J.G. McDonnell, R.J. Marks II and L.E. Atlas, "Neural networks for solving combinatorial search problems: a tutorial," Northcon/88 Conference Record, vol.II, pp.868-876, (Western Periodicals Co., North Hollywood, CA), Seattle WA, October 1988

Volume II

Northcon/88®

CONFERENCE RECORD

Sessions Presented at
Northcon/88
Seattle, WA
October 4 - 6, 1988

Sponsored by: Seattle and Portland Sections, Institute of Electrical and Electronic Engineers, IEEE

Cascade Chapter, Electronic Representatives Association, ERA

Electronics Manufacturers Association, EMA

NEURAL NETWORKS FOR SOLVING COMBINATORIAL SEARCH PROBLEMS: A TUTORIAL

Judson G. McDonnell
Boeing High Technology Center
Bellevue, WA 98124

Robert J. Marks II and Les E. Atlas
Interactive Systems Design Laboratory
University of Washington
Seattle, WA 98195

INTRODUCTION

In concert with the theme of this special session, our charter is to provide a tutorial introduction to artificial neural networks (ANN's) that solve certain combinatorial search problems. In the section to follow, we outline an architecture for a continuous time homogeneous neural network. Using simple lateral inhibition, the network is shown capable of locating that neuron with the maximum initial state. Building on this conceptual foundation, neural networks are developed that solve the *Queens Problem* and *Traveling Salesman Problems*.

Although the concept of applying ANN's to search problems of this type is intriguing, the jury is still out on whether the approach, when mature, will perform better than more conventional techniques. The potential advantages of ANN's include fault tolerance, regularized architectural structure, parallelism, and asynchronicity (i.e. no clock is needed).

A HOMOGENEOUS NEURAL NETWORK

Consider a set of L neurons (or nodes) each of which is assigned a value or state, $\{s_k : 1<k<L\}$. The j^{th} neuron is connected to the k^{th} neuron with an interconnect value or transmittance of t_{jk}. We will assume symmetric interconnects ($t_{jk} = t_{kj}$) and zero autoconnects ($t_{kk}=0$). The state of the k^{th} neuron is determined by the state of every other neuron. A first order analog circuit model for a single neuron [1] is shown in Figure 1. (Alternately, discrete synchronous and asynchronous linear algebra models can be used [2-3].) The neural states are introduced as voltages at the left and are fed to the neuron through the interconnect resistors shown. The voltage, e, is referred to as the excitation and is used in part to provide energy to the network. The capacitor provides the voltage inertia required to maintain a state sufficiently long to affect other neurons. Using KCL, it is easy to show that the circuit obeys the dynamics:

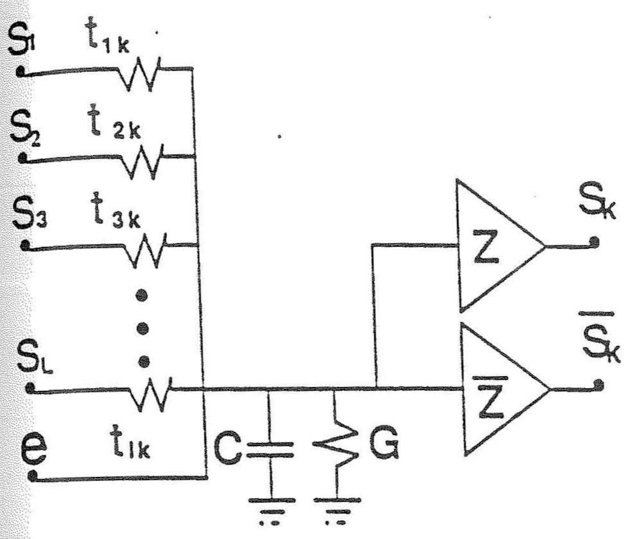

Figure 1: A first order circuit model for a neuron.

