A Neural Algorithm To Solve The Hamiltonian Cycle Problem

Shashank Mehta and Laszlo Fulop
ELECTRICAL COMPUTER AND SYSTEMS ENGINEERING DEPARTMENT
RENSSELAER POLYTECHNIC INSTITUTE
TROY, NY 12180-3590

ABSTRACT

A network of analog neurons to solve the Hamiltonian Cycle Problem (HCP) is described in this paper. This neural net is a modification of the network proposed by Hopfield to solve the TSP. A result on convergence of quasi-stationary flow and a bound for the strength of a inhibitory self-connection is presented. Results are reported of successful experiments with graphs of up to 500 nodes. The result of the experiment with the well known 318-city TSP is also reported.

1. INTRODUCTION

A significant attention is drawn to the application of neural networks to optimization problems since the Hopfield and Tank's paper [6] on the TSP was published.

The neurons in the original model had two states only (0 and 1). A couple of years after their original paper [4], Hopfield extended his ideas from binary neurons to those with analog output [5]. This transition was primarily motivated by his desire to implement the model using real-world hardware components such as op-amps and resistors.

Nearly three years after Hopfield and Tank's paper describing the neural net solution for the TSP, Wilson and Pawley [12] reported their inability to reproduce Hopfield's original results and shed a very unfavorable light on the usefulness of optimizing neural nets.

Wilson and Pawley's paper seemed to act as a catalyst to ignite the interest of other researchers in this area. A number of research groups also tried to copy Hopfield's experiments and generally found the same result as that of Wilson and Pawley. Simple modifications were proposed to Hopfield's energy function in [1] [10], while others experimented to find better ways to determine the very illusive coefficients that make up the connection weights [2] [3] [9]. All the above cited works have three major points in common.

First, they typically report results for TSPs with number of cities in the 10-30 range and the results generally degraded with increasing problem sizes. Secondly, in all the results reported, there were problems in consistently arriving at even a valid tour (Hamiltonian cycle), be it optimum or not. Thirdly, all these papers have depended on trial and error experimentation to determine the network coefficients as was the case in the Hopfield's paper.

A somewhat more novel approach has been presented in [11]. A solution to the TSP is described which combines features of neural networks and simulated annealing. The resulting 'neural net' (loosely termed) finds valid tours consistently (for the 30 city case reported), but gives somewhat longer (costlier) path than those given by pure simulated annealing [7]. It is also widely believed, that simulated annealing solutions are much slower than pure neural net solutions.

In this paper we present modifications in the Hopfield's approach by (i) converging the net quasi-stationary manner, (ii) using inhibitory self-connections to neurons, (iii) giving a method to determine the values of the energy coefficients, and (iv) redefining the energy function to be minimized to solve the problem. Solutions of the HCP on up to 500 node graphs and the well known 318-city [8] TSP are reported.

Problem Definition. The Hamiltonian Cycle Problem is to determine a simple cycle passing through each node of the given graph, G, which may not be fully connected.

In the design of the network we shall use an $n \times n$ matrix of neurons as in [6]. A zero-one matrix will be called *legal* if it has one 1 in each row and column. A legal matrix will be called *valid* if it corresponds to a Hamiltonian cycle in the given graph.

The HCP has only two types of legal cycles, valid and invalid, unlike TSP where a degree of validity can be associated with each Hamiltonian cycle because G is assumed to be fully connected. The experiments were conducted, primarily, with the HCP (rather than with the general TSP) because it is easy to judge the quality of its results.

2. Quasi Stationary Convergence

In this section a method of controlling the system flow through the state space is presented which uses the neuron gain factor for that purpose.

Hopfield [5] studied analog-neuron networks as a surrogate for discrete-neuron networks to solve optimization problems because it is possible to build analog devices which are cheaper than digital devices. A sketchy proof is also presented in [5] for the convergence of such (analog) networks with no self-connection, but a condition of quasi-stationarity, necessary for the validity of the proof, is not acknowledged.

