Ineffectiveness in Solving Combinatorial Optimization Problems Using a Hopfield Network: A New Perspective from Aliasing Effect

Tai-Wen Yue and Li-Chen Fu
Department of Computer Science & Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

A Hopfield network has been proposed as a novel approach to achieve memory associativity and to solve combinatorial optimization problems. In the present paper we relate optimization problems to problems of memory association of a Hopfield network, and show the limitations of the network when it is used to solve NP-complete problems viewing from the aliasing effect among pattern sets (to be defined in this paper) and from the information capacity embedded in such a network. A simplest Hopfield network for solving the so called race traffic problem is constructed to manifest the similarity between memory association and optimization problem resolution as well as to discuss the stablility of convergence in synchronous and asynchronous operation mode. By transforming the TSP problem to memory association problem, we are able to show that the use of a Hopfield network for solving NP-complete problems is, in fact, overloaded.

1. Introduction

The Hopfield associative/optimization network [1,2] has found many applications including in memory association [6], in pattern recognition [7], and in solving linear programming problems [4] and combinatorial optimization problems [3,5]. A Hopfield neural network is a single-layer (time-iterative) feedback network, which consists of N binary-valued neurons linked to each other with symmetric weight ($t_{ij} = t_{ji}$). The basic behavior of each neuron is simply summing the weighted input signals and grade it by a hard limiting nonlinear function [6] to produce an output on the values +1 or -1. The output of each node is fed back to all other nodes via weights denoted t_{ij}. For synchronous operating mode all neurons update their outputs simultaneously but for asynchronous operating mode only one neuron is allowed to update its output during each iteration.

The system dynamics of a Hopfield network is easily seen through an analysis of its corresponding Liapunov energy function [2,3]. Due to fact that the output of each neuron is bounded, the energy corresponding to each possible state is also a finite value. For asychronous operating mode, via the update function of neurons described above the energy can be shown to be monotonically decreasing so that the system eventually will be stuck at a local minimum. But for synchronous operating mode, the energy is non-increasing only and, hence, the system may oscillate among some states with equal energy. This point will be clear in the later dicussion.

The Hopfield network is functioned as an associative memory according to the information storage algorithm [2], which is to locate the autoassociated patterns at local minima of a well defined energy function. During the association phase, the key pattern is presented as an initial state of the network and, then, the stored pattern which best matches the key will be autoassociated by following the energy declination to a local minimum of the energy function. Also, due to the non-increasing property of energy, many binary optimization problems can be solved by letting the networks (each output pattern of the nodes gives a solution of the problem) be constructed such that the costs of their solutions (may contain infeasible solutions) are in the same order of their corresponding energy levels (order preserving). Moreover, the local minima of the energy function correspond to feasible or optimum solutions in their solution space such that the solution appears after the network settles down.

Unfortunately, due to the information capacity [8] and some undesired effects, the desired function goal of a Hopfield network usually fails to be reached [5,9]. The following sections address some critical problems existing in a Hopfield network. In section 2, a network sample used for solving the so called race traffic problem is constructed, which is, then, used to relate the optimization problem to problem of memory association and to illustrate the unstable behavior of that network when running in synchronous mode. In section 3, the aliasing effect among pattern sets is defined, and is shown to find possible attraction points (local minima) in a Hopfield associative memory. In section 4, the traveling salesman problem (TSP) is transformed to a memory association problem, which indicates that the Hopfield network is, in fact, overloaded when it tries to solve NP-complete problems with a view to the aforementioned aliasing effect and information capacity.

2. Associative Memory vs. Optimization Problem

For a Hopfield network with N neurons, we let $T=[t_{ij}]_{N\times N}$ be a symmetric matrix which determines the connection strengths among neurons and let $V_i(t)$ (=-1 or +1) denote the i-th neuron's output at time t. The update function of i-th neuron is given as:

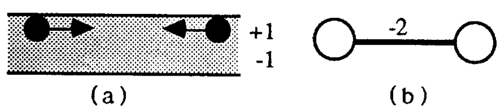
$$V_i(t+1) = f_h\left(\sum_{j=1}^{N} t_{ij} V_j(t)\right) \quad \text{if i–th neuron selected at time } t+1$$
$$= V_i(t) \quad \text{otherwise} \qquad (1)$$

where $f_h(.)$ is a hardlimit nonlinearity [6] and the Liapunov energy function of the overall network [2] is defined as:

$$E = -\frac{1}{2}\left(\sum_i \sum_j t_{ij} V_i V_j\right) \qquad (2)$$

It has been proved[2] that continuous application of the update function (1) will cause the Liapunov function E to be a monotonically decreasing function, i.e. $\Delta E \leq 0$, so that the network always converges to a local minimum of the energy after enough iterations. The following is a general procedure to map an optimization problem onto a Hopfield network:

1. Analyze the problem and define the cost function to be minimized.
2. Represent the problem in terms of a set of variables V_i = -1 or 1 (binary programming), i = 1,2,...,N.
3. Define an energy function $E(V_1,...,V_N)$ such that the desired solution occurs at a minimum of E.
4. Define the connection matrix T from the energy function E.



async. operation

t	V_1	V_2	E	ΔE
0	+1	+1	2	-
1	+1	-1	-2	-4

node 2 is selected for update

(c)

sync. operation

t	V_1	V_2	E	ΔE
0	+1	+1	2	-
1	-1	-1	2	0
2	+1	+1	2	0
⋮	⋮	⋮	⋮	⋮

(d)

Figure 1. Race Traffic Problem

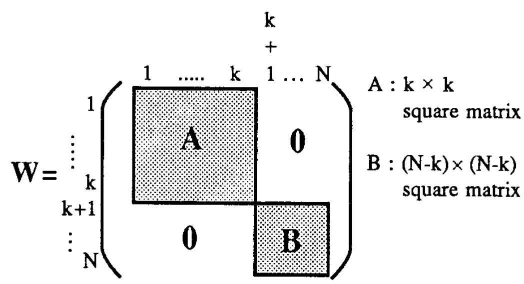


Figure 2. The matrix which cause aliasing

2.1 Race Traffic Problem

Suppose there is a narrow street whose width is only good for two persons side by side. Now there are two persons who want to pass this street in the opposite directions as shown in Figure 1a. The problem is to find a traffic condition such that they can pass the street successfully. To map this problem onto a Hopfield network, we let V_i = 1 (i = 1 or 2) denote the i-th person who occupies the up-traffic and V_i = -1 denote the i-th person occupying the down-traffic, and define the cost function or energy function (to be the same in this problem) as:

$$E = (V_1 + V_2)^2 = V_1^2 + V_2^2 + 2V_1V_2.$$

where the square terms in the above equation are constant values and, hence, can be discarded without influencing the optimization result. Thus, the new version of the energy function is defined as :

$$E = 2V_1V_2, \qquad (3)$$

which corresponds to a Hopfield network using two nodes and connections $t_{12} = t_{21} = -2$ as shown in Figure 1b via definition in (2). This is possibly the simplest network throughout discussions on a Hopfield network in the literature. Becuause of its simplicity, some generic problems are more easily to surface. Figure 1c and 1d show the snapshots of this network running in an asynchronous and a synchronous mode respectively where the initial state is set to (+1,+1), i.e. these two persons both occupy the up-traffic initially. For asynchronous operation mode, only one iteration is needed to cause the network to be stuck at a local minimum (in this case also global minimum) but for synchronous operation mode there is no way to lead the network to converge to a stable state. Thus we conclude that for a discrete Hopfield model the network have to run in an asynchronous mode to ensure the network to reach a stable state eventually. In the following discussions on the neuron's activity are given assuming asynchronous operation mode unless otherwise stated.

2.2 Associative Memory Approach to Solve Race Traffic Problem

Using the Hopfield model as an associative memory to store M N-dimensional patterns, the network can be constructed by N neurons with connection matrix T [2] defined as:

$$T=\sum_{s=1}^{M} {}^{s}x\,{}^{s}x^{T} - M\,I \qquad \text{where } {}^{s}x = ({}^{s}x_1,\ldots,{}^{s}x_N)^T,\ {}^{s}x_i \in \{+1,-1\}. \qquad (4)$$