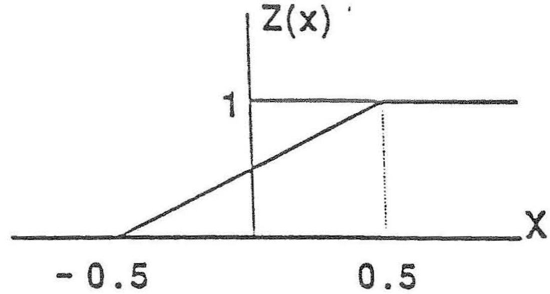

Figure 2: A sigmoid nonlinearity that has computational advantages in numerical simulations.

$$C\,u_k' = \Sigma\, t_{jk}\, S - u_k\, \Sigma\, t_{jk} - u_k\, G + e$$

where the prime denotes temporal differentiation.

The two triangular shaped circuit elements in Figure 1 are memoryless sigmoid nonlinearities. Sigmoid nonlinearities are continuous and monotonically increasing over the range of zero to one. If $Z(\cdot)$ denotes this nonlinearity, then the current neural state is:

$$s_k = Z(u_k)$$

The simulations in this paper use the easily computed nonlinearity shown in Figure 2. (In some ANN's, strictly increasing nonlinearities are required.) The bottom nonlinearity in Figure 1 outputs the negative of that produced at the top. Availability of this voltage allows simulations of negative interconnect resistances.

SOLUTION OF SEARCH PROBLEMS BY LATERAL INHIBITION ANN'S

If a neuron has a large state value and its interconnects to other neurons are negative, it attempts to reduce the state values of or 'turn off' the other neurons. This process is referred to as *lateral inhibition.* Such networks have demonstrated utility in spectral enhancement and vowel recognition [4-5]. Our attention will be restricted to application of lateral inhibition neural networks (LINN's) to solving certain combinatorial search problems. In order to effectively demonstrate this capability, we will show how a LINN can perform a 'search' for the maximum value in a sequence of numbers in the *King of the Hill Problem.* Dimensional

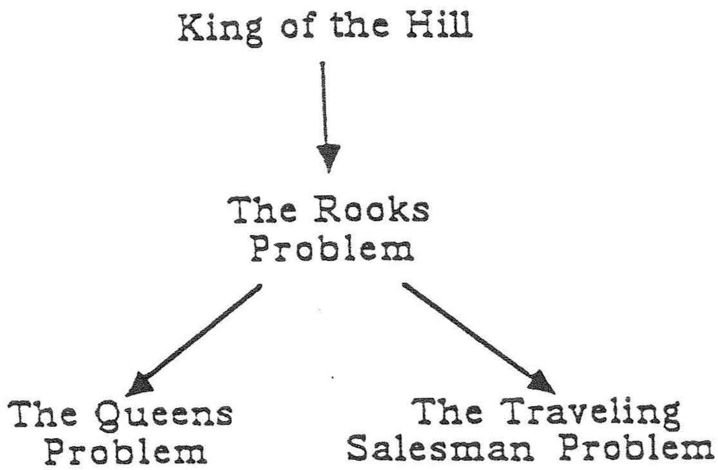

Figure 3: A hierarchical flowchart of search problems considered in this paper.

---

generalization leads us to a LINN that can solve the *Rooks Problem*. Specifically, how many rocks can be placed on a chess board so that no rook can capture another? (Placing them on a diagonal is an obvious solution.) Trivial though it may be, the *Rooks Problem* directly generalizes into both the *Traveling Salesman Problem* and *Queens Problem*. A flow chart of this developmental hierarchy is shown in Figure 3.

King of the Hill

A simple search problem is the *King of the Hill* or *Winner Take All Problem*. The goal of this problem is to isolate that neuron with the largest initial state, i.e., find the maximum. To do this, we construct a neural network where every neuron laterally inhibits every other neuron. Ideally, the neuron with the largest initial state inhibits the most and, due to the sigmoid, saturates in steady state at one. The remainder of the neurons ideally go to zero. In a 3 neuron system, for example, the matrix, T, of interconnects, t_{jk}, would be

$$T = \begin{bmatrix} 0 & -c & -c \\ -c & 0 & -c \\ -c & -c & 0 \end{bmatrix}$$

where the positive number c is the interneuron inhibition strength. In order to prohibit the network from degenerating to an all zero state, the LINN requires external excitation, e, applied to each neuron. Balancing the excitation against the constraints is, in general, still an art. For the King of the Hill Problem, however, we can straightforwardly reason through certain performance dynamics. With only one neuron on, all of the other neurons must be turned off with an inhibition strength of c. The external excitation, e, must therefore be less than c.