An analog neuron is one which can output any real number in the interval [0,1]. Therefore, the state of an N-neuron analog system can flow through the volume of the N-dimensional unit cube (state space) and not remain restricted to the surface (corners) as does the discrete system. The final state of a discrete system is at the mercy of the initial state, since it will settle down at the first encountered minimal point. That is, if a path to a minimal point p_1 passes through another minimal point p_2 then, p_1 can not be reached through that path. Although the initial state plays a very important role, Hopfield-Tank method selects it randomly. To overcome this deficiency of the discrete network we shall consider an analog network, which could directly converge to a solution point on the surface starting from an inner point of the cube.

To begin with, let us define a family of transfer functions in which the step-function (discrete-neuron) is a member,

$$g_\lambda(A) = g(\lambda A), \quad \lambda \geq 0$$

where g is the sigmoidal or linear function. The function g_λ approaches the step-function as $\lambda \to \infty$. On the other extreme, $g_0(A)$ is equal to 1/2. This is an interesting function, because under it the system can only be at one state, namely, the center $(\langle 1/2, 1/2, \ldots, 1/2\rangle)$ of the cube. In other words, the feasible region of the system shrinks to the point $\langle 1/2, 1/2, \ldots, 1/2\rangle$ at $\lambda = 0$. Starting from $\lambda = 0$, suppose λ is increased to some small value λ_1. Since there are only a finite number of inputs to each neuron, the activities of the neurons are bounded, say, between $-B_{\lambda_1}$ and $+B_{\lambda_1}$. Therefore, the outputs of the neurons are bounded between $g(-\lambda_1 B_{\lambda_1})$ and $g(\lambda_1 B_{\lambda_1})$. These bounds restrict the feasible region for the system state. The feasible region R_λ changes continuously with λ for a sigmoidal-neuron network, because $dg_\lambda/d\lambda$ exists. The same is true for linear-neuron nets except when the boundary approaches that of the cube, because the linear function is not differentiable near the bounds. The feasible region for a larger λ includes that for a smaller λ, i.e., $R_{\lambda_1} \subset R_{\lambda_2}$ for $\lambda_1 \leq \lambda_2$. For the two extreme values of λ the feasible regions are $R_0 = \langle 1/2, \ldots, 1/2\rangle$ and $R_\infty =$ entire cube.

If the energy function is continuous and differentiable (i.e., $\partial E/\partial V_a \, \forall a$ exists), then every minimal state must be a low point of a valley of the energy terrain. In other words, the minimal points are not isolated and scattered craters, but approachable by traveling through a valley. Furthermore if $\partial^2 E/\partial V_a^2 > 0 \, \forall a$, then every valley has one minimal point.

Suppose p' is a minimal point of the surface, and λ' is large enough so that $R_{\lambda'}$ contains p'. Also, suppose λ is slowly decreased until R_λ shrinks just past p'. Let the new value of λ be λ''. Since the valley of p' is a continuous curve, there must be a new point p'' near p' on the valley which is minimal w.r.t. $R_{\lambda''}$. If the process is continued, then the locus of these intermediate minimal points will terminate at the center of the cube as $\lambda \to 0$. If $\partial^2 E/\partial V_a^2 > 0 \, \forall a$ then p'' will be infinitesimally close to p' and the locus will be a continuous curve. It will be referred to as the *quasi-stationary path of* p'. The quasi-stationary paths of all the minimal points form a tree rooted at the center.

It is now natural to ask what happens to a system, which tends to stay at a minimal energy state with respect to R_λ, if λ is gradually increased from 0. Intuitively, it seems that the system will travel outwards along one of these quasi-stationary paths to the corresponding stable state. This, indeed, is the case as will be established later in this section. This approach to a stable state seems more direct and fast compared to the surface restricted approach in the case of discrete nets. Also, as discussed in the beginning of this section, it has better odds of reaching an absolute minimum point compared to the surface restricted flow.

Furthermore, condition $\partial^2 E/\partial V_a^2 > 0 \forall a$ ensures that there is only one minimal point in any path from the center to the surface of the cube which has $dpath - length/dV_a > 0$.