In last subsection, we know persons must occupy different traffic line in order to pass the street and, hence, define an energy/cost function to fit in a Hopfield network, i.e. we treat the race traffic problem as an optimization problem. Here, we will solve the same problem by using the associative memory approach. We know there are only two kinds of traffic patterns, (+1,-1) and (-1,+1), that can be chosen for successful passing. By exhaustively finding all feasible patterns, the race traffic problem can also be thought of as a memory association problem. Therefore, using S = {(+1,-1) , (-1,+1)} as a stored pattern set, the resulting connection matrix T will be :

$$T=\begin{pmatrix} 0 & -2 \\ -2 & 0 \end{pmatrix}$$

The network constructed using the above connection matrix is the same as the one we have done in last subsection. Consequently, when we define an energy/cost function for solving an optimization problem using a Hopfield network, what we have really done is just using different methods to "tell" the network all the possible solution patterns while the network's behavior is the same as the memory association process. Hence, knowing what limitations may exist when a Hopfield model is used as an associative memory will help us to know what limitations also exist when it is used for solving optimization problems.

3. Aliasing Effect among Pattern Sets and Information Capacity of a Hopfield Model

Using information storage algorithm (4) to construct a Hopfield associative memory, we hope all the local minima of the energy function are corresponding to the stored patterns such that a valid associated pattern can always be obtained when the network settles down. Unfortunately, aliasing effect among pattern sets to be defined below shows that many undesired patterns are also stored in the network according to the rule in (4) and, hence, causes the deterioration of the associability of a Hopfield associative memory.

Definition 1. (Aliasing (≅) Effect among Pattern Sets)

Two pattern sets

$S=\{{}^{i}x \mid {}^{i}x = ({}^{i}x_1,\ldots,{}^{i}x_N)^T,\ 1 \le i \le M\}$ and $S'=\{{}^{i}y \mid {}^{i}y = ({}^{i}y_1,\ldots,{}^{i}y_N)^T,\ 1 \le i \le M\}$

are said to be aliasing (denote as $S \cong S'$) if they induce the same connection matrices i.e. T = T', by applying information storage algorithm (3) over pattern sets S and S'.

Indeed, if two different pattern sets are aliasing, to store the information for one pattern set to the network enables the other pattern to be stored as well because there is no way for us to tell which pattern set is really stored by viewing the connection matrix only. Obviously, if too many pattern sets which are different from S but are aliasing with it, its associablility will be considerably degraded. It can be easily seen from (4) that the pattern set which is obtained by reversing a collection S of some patterns in binary sense will be aliasing with S, i.e. when to store the information of a pattern set to the network its reversed counterpart is also stored. Furthermore, when the size of stored pattern set gets larger the aliasing effect becomes more serious, which can be seen in the following discussion.

When given a key to autoassociate with one of the stored patterns, after the system converges the associated pattern will appear at outputs of neurons. If the associated pattern is different from the original stored one by a few sign bits, we may suspect whether this wrong associated pattern is also stored in the network, i.e. the stored pattern set may be aliasing with some pattern set which includes this wrong pattern. Now, we will describe how to find the aliasing pattern set(s) systematically as follows.

Suppose we are given a pattern set $S=\{{}^{i}x \mid {}^{i}x = ({}^{i}x_1,\ldots,{}^{i}x_N)^T,\ 1 \le i \le M\}$. Without loss of generality, assume a pattern set S' which is aliasing with S can be obtained by reversing the first k sign bits of the first M' patterns in S, i.e.

$S'=\{{}^{i}x' \mid {}^{i}x' = (-{}^{i}x_1,\ldots,-{}^{i}x_k,\ {}^{i}x_{k+1},\ldots,{}^{i}x'_N)^T$ if $i \le M'$ else ${}^{i}x' = {}^{i}x$,

where $M' \le M$, $k \le N$ and $1 \le i \le M\}$.