If e is very small, the network will approach the solution very slowly. If e is only slightly smaller than c, a neuron must be almost totally on in order to overwhelm the excitation. This will cause the equivalent of a overshoot, as other neurons will be partially on.. If e is greater than c but less than twice c, exactly two neurons will be on in steady state.

The Rooks Problem

The Rooks Problem requires the placing of rooks on an N by N chess board so that no rook can capture any other rook. Although easily solved, the *Rooks Problem* is a straightforward extension of the *King of the Hill Problem* and generalizes nicely to the *Queens, Traveling Salesman* and other problems [6].

Our LINN is best visualized as an N by N array of neurons -- one for each chess board square. We desire a steady state solution wherein there is only one neuron on in each row and column corresponding to the locations of the rooks on the chess board. Each row and column thus competes in a *King of the Hill Problem.* In each row and column of neurons, the interconnects are chosen to inhibit all of the other neurons in that row or column. Since every off neuron will be inhibited by two on neurons, the excitation, e, must be less than twice c.

To illustrate, consider the case where N=3 (or 9 neurons). If the neurons in the first row are numbered 1,2,3, -- those in the second 4,5,6, etc., then the matrix of interconnects is:

$$T = \begin{bmatrix} 0 & -c & -c & -c & 0 & 0 & -c & 0 & 0 \\ -c & 0 & -c & 0 & -c & 0 & 0 & -c & 0 \\ -c & -c & 0 & 0 & 0 & -c & 0 & 0 & -c \\ -c & 0 & 0 & 0 & -c & -c & -c & 0 & 0 \\ 0 & -c & 0 & -c & 0 & -c & 0 & -c & 0 \\ 0 & 0 & -c & -c & -c & 0 & 0 & 0 & -c \\ -c & 0 & 0 & -c & 0 & 0 & 0 & -c & -c \\ 0 & -c & 0 & 0 & -c & 0 & -c & 0 & -c \\ 0 & 0 & -c & 0 & 0 & -c & -c & -c & 0 \end{bmatrix}$$

Unlike the *King of the Hill* problem, the *Rooks* problem has a number of solutions. Indeed, different initializations will result in different solutions. Do to this diversity, we have the freedom of constraining the solution by *clamping* desired neural states to one, i.e. the inputs to the neuron are ignored and the neural state is set to one. If, for example, we clamped the upper left neuron to one, then the network will converge to a solution that places a rook in the upper left corner of the chess board. Multiple neural clamping, if consistent with the problem constraints, will produce a similar result. In the absence of clamping, the initial states of the neurons can be chosen stochastically.

The Queens Problem

The Queens Problem is a straightforward extension of the rooks problem: queens are used instead of rooks. The only further alteration to the network is use of lateral inhibition along