Now we shall look into the changes in the energy of an analog system as it evolves. It will be shown that if the system flows in a quasi-stationary fashion (i.e., each firing results in a small change in the output), then the energy will decrease monotonically. One direct consequence of this result is that if the initial state is already very close to a minimal state of R_λ, then the system is sure to converge to it and stabilize. At the first glance this does not sound too significant. To understand its importance let us recall the quasi-stationary paths discussed earlier. Suppose λ_{max} is large enough to include a minimal point p_{max} of the cube in $R_{\lambda_{max}}$. Suppose we break up the interval $[0, \lambda_{max}]$ as $\lambda_0 = 0, \lambda_1, \ldots, \lambda_m = \lambda_{max}$. Also, suppose $p_0(=$ center$), p_1, \ldots, p_m(= p_{max})$ denote the intermediate minimal states of the quasi-stationary path of p_{max} in the regions $R_0, R_{\lambda_1}, \ldots, R_{\lambda_m}$ respectively. Traveling along the path can be accomplished by traveling from p_i to p_{i+1} keeping λ at λ_{i+1}, for $i = 0, 1, \ldots, m-1$. If p_i is close enough to p_{i+1}, then from the above assertion we are assured to arrive at p_{i+1}. The separation between p_i and p_{i+1} depends upon $\lambda_{i+1} - \lambda_i$. Therefore, by keeping the interval between the successive values of λ small enough, we can guarantee the convergence to p_{max}. The result also implies, as in the discrete case, that a state is minimal if and only if it is stable. Let us now present the convergence result formally. In the statement of the theorem, $w_{a,b}$ is the weight of the connection from neuron a to neuron b and $w_{ES,b}$ is the weight of the connection from the constant source (with output 1) to the neuron b.

THEOREM. *A quasi-stationary flow (i.e. the system evolves with $\lambda = 0$ to λ_{max}. At $\lambda = \lambda_1$ system is first allowed to arrive at a stationary state then λ is incremented to $\lambda_1 + \delta$ where δ is 'infinitesimally' small.) always converges to a stationary state if (i) $w_{a,b} = w_{b,a} \forall ab$, and (ii) $-w_{a,a} < {g^{-1}}'_{min} \quad \forall a$, where ${g^{-1}}'_{min}$ is the minimum value of the derivative of g^{-1}.*

This result can be established using the energy function originally proposed by Hopfield:

$$E = -\frac{1}{2} \cdot \sum_{ab} w_{a,b} V_a V_b - \sum_b w_{ES,b} V_b + \sum_a \int_0^{V_a} g^{-1}(u)\, du. \tag{1}$$

This section can be summed up as follows. Releasing λ slowly from 0, a quasi-stationary flow can be created. This, in turn, assures that the system will converge to a minimal point and stabilize if the conditions of symmetry and self-connection of the theorem are met. It is, of course, not yet obvious whether valid states are minimal points and what is the chance of converging to them if they are minimal points. This is the issue of the next two sections.

SELECTION OF CONNECTION WEIGHTS

Our goal in this section is to determine a set of connection weights such that the quasi-stationary process, discussed in the previous section, may successfully lead to a valid matrix. It is easy to see that if the system is allowed to evolve to a large value of λ, then the state will approach a corner of the cube because the transfer function of an analog neuron (sigmoidal or linear) approaches the step-function for a large value of λ. It remains possible that the system may end up in an illegal state or a legal but invalid state. In any case, the energy in the final state will be

$$E_{final} = -\frac{1}{2} \sum_{ab} w_{a,b} V_a V_b - \sum_b w_{ES,b} V_b, \tag{2}$$

because the last term in (1) is equal to zero for all the corner states.

