pmc. Natl. Auld. Sci. USA,* 81, 3088/3092 ,1984.

[2] J.J. Hopfield, D.W. Tank. “Neural computation of decisions in optimization problems”, Biological Cybernetics, Vol 52, pp.141-152, 1985.

[3] K. Aihara, T. Takabe, and M. Toyoda, “Chaotic neural networks,” *Phys. Lett. A*, vol. 144, nos. 6, 7, pp. 333–340, 1990.

[4] T. Yamada, K. Aihara, and M. Kotani, “Chaotic neural networks and the travelling salesman problem,” in *Proc. Int. Joint Conf. Neural Networks*, 1993, pp. 1549–1552.

[5] H. Nozawa, “A neural-network model as a globally coupled map and applications based on chaos,” *Chaos*, vol. 2, no. 3, pp. 377–386, 1992.

[6] L. Chen and K. Aihara, “Chaotic simulated annealing by a neural network model with transient chaos,” *Neural Networks*, vol. 8, no. 6, pp. 915–930, 1995.

[7] L. Chen and K. Aihara, “Chaos and asymptotical stability in discrete time neural networks,” *Phys. D*, vol. 104, pp. 286–325, 1997.

[8] L. Wang and K. Smith, “On chaotic simulated annealing,” IEEE Trans. Neural Net., vol. 9, pp. 716–718, 1998.

[9] M. Hayakawa and Y. Sawada, “Effects of the Chaotic Noise on the Performance of a Neural Network Model”, *Technical report of IEICE,* NLP94(39), 47/52, 1994.

[10] T. Ueta, Y. Nishio and T. Kawabe, “Solving Abilities of the Hopfield Neural Networks with Burst Noise”, Proc. *NOLTA'97,* 409/412, 1997.

[11] D. Zhou, K. Yasuda, and R. Yokoyama, “A method to combine chaos and neural-network based on the fixed point theory,” *Trans. Inst. Elect. Eng. Japan*, vol. 117-C, no. 5, pp. 599–608, 1997.