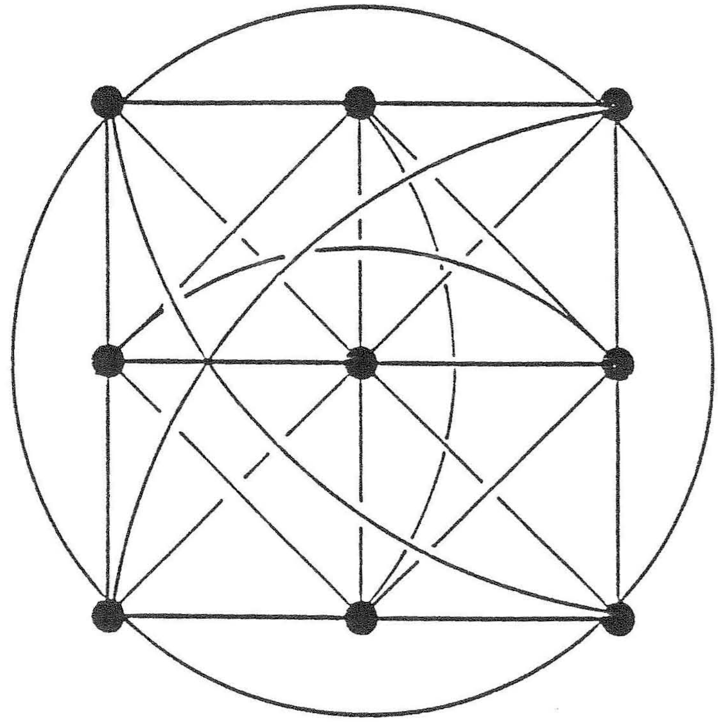

Figure 4: A LINN to solve the *Queens Problem* on a 3 by 3 chess board.
Every interconnect has a value of - c.

---

the diagonals of the chess board. An example of the interconnects for a 3 by 3 chess board is shown in Figure 4.

Examples of the dynamics of a LINN solution to the *Queens Problem* for a standard 8 by 8 chess board are shown in Figures 5 and 6. The neural states are coded by grey level: black is one and white zero. In both figures, the net's initial states were chosen randomly on the interval from zero to 1/2. In figure 5, various stages are shown in the evolution towards steady state. In general, for N>3, N queens can be placed on the chess board and satisfy the *Queens Problem* constraint. In Figure 5, there are only seven queens in steady state. Although the constraints of the problem have been met, an optimal solution was not obtained. Such results are characteristic of LINN's. The networks many times produce good but not optimal results. Unlike the *Queens Problem*, evaluation of the quality of the result in other search problems is usually not straightforward.

The solution in Figure 5 was not optimum because the networks excitation was too low. The value of was raised and the simulation illustrated in Figure 6 resulted. Here, convergence is to a proper solution using 8 queens.

As with *The Rooks* problem, one or more neurons can be clamped consistent with the problem constraints and the LINN will produce a consistent solution.

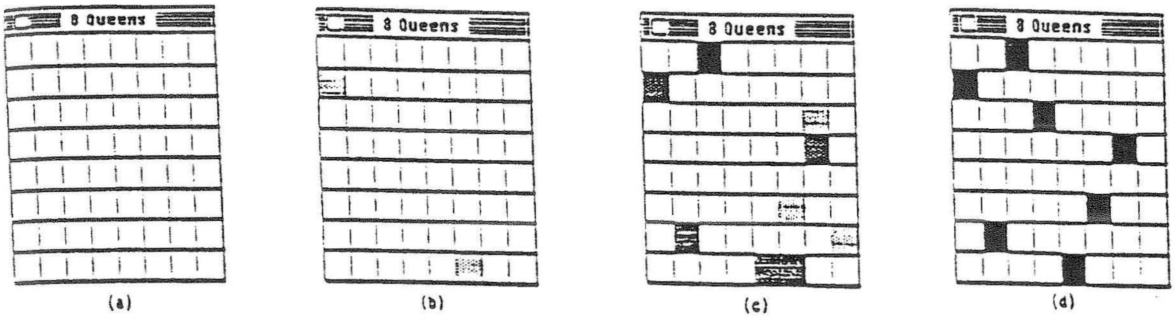

Figure 5: Snapshots of a LINN solving the *Queens Problem*. The final solution shown at the left, although meeting the constraints required of the problem, does not contain the maximum number of queens (8). This is because the excitation was too small (e = 0.1 and c = 0.15).