Let us consider the energy function for the problem $E_{problem} = \frac{H}{2} \sum_{x,i \neq j} V_{xi} V_{xj} + \frac{V}{2} \sum_{x \neq y,i} V_{xi} V_{yi} + \frac{G}{1}(n - \sum_{xi} V_{xi}) + \frac{D}{2} \cdot \sum_{xyi} d_{xy} V_{xi}(V_{y,i+1} + V_{y,i-1})$, where d_{xy} is 0 if vertices x and y are adjacent and 1 otherwise. The first two terms penalize more than one 1 in any row or column. Since these terms, implicitly, have already taken care of the case of more than n 1's, the third term only penalizes the configurations with less than n 1's. The last term is the penalty for the selection of incompatible vertices in the successive positions. The constant term $G.n$ does not play any useful role so it can be removed, giving $E_{problem} = \frac{H}{2} \sum_{x,i \neq j} V_{xi} V_{xj} + \frac{V}{2} \sum_{x \neq y,i} V_{xi} V_{yi} - \frac{G}{1} \sum_{xi} V_{xi} + \frac{D}{2} \sum_{xyi} d_{xy} . V_{xi}(V_{y,i+1} + V_{y,i-1})$. Since the G-term is

negative and the other terms are positive, every additional 1 will be favored if G is large compared to the other coefficients. Similarly the system will tend to have fewer 1's when G is relatively smaller than H, V, and D. To achieve a balance between the two situations we introduce a self-connection term, giving the object function

$$E_{problem} = \frac{H}{2}\sum_{x,i\neq j} V_{xi}V_{xj} + \frac{V}{2}\sum_{x\neq y,i} V_{xi}V_{yi} - G\sum_{xi} V_{xi} + \frac{D}{2}\sum_{xyi} d_{xy}V_{xi}(V_{y,i+1}+V_{y,i-1}) + \frac{S}{2}\sum_{xi} V_{xi}V_{xi}. \quad (3)$$

Before we explain the role of the self-connections let us determine the connection weights. Since the expression (3) is to be minimized and in the ideal case (2) is the minimum expression, comparison of the two expressions gives $w_{xi,yj} = -H\delta_{xy}(1-\delta_{ij}) - V\delta_{ij}(1-\delta_{xy}) - S\delta_{xy}\delta_{ij} - Dd_{xy}(\delta_{i,j+1}+\delta_{i,j-1})$ and $w_{ES,yj} = G$.

The connection weights are symmetric but the self-connection condition for convergence is yet to be satisfied. Furthermore, we also need to ensure that the valid states, which are the ultimate goal, be among the minimal points of the energy surface. By choosing suitable values for the coefficients we will show, in the next section, that both of these requirements can be met.

Comparison with the connection weights in the Hopfield net shows that now there is no global inhibition. In the design of a VLSI chip it is advantageous to have lesser and more localized connectivity. In a simulation, computation of the activity of a neuron requires scanning of $4n-1$ inputs. Therefore, the time complexity of one cycle of firing is $\Theta(n^3)$ and the total time of the process is $\Theta(n^3 \times$ No. of cycles$)$.

Self-connection plays an important role in controlling the flow of the system at all the stages. In the process of approaching a desirable low energy state, neurons should constantly adjust their values in order to resolve incompatibility with their neighbors. However, a grown (large output) neuron tends to dominate over weaker neighbors depriving them of a fair chance. This is where inhibitory self-connection comes to play a role. While having insignificant effect on a small output neuron, it becomes a strong force against a grown neuron. As a result, weak surrounding neurons can put an effective force against the grown neurons to get a fair settlement. The inhibitory self connection also gives $\partial^2 E/\partial V_a^2 = S > 0 \forall a$, required for a smooth downhill surface from the center to the boundary of the cube.

Coefficient Values

One major difficulty with Hopfield formulation was that there were no guidelines to determine suitable values for the coefficients. In this formulation, on the other hand, we shall derive relations between the coefficients so that (i) convergence condition on the self-connections be satisfied, (ii) the valid states become the minimal points on the surface, and (iii) there be significantly fewer unwanted minimal points. The last criterion will help increase the probability of reaching the valid states.

The output of an analog neuron is given by $g_\lambda(A_{yj}) = g(\lambda A_{yj}) = g(\lambda\sum(w_{xi,yj}V_{xi}) + \lambda w_{ES,yj})$, therefore the absolute values of the coefficients are not important as λ plays the role of the scaling factor. Thus, we will only need to determine their ratios.

