Inhibitory Grids and the Assignment Problem

William J. Wolfe, *Member, IEEE,* James M. MacMillan, George Brady, Robert Mathews, Jay Alan Rothman, Donald Mathis, Michael Donald Orosz, Charlie Anderson, and Gita Alaghband

Abstract— **This paper analyzes a family of symmetric neural networks that *solve* a simple version of the Assignment Problem (AP). We analyze the suboptimal performance of these networks and compare the results to optimal answers obtained by linear programming techniques. We use the Interactive Activation model to define the network dynamics—a model that is closely related to the Hopfield-Tank model. A systematic analysis of hypercube corner stability and eigenspaces of the connection strength matrix leads to network parameters that provide feasible solutions 100% of the time and to a novel projection algorithm that significantly improves performance. Two formulations of the problem are discussed: i) *nearest* corner: encode the assignment numbers as initial activations, and ii) *lowest energy* corner: encode the assignment numbers as external inputs.**

I. Introduction

A solution to the Assignment Problem (AP), for the purposes of this paper, consists of choosing one entry from each row and column of a two-dimensional array of numbers in such a way as to attain the largest sum. For example, consider the following 10×10 array of numbers (randomly generated between 0 and 1), shown at the bottom of the page.

The array defines an AP for which the optimal answer is indicated by the underlined values. This solution corresponds to the permutation matrix on the bottom of the next page.

It is well known that the optimal selection can be found using standard linear programming techniques. The purpose of this paper is to compare these optimal selections with the selections found by a class of neural networks that use inhibitory rows and columns.

Several authors have shown how to apply neural networks to a variety of optimization problems ([1]–[8]). Much of the research focuses on analog circuits (and simulations) for the Traveling Salesman problem (TSP), while some researchers ([6], [9]–[11]) have analyzed other nonlinear programming problems in a similar manner. We are interested in the AP, for which related results are provided in [1], [15], and [11]. These results are incomplete and leave many unanswered questions concerning the convergence of such networks. It is our contention that some of the difficulties encountered in implementing neural networks to solve nonlinear problems such as the TSP, are also present—to a lesser degree— in linear problems such as the AP. This paper focuses on the synthesis of a class of neural networks that generate *good* solutions to the AP, providing insight into why the solutions are suboptimal.

We are not particularly interested in deriving a *new* way to solve the AP, since efficient linear programming methods exist. The only advantage to the neural approach is in providing a parallel representation—compatible with the use of parallel processors—which may be more effective for very large arrays, but we provide no conjecture as to how large an AP must be before suboptimal methods are beneficial.

We use the Interactive Activation model of dynamics [12], an apparently significant departure from the more common sigmoidal nonlinearity (i.e., Hopfield-Tank model), but surprisingly similar in overall performance. With Interactive Activation we can develop a relatively clear analytical foundation, similar to the analysis found in [13]–[16] for related networks, by emphasizing the eigenspaces of the connection strength matrix. In particular, the eigenspace analysis in [13] is very similar to that found here, a fact that we will expound upon in a later section. Furthermore, our approach can be viewed as an extension of our previous results for k-winner networks [17]. Finally, we demonstrate our general results with specific simulations of 10×10 arrays of randomly generated AP's.

Manuscript received April 1, 1991; revised May 10, 1992.

The authors are with the Department of Computer Science and Engineering, University of Colorado, Denver, CO 80204.

IEEE Log Number 9203744.

0.76	0.63	0.93	0.66	0.04	0.34	0.33	0.31	0.82	0.17
0.96	0.90	0.78	0.32	0.04	0.97	0.28	0.78	0.35	0.69
0.35	0.47	0.34	0.12	0.33	0.85	0.10	0.93	0.29	0.35
0.33	0.24	0.72	0.04	0.79	0.34	0.17	0.38	0.27	0.91
0.72	0.68	0.51	0.59	0.10	0.99	0.76	0.57	0.34	0.41
0.91	0.04	0.28	0.01	0.52	0.72	0.83	0.20	0.32	0.26
0.20	0.47	0.18	0.42	0.99	0.95	0.57	0.43	0.90	0.73
0.55	0.09	0.74	0.91	0.43	0.03	0.55	0.52	0.77	0.76
0.31	0.65	0.77	0.58	0.56	0.26	0.54	0.04	0.15	0.76
0.47	0.12	0.93	0.98	0.11	0.28	0.46	0.98	0.56	0.99

II. Definitions

Consider a symmetric network consisting of n units with bounded continuous activation values and the following parameters:

$\min \le a_i(t) \le \max, \qquad i = 1, \cdots, n;$

$w_{ij} = w_{ji}$(the connection strength between units $i \neq j$);

$w_{ii} = s$(self-connection);

ext_i (the external input to unit i).