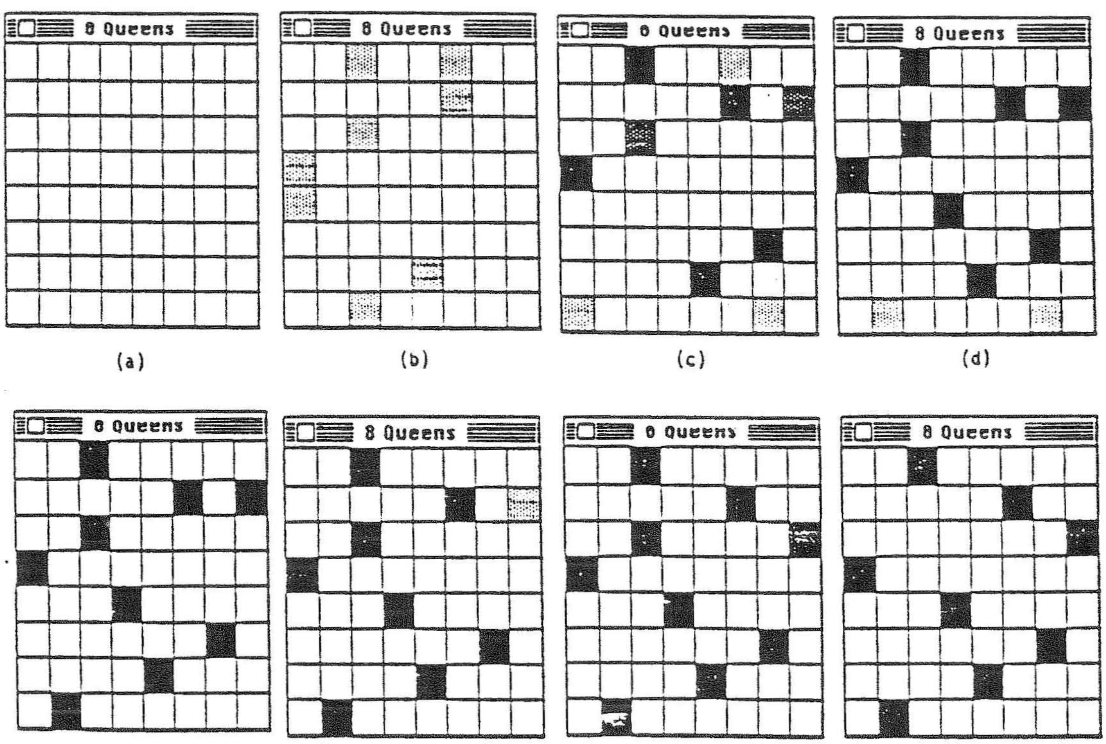

Figure 6: Snapshots of the same LINN as in Figure 5 with the excitation raised to e = 0.25. The steady state solution contains the maximum number of queens and meets the problem constraints.

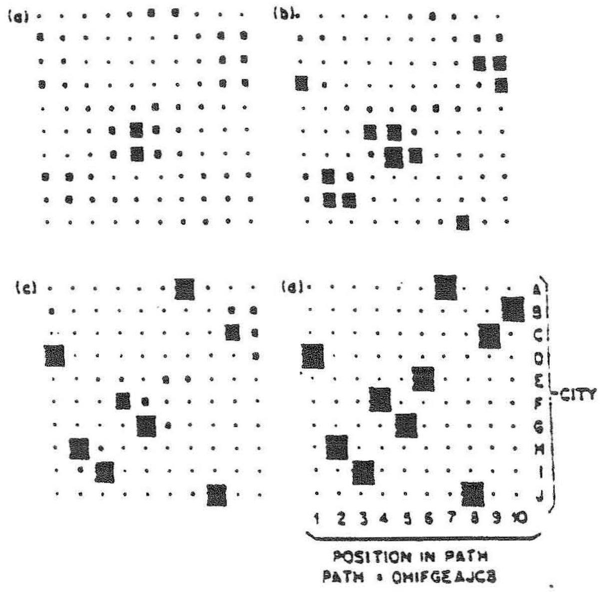

Figure 7: Snapshots of a LINN solving the *Traveling Salesman Problem* for a ten city tour. The result is that city D should be visited first, city H second, etc. The result is optimum (from Hopfield and Tank [6]).

The Traveling Salesman Problem

The Traveling Salesman Problem can also be considered as a generalization of the *Rooks Problem*. There are N cities. The distance between city X and Y is d_{XY}. We wish to arrange the cities so that the distance of a total round trip is minimum. We draw freely from the work of Hopfield and Tank [1].