The role of H and V is to penalize multiple 1's in a row and in a column respectively. A matrix is legal if and only if its transpose is legal. As the two coefficients play similar roles, it is reasonable to require

$$H = V. \quad (4)$$

To make a valid matrix stable, we must require that any increase in the output of a neuron at 0 or any decrease in the output of a neuron at 1 must increase the energy, i.e., $\left.\frac{\partial E}{\partial V_{xi}}\right|_{V_{xi}=1,validstate} \leq 0$ and $\left.\frac{\partial E}{\partial V_{xi}}\right|_{V_{xi}=0,validstate} \geq 0$. Since $\frac{\partial E}{\partial V_{xi}} = H.\sum_{i\neq j} V_{xj} + V.\sum_{x\neq y} V_{yi} - G + D.\sum_y d_{xy}.(V_{y,i+1} + V_{y,i-1}) + S.V_{xi}$, the above conditions transform into the following inequalities on the coefficients

$$-G + S \leq 0 \quad (5)$$

and

$$-G + H + V + D\sum_y d_{xy}.(V_{y,i+1}+V_{y,i-1}) \geq -G + H + V = -G + 2H \geq 0. \quad (6)$$

The *D-factor* in (6), which depends upon the graph, has been replaced by its worst case (minimum) value, zero.

Conditions (5) and (6) ensure that every valid state is a minimal point of the energy surface. It is impossible to satisfy both the conditions if $S > 2H$, but so long as $S \leq 2H$ any value of G in the range from S to $2H$ satisfies them. In the previous section, we observed that the choice of the value of G is critical in determining where the final state will end up between the two extremes, too few 1's and too many 1's. To determine a suitable value for G let us assume that *the 0's and the 1's in the final state should be equally stable.*

The stability of a neuron output can be measured by how little it changes for a given perturbation in the activity. From the observation of the transfer function, sigmoidal or linear, it can be concluded that the farther the activity from zero, the more stable the output. The equal stability assumption, therefore, translates into equal magnitude of the activities of all the neurons. Since the activities of neurons at 0 and 1 in a valid state are $G - 2H$ and $G - S$ respectively, the equal stability assumption results in the equation $|G - S| = |G - 2H|$. Giving the value of G,

$$G = H + S/2, \tag{7}$$

which is also the midpoint of the allowed range for G. The value of G given by equation (7) satisfies conditions (5) and (6), if $H > S/2$.

States containing two or more 1's in a row or a column constitute a majority among the illegal corner states. The activity of each of these 1's in the final state (a corner state) will be $A \leq (G - H - S)$. To make sure that no such state be stable, it is sufficient to have $G - H - S < 0$. Using (7) the condition reduces to

$$S > 0. \tag{8}$$

Condition (8) ensures that no illegal state with more than one 1 in any row/column can be stable.

The condition $H > S/2$ will now be shown to be subsumed by a condition resulting from the convergence requirement on the self-connection. The convergence condition of the theorem requires that

$$-w_{a,a} = S < \left.\frac{dg_\lambda^{-1}(u)}{du}\right|_{min} = \frac{1}{\lambda}.\left.\frac{d(g^{-1}(u))}{du}\right|_{min}$$

so λ can not be increased beyond

$$\lambda_{max} = \frac{1}{S}.\left.\frac{dg^{-1}(u)}{du}\right|_{min}. \tag{9}$$