Because S and S' are aliasing, their connection matrices T and T' must be equal, i.e.

$$T = [t_{ij}]_{N\times N},\ T'=[t'_{ij}]_{N\times N},\ \text{and}\ t_{ij} = t'_{ij} \qquad \forall\ 1 \le i,j \le N.$$

Because only M' patterns are different between S and S', to attain the above equlity the following condition must hold:

$$\text{If } W = [w_{ij}] = \sum_{s=1}^{M'} {}^{s}x\,({}^{s}x)^T \text{ and } W' = [w'_{ij}] = \sum_{s=1}^{M'} {}^{s}x'\,({}^{s}x')^T \text{ then } W=W'. \qquad (5)$$

Now suppose that the matrix has the shape as shown in Figure 2 (after some possible permutations), then with the following equalities the condition in (5) is trivially satisfied.

$$w_{ij} = \sum_{s=1}^{M'} {}^{s}x_i\,{}^{s}x_j = \sum_{s=1}^{M'} (-{}^{s}x_i)(-{}^{s}x_j) = \sum_{s=1}^{M'} {}^{s}x'_i\,{}^{s}x'_j = w'_{ij} \qquad i \le k, j \le k,$$

$$w_{ij} = \sum_{s=1}^{M'} {}^{s}x_i\,{}^{s}x_j = \sum_{s=1}^{M'} {}^{s}x'_i\,{}^{s}x'_j = w'_{ij} \qquad i > k, j > k,$$

$$w_{ij} = \sum_{s=1}^{M'} {}^{s}x_i\,{}^{s}x_j = 0 = \sum_{s=1}^{M'} (-{}^{s}x_i)\,{}^{s}x_j = \sum_{s=1}^{M'} {}^{s}x'_i\,{}^{s}x'_j = w'_{ij} \qquad i \le k, j > k,$$

$$w_{ij} = \sum_{s=1}^{M'} {}^{s}x_i\,{}^{s}x_j = 0 = \sum_{s=1}^{M'} {}^{s}x_i\,(-{}^{s}x_j) = \sum_{s=1}^{M'} {}^{s}x'_i\,{}^{s}x'_j = w'_{ij} \qquad i > k, j \le k.$$

Hence, when using a Hopfield network as an associative memory, the larger the pattern size is the more likely the network will produce considerable amount of aliasing pattern sets so that the associablility will be seriously lost. Such a condition has also appeared after an analysis on the information capacity of a Hopfield associative memory [8], where the maximum number of the stored patterns is of order N/log N with N being the number of neurons in the network. For illustration, Figure 3 shows some pattern sets aliasing with a given one. It is note worthy that the the aliasing effect among pattern set is under investigation without any resort to stochastic hypothesis, the numbers of the undesired aliasing patterns (patterns belong to some aliasing pattern sets but not in S), however, is not only related to the size of the stored pattern set but also depends on the relationship among the stored patterns. Furthermore, the aliasing relation ($\cong$) defined in previous definition is, in fact, an equivalent relation which has reflexive, symmetric and transitive properties and, hence, for any given fixed pattern set, algorithms can be designed to find all its aliasing patterns.

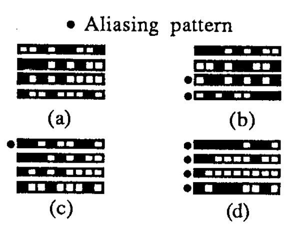


Figure 3. (a). Given pattern set (black cell = +1 white cell =-1). (b-d). Pattern sets aliasing with (a).

	1	2	3	4	5	6	7
A	-1	+1	-1	-1	-1	-1	-1
B	-1	-1	-1	-1	+1	-1	-1
C	-1	-1	+1	-1	-1	-1	-1
D	-1	-1	-1	+1	-1	-1	-1
E	+1	-1	-1	-1	-1	-1	-1
F	-1	-1	-1	-1	-1	-1	+1
G	-1	-1	-1	-1	-1	+1	-1

Figure 4. An example of 7-city TSP.

4. NP-Complet Problem vs. Pattern Set Aliasing

The traveling salesman problem (TSP) is proved to be an NP-complte problem. It is widely used as a benchmark case for testing the effectiveness of the newly developed computing techniques for NP-complete problems. Given N cities, the problem is to find the shortest path through the cities such that the salesman will visit every city only once and then return to the starting city. The cost function is the length of the tour. In this section we will relate TSP problem to a memory association problem and show the ineffectiveness of using a Hopfield model to solve it due to the aliasing effect among pattern sets.