An ANN solution for the *Traveling Salesman Problem* can be obtained from an N by N array of neurons. The k^{th} column of this array corresponds to the k^{th} city. The j^{th} row dictates the rank of the city on the salesman's itinerary. Thus, if the third neuron in column A is on, the salesman would visit city A third.

There are three classes of interconnects that comprise the LINN for this problem. The final value of an interconnect is obtained simply by summing the values of the three components:

(1) As with the *Rooks Problem*, a valid steady state solution can only have one neuron on in each row and each column. In order to assure that all cities are visited, each row and column must have one on neuron. The corresponding interconnect component values are identical to the interconnect values in the *Rooks Problem*.

(2) Additional inhibitory interconnect components are required to parameterize the distances between cities. Specifically, we desire neuron pairs corresponding to distant cities to inhibit more than those corresponding to cities that are close. Every neuron in the row for city X is thus connected to the corresponding neuron in city Y with the transmittance $-b\, d_{XY}$ where b is a some appropriate constant.

(3) Hopfield and Tank [1] also found use of a global inhibition useful. Each neuron inhibits every other neuron by a negative transmittance, $-\alpha$. This third interconnect component has the effect of squelching uprisings when a number of neurons are intensively competing.

Snapshots of the convergence of the neural network using a hyperbolic tangent sigmoid are shown in Figure 7. Here, the size of the square parameterizes the neural state. This particular solution is indeed optimum, but 'tweaking' of the system parameters was required. Typically, the network produces 'good' though not optimal results.

FINAL REMARKS

Artificial neural networks (ANN's) provide an intriquing architecture for the solution of combinatorial search problems. There are clear obstacles to using neural networks in such problems. To date, choosing inhibition values is more of an art than a science. The non-optimality of the solution can be a problem. Techniques such as simulated annealing [7-8], however, can be used to assure convergence to the optimal solutions in such cases. Convergence of such techniques can be quite slow. Nevertheless, preliminary results such as those presented in this paper are quite encouraging.

ACKNOWLEDGEMENTS

The authors express their gratitude to Ms. Cindy Chernoff for her assistance in the preparation of the manuscript. They also gratefully acknowledge the support of the Boeing High Technology Center in Bellevue Washington for their support of their work in neural networks. Marks and Atlas' work in neural networks is supported by the Washington Techology Center and Physio-Control. Les Atlas is also supported in part by an NSF PYI award.

REFERENCES

[1] J.J. Hopfield and D.W. Tank "Neural computation of decisions in optimization problems," Biol. Cybern., vol.52, p.141 (1985).

[2] Y.S. Abu-Mostafa and J.M. St. Jaques "Information capacity of the Hopfield model," **IEEE Trans. Inf. Theory**, vol.IT-31, p.461 (1985).

[3] R.J. Marks II "A class of continuous level associative memory neural networks," **Applied Optics**, vol.26, p.2005 (1987).

[4] E.F. Evans "Representation of complex sounds at the cochlear nerve and cochlear nucleus level" in **The Representation of Speech in the Peripheral Auditory System**, R. Carlson and B. Granstom, editors, Elsevier, Amsterdam (1982).

[5] L.E. Atlas, T. Homma and R.J. Marks II "A neural network for vowel classification," International Conference on Acoust., Speech and Sig. Proc., Dallas (April 1987).

[6] M. Takeda and J.W. Goodman "Neural networks for computation: number representations and programming complexity," **Applied Optics**, vol.25, p.3033 (1986).

[7] S. Kirkpatrick, C.D. Gelatt, Jr. and M.P. Vecchi "Optimization by simulated annealing," **Science Magazine**, vol.220, #4598, p.671 (May 1983).

[8] S. Geman and D. Geman "Stochastic relaxation, Gibbs distributions, and the Bayesian Restoration of Images," **IEEE Trans. Pattern Recognition and Machine Intelligence**, vol. PAMI-6, p.721 (1984).