This equation shows that the choice of S must be made prudently, otherwise λ_{max} may not be large enough for the process to converge. Suppose for a constant A_0, $g(A_0)$ is *close enough* to 1, then the magnitude of the activity of every neuron in the final (valid) state must, at least, be equal to A_0/λ_{max}. So, from (9), $A_0.S/{g^{-1}}'_{min} \leq (G - S)$ or $A_0/{g^{-1}}'_{min} \leq (H - S/2)/S$, from (7). For a linear function the left hand side is greater than 1/2 because $A_0 \geq 1/2$. So $(H - S/2)/S$ should be greater than or equal to 1/2 or

$$S \leq H. \tag{10L}$$

For sigmoidal function it is not possible to choose A_0 such that $g(A_0) = 1$ so let us assume that $g(A_0 = 1) = .88$ (or $g(-1) = .12$) is close enough to 1 (or 0). Then $A_0/{g^{-1}}'_{min}$ is equal to 1/2, resulting in condition (10L). If the outputs of the neurons are expected to be closer to 1 (or 0) then a higher value of A_0 is required, giving $A_0/g^{-1'} >> 1/2$. This in turn implies that

$$S << H. \tag{10S}$$

DEFINITION: Σ-*states* are those illegal states which have no more than one 1 in each row and column and less than n 1's in all.

Let us now consider D. If D is set to zero, then valid and invalid legal states become indistinguishable because they have the same energy. Thus far D was ignored in the discussion. Conditions (4), (7), (8) and

(10) do not impose any constraints on D. Therefore, at this point we can only claim that a system subject to these condition can either converge to a Σ-state or a legal state. Our interest, however, is not in just any legal state but only in the valid ones. For that we turn to D.

The role of D is not to allow an edge, in the resulting Hamiltonian cycle, which does not belong to the original graph. The configuration in an invalid legal state which differentiates it from a valid one is 1's at (x,i) and $(y,i+1)$ where $d_{xy} = 1$. Repeating the argument applied to the case of two 1's in a row or a column, it can be concluded that $G - D - S < 0$ guarantees that such a configuration will never occur in the final state. The condition $D > G - S$ further restricts the types of states that can be finally arrived at (i.e., minimal points on the energy surface). Now, with the five conditions, only valid states or Σ states can be minimal points.

Experiments with the coefficients satisfying the conditions (4), (7), (8), (10) and $D > G - S$ were successful with graphs of medium and high level of connectivity. However, a new factor began to influence the outcome in very low connectivity graphs (most edges with weight 1). It was found that $D > G - S$ was too restrictive. The system frequently began to converge to Σ-states rather than to valid states. A closer look revealed that early in the process (i.e., small λ) system jumped to a state in which alternate columns were completely zero (two consecutive columns were zero at one place in case of an odd n). These were the only stable states in R_λ because the D-factor in (3), which is zero for these striped states, was too large for the others. These (striped) states are on the quasi-stationary paths of Σ states. Therefore, the restriction $D > G - S$ was relaxed. As the results of the experiments will show in the following section, low D was required for graphs with low connectivity.

5. Experiments with the network

Results of experiments with the graphs on up to 500 nodes are presented. Besides the HCP, a 318 city TSP [8] was solved with this net and the result is reported here.

Based on the constraints derived in the previous section, values of the coefficients were chosen to be $G = 1.0$, $H = V = 0.7$, and $S = 0.6$. Value of D was chosen in the range from 0.05 to 0.7 in the steps of 0.05 for each graph. The graphs were generated randomly with varying connectivity c, in the range from 10% to 90%. Following table shows the range of D for which the success rate was 100% for every family of graphs for a given n (number of nodes) and a connectivity.

n	c	D	%Success	Av. Cycles	c	D	%Success	Av. Cycles
20	.10	–	–	–	.20	.35 → .15	100%	311
	.50	.7 → .05	100%	281	.90	.7 → .05	100%	378
100	.10	–	–	–	.20	.55 → .05	100%	503
	.50	.7 → .05	100%	478	.90	.7 → .05	100%	358
200	.10	–	–	–	.20	.55 → .05	100%	666
	.50	.7 → .05	100%	458	.90	.7 → .05	100%	388
400	.10	.5 → .2	100%	912	.20	.7 → .05	100%	756
	.50	.7 → .05	100%	511	.90	.7 → .05	100%	407
500	.10	.5 → .1	100%	996	.20	.7 → .05	100%	858
	.50	.7 → .05	100%	550	.90	.7 → .05	100%	409