4.1 Map the TSP Problem into Hopfield Network

The work by Hopfield and Tank [3] illustrated how the TSP could be mapped into a neural network using an analog circuit configuration. Each neuron's activity in the original model is governed by a differential equation but in this section we will reproduce a discrete version of that model to view the aliasing effect among pattern sets in solving combinatorial optimization problems. Indeed, this can be thought of as if we replaced each neuron with a high gain amplifier [2]. For a N-city TSP, assign an identifier (A,B,C,...) to each city in arbitrary order. The solution to the TSP can be represented as an N×N permutation matrix as shown in Firgure 4 for a 7-city problem. Each row of this matrix represents a particular city and each column represent a particular visiting sequence in the tour. Here the row index of a "+1" entry is the city name and its column index is the visiting sequence. Therefore, Figure 4 represents the tour EACDBFGF.

When applying the Hopfield model to TSP, each entry in the permutation matrix is replaced by a neuron. By properly choosing connections and assigning weights, this network will reach a final state and represent a TSP solution through the interpretation defined above. The key in this process is to transform the TSP solution requirements into the neural net energy function and to construct the connection matrix from the energy function. The solution requirements of TSP consist of two parts:

(1) Ensure that the final matrix is a legal TSP solution, i.e. only one neuron is "on" (+1 in permutation matrix) in each column and in each row, and the total number of "on" neurons should not exceed the number of cities.

(2) Ensure that the final matrix favors the shortest tour.

These two requirements can be specified as the energy functions E_1 and E_2 defined below:

$$E_1 = \frac{\alpha}{2}\sum_X\sum_i\sum_{j\neq i}\left(\frac{V_{Xi}+1}{2}\right)\left(\frac{V_{Xj}+1}{2}\right) \quad \text{/* minimum when 0/1 neuron "on" each row*/}$$

$$+\ \frac{\beta}{2}\sum_i\sum_X\sum_{X\neq Y}\left(\frac{V_{Xi}+1}{2}\right)\left(\frac{V_{Yi}+1}{2}\right) \quad \text{/* minimum when 0/1 neuron "on" each column*/}$$

$$+\ \frac{\gamma}{2}\left\{\sum_X\sum_i\left(\frac{V_{Xi}+1}{2}\right)-N\right\}^2 \quad \text{/* minimum when exactly N out of } N^2 \text{ neuron "on"*/} \qquad (6)$$

$$E_2 = \frac{\delta}{2}\sum_X\sum_{Y\neq X}\sum_i d_{XY}\left(\frac{V_{Xi}+1}{2}\right)\left(\frac{V_{Y,i+1}+V_{Y,i-1}}{2}+1\right) \quad \text{/*the total cost of selected tour*/} \qquad (7)$$

where d_{XY} in E_2 is the distance between city X and Y. Thus, the total energy function for TSP defined as :

$$E = E_1 + E_2$$

By properly choosing the parameters $\alpha, \beta, \gamma, \delta$ the global minimum of E will corresponding to the TSP optimum solution and all the feasible solutions will be located at the local minima of E.

In a wide scope of experiment, Wilson and Pawley [9] applied the Hopfield model to a 10-city TSP. The result is that 92% of all trials failed to give a legal tour and only 8% of the trials converged to a legal solution which is barely better than the randomly chosen tours. So far, there has not been much work on explaining what may cause this excessively deteriorated performance of the network.

In Section 3, we know for an associative memory, the more information patterns are stored in the network the more aliasing patterns will be produced so that the associability declined extremely fast. And in Section 2, we illustrate that the role of an energy function for an optimization problem can be thought of as an effective method to "tell" the network possible solution patterns. Now, referring back, especially, to the term E_1 given by eq. (6) in the energy function of TSP, its role is obviously to lead the network to settle down to some feasible state (feasible solution) only. Hence, the term E_1 provides the information needed for feasible solutions, i.e. requirement (1) discussed above. On the other hand, E_1 also contains the information regarding what the feasible solutions of the network may be, i.e. to use all the feasible solutions as stored patterns and the information storage algorithm to construct the connection matrix. Indeed, it can be easily proved that these two approach construct the same matrix (scale one matrix produce the other). In the bove discussion, we have temporarily dropped the data about distance among cities so that the problem becomes a special TSP where all the city distances are set equal, abbreviated as EQTSP. The problem with different city distance will be reconsidered later in this section. Of course, the constraints in EQTSP is much less than those in TSP. The only constraint here is only to find a tour through every city exactly once and return to the starting city. A question should be raised here is that should the Hopfield model work better in such a case? But by using aliasing analysis in Section 3 it seems that the network contains too much information. In words, there are N! possible solutions in EQTSP and, hence, N! patterns are stored in a network with only N^2 neurons, which cause the network to be overload . In Figure 5, we demonstrate the aliasing effect among pattern set for 4-city EQTSP, where each template in that figure corresponds to a possible (aliasing) solution set that results in the same connection matrix T, and each row of that templates corresponds to an elongated permutation matrix. Figure 5a is the feasible solution patterns that is originally used to construct the connection matrix but Figure 5b and 5c indicate the other two quite different and illegal pattern sets which are aliasing with the legal one. Indeed, we can also construct very many strange templates that are aliasing with the original one. This somehow suggests that the unsatisfactory performance of network is due to the overloading from excessive amount of information.