Results of the 318 city TSP. One experiment with a TSP was also performed to compare the performance of this net with earlier nets. The well known 318-city problem [8] was selected since it is the largest problem attempted by any method for which the results are published. The cycle cost varied, from 64552 to 61337, with the value of D. The cost of 61337 was achieved with $H = 0.7, S = 0.6$ and $D = 0.35$. It took a total of 446 cycles of firing. The result was no where near the best known cost of 41269 computed using linear programming and other optimization techniques. Yet, this is the only published result of a neural

net algorithm for any graph of more than 30 nodes to the best of our knowledge. The significance of the result can further be emphasized by observing that the average length of an edge in the 318 TSP graph is 1849.03 while that in the cycles determined by the neural net and the linear programming solution (best known cycle) are only 192.88 and 129.78 respectively.

CONCLUSION

The results of the experiments are stable enough to draw conclusions with confidence. Contrary to the intuition, the performance improves with the size of the graphs. As the results show, the 20 node graphs fail to give consistent results for 10% connectivity while 400 and 500 node graphs were solved successfully.

The cost of an iterative procedure such as this one is difficult to estimate with precision, but the experiments clearly show that the number of cycles grew much slower than $\sqrt{n}$. Therefore, it is fair to estimate the time complexity by $O(n^{3.5})$. Actually, the number of the cycles was more sensitive to c.

It is also interesting to note that the procedure did find a reasonably good solution of 318-city problem, the largest ever solved TSP [8]. It is true that the result was no where near the best known cycle of 41269, but still it is the only reported result computed using a neural network. In fact, no neural net succeeded in even arriving at any tour in a consistent manner, much less a near-optimal one, for n above 30, [9]. It is also worth noticing that the other algorithms which found the best known cycles have used the fact that the graph is Euclidean. The neural net method, on the other hand, does not exploit such information and it is perfectly general in its applicability.

REFERENCES

1. Brandt R.D., Wang Y., Laub A.J., Mitra S.K., "Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem," *Proceedings of the 2nd IEEE Neural Network Conference*, Vol. II, 1988, pp 333-340.
2. Gunn J.P., Weidlich R.B., "A Derivative of the Hopfield-Tank Neural Network Model that Reliably Solves the Traveling Salesman Problem," *Proceedings of the First International Joint Conference on Neural Networks*, June 1989, pp 588.
3. Hedge S.U., Sweet J.L., Levy W.B., "Determination of parameters in a Hopfield/Tank Computational Network," *Proceedings of the 2nd IEEE Neural Network Conference*, Vol. II., 1988, pp 291-298.
4. Hopfield J.J., "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, 79:2554-2558, 1982.
5. Hopfield J.J., "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences*, 81:3088-3092, 1984.
6. Hopfield J.J., Tank D.W., "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, Vol 52, 1985, pp 141-152.
7. Kirkpatrick S., Gelatt C.D. Jr., and Vecchi M.P., "Optimization by simulated Annealing," *Science*, Vol 220, 13 May 1983, pp 671-680.
8. Lawler E.L., Lenstra J.K., Rinnooy Kan A.G.H., and Shmoys D.B., "The traveling Salesman Problem: a Guided Tour of Combinatorial Optimization," John Wiley, 1985.
9. Protzel P., Palumbo D., and Arras M., "Fault-Tolerance of a Neural Network Solving The Traveling Salesman Problem," *NASA Contract Report* 181798, NASA Langley Research Center, Hampton, Virginia, Feb.1989.
10. Szu H., "Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input Using the Zero-Diagonal Interconnect Matrix and Necessary and Sufficient Constraints of the Permutation Matrix," *Proceedings of the 2nd IEEE Neural Network Conference*, Vol. II., 1988, pp 259-266.
11. Van den Bout D., Miller T.K., "A Traveling Salesman Objective Function That Works", *Proceedings of the 2nd IEEE Neural Network Conference*, Vol. II., 1988, pp 299-303
12. Wilson G.V. and Pawley G.S., "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank," *Biological Cybernetics*, Vol 58, 1988, pp 63-70.