Now, reconsider the original TSP where the energy function actually consists of E_1 and E_2. Since the solution requirement (1) given previously has to be fulfilled before the second one is brought for consideration, more weighting is usually laid on term E_1 and the less on E_2 so that a legal solution can be generated after the network settles down. When that is the case, locations of the states with local minimal energy (energy well) due to E_1 solely will be almost unchanged when $E_1 + E_2$ is considered instead. The only change which may occur is in the depth of the energy well. Consequently, the aliasing problem among pattern sets still exists in this TSP and, hence, challenge the capability of a Hopfield model in solving that kind of problems. Unless, we can find a more effective energy function to prune large amount but poor solutions, use of a Hopfield network to solve optimization problems should be ineffective.

5. Conclusion

This paper addresses several issues related to the applications of a Hopfield model. Using it as an associative memory, the concept discussed in Section 3 can be used for a deterministic code analysis to discover all possible aliasing patterns such that the usability of this network can be estimated. Moreover, using it for solving (combinatorial) optimization problem, the size of (feasible) solution space embedded in the network should not exceed the capacity of the network, otherwise the performance of the network will be by far behind our expectation.

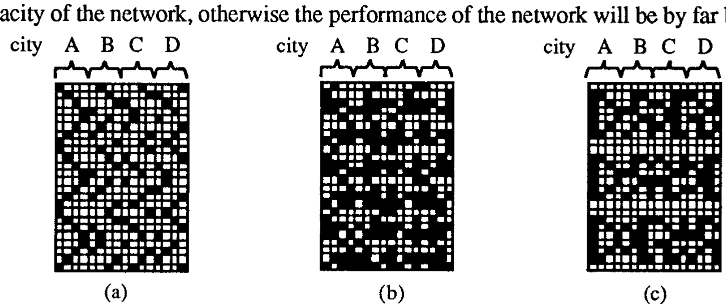


Figure 5. Aliasing effect among pattern sets for 4-city EQTSP (a) Feasible solution patterns (b)(c) Infeasible solution sets due to the aliasing effect among pattern sets.

REFERENCES

[1] J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Natl. Acad. Sci. USA*, Vol. 79, 2554-2558, April 1982.

[2] J.J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," *Proc. Natl. Acad. Sci. USA*, Vol. 81, 3088-3092,May 1984.

[3] J.J. Hopfield, and D.W. Tank, "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*. 52,(1985), pp. 141-152.

[4] D.W. Tank, and J.J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programing Circuit," *IEEE Tran. on Circuits and Systems*, Vol. CAS-33, No. 5, May 1986.

[5] B. Shirazi, and S. Yih, "Critical Analysis of Applying Hopfield Neural Net Model to Optimization Problems," *Proc. of IEEE Intl. Conf. on SMC*, Cambridge, MA, pp. 210-215, 1989.

[6] R.P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magine*, April 1987, pp. 4-22.

[7] N.H. Farhat, D. Psaltis, A. Prata, and E. Peak, " Optical implementation of the Hopfield Model," *Apply Optical*, Vol. 24, pp. 1469-1475, May 1985.

[8] R.J. McELIECE, E.C. EDWARD, E.R. RODEMICH, and S.S. VENKATESH, "The Capacity of the Hopfield Associative Memory," *IEEE Trans. Inform. Theory*, 1987, IT-33, (4), pp. 461-482.

[9] Wilson, G.V. and G.S. Pawley, "On the stability os the Traveling Salesman Algorithm of Hopfield and Tank," *Biological Cybernetics*, 58, (1988), pp. 63-70.