Let $\boldsymbol{A}$ denote the symmetric $n \times n$ connection strength matrix. The n activation values form an n-dimensional column vector (or *state* vector), $\underline{a}$, restricted to lie within the n-cube defined by the minimum activation (min) and the maximum activation (max). When applied to a square array of units we take $n = k^2$ and refer to those corner states that correspond to permutation matrices as *feasible* states. An example of a feasible state for $k = 3$, written as a k^2-dimensional column vector and equivalently as a $k \times k$ array, with the corresponding permutation matrix, helps clarify the notation:

$$\underline{f} = \begin{bmatrix} \min \\ \mathbf{max} \\ \min \\ \mathbf{max} \\ \min \\ \min \\ \min \\ \min \\ \mathbf{max} \end{bmatrix} = \begin{bmatrix} \min & \mathbf{max} & \min \\ \mathbf{max} & \min & \min \\ \min & \min & \mathbf{max} \end{bmatrix} \xrightarrow[\text{permutation matrix}]{\text{corresponds to the}} \cdot \begin{bmatrix} 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} \end{bmatrix}.$$

Note that there are $k!$ feasible states. We interpret $k \times k$ arrays of activation values as k^2-dimensional column vectors by concatenating the rows of the array. In all computations we treat $k \times k$ arrays of activations as column vectors, and never as matrices, but we find it convenient to display them as matrices in order to describe particular patterns, such as the feasible states. The $k \times k$ array of numbers that define an AP will be used as the initial activations of the array of units, but we will also discuss the use of external inputs in a similar fashion. Henceforth, the $k \times k$ array of activation values is treated as an element of $\mathcal{R}^{k^2}$ that traverses the interior of the n-cube as it converges to a feasible state.

Activation values are updated synchronously using the following variant of Interactive Activation [12]:

$$a_i(t+1) = a_i(t) + \eta \Delta a_i(t)$$

$$\Delta a_i(t) = \text{Net}_i(t)[\max - a_i(t)], \qquad \text{if } \text{Net}_i(t) \ge 0;$$

$$\Delta a_i(t) = \text{Net}_i(t)[a_i(t) - \min], \qquad \text{if } \text{Net}_i(t) < 0;$$

$$\text{Net}_i(t) = \sum\nolimits_j a_j(t) w_{ij} + \text{ext}_i,$$

$\eta > 0$ is the step size.

The induced dynamics consists of two competing parts:

1) Gradient descent with respect to an energy function, implemented by the Net_i calculations. The energy of the network is defined to be: $E = -1/2 \underline{a}^T \boldsymbol{A} \underline{a} - \underline{a}^T \underline{\text{ext}}$. ($\underline{\text{ext}}$ = the vector of external inputs). Since $\partial E / \partial a_i = -\text{Net}_i$, $\Delta a_i(t) = \text{Net}_i(t)$ would be gradient descent with respect to this energy function. But the activation vector would not necessarily be restricted to the n-cube.
2) A *corner-seeking* part, implemented by $[\max - a_i(t)]$ and $[a_i(t) - \min]$. These factors keep the activation vector within the n-cube and *pushes* toward certain corners.

Theorem 1: Interactive Activation dynamics is equivalent to:

$$a_i(t+1) = a_i(t) + \eta \, \text{Net}_i(t) |b_i(t)|.$$

where

$$b_i(t) = c_i(t) - a_i(t)$$

and

$$c_i(t) = \max, \qquad \text{if } \text{Net}_i(t) \ge 0$$

$$c_i(t) = \min, \qquad \text{if } \text{Net}_i(t) < 0$$

$$\begin{array}{cccccccccc} 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Proof: $\mathrm{Net}_i(t) \geq 0 \Rightarrow c_i(t) = \max \Rightarrow b_i(t) = \max - a_i(t) \geq 0 \Rightarrow \Delta a_i(t) = \mathrm{Net}_i(t)[\max - a_i(t)]$. Similarly for $\mathrm{Net}_i(t) < 0$. Q.E.D.

This theorem provides geometric insight: $\underline{\mathrm{Net}}$ *chooses* a corner of the n-cube ($\underline{c}$) by virtue of the sign of each component. $\underline{b} = \underline{c} - \underline{a}$ runs from the tip of $\underline{a}$ to the chosen corner. The nonlinear dynamics attempts to move $\underline{a}$ toward the chosen corner but the motion is corrupted by multiplying the magnitude of each component by the corresponding component of $\underline{\mathrm{Net}}$. Thus, $\underline{a}$'s that are relatively far from their final, asymptotically stable, corner state are being updated using corners that may have very little relevance to the final state. This is balanced to some extent by the fact that $\underline{\mathrm{Net}}$ is pointing (locally) toward lower energy. Unfortunately, the local direction of $\underline{\mathrm{Net}}$ is used to select corners that, by virtue of their remoteness, may direct the motion away from a more predictable flow. This is especially likely when $\underline{a}$ is near one side of the n-cube and the negative gradient points toward the interior selecting a corner that is on the other side of the n-cube.

Therefore we must be careful while analyzing the gradient descent process in isolation, but the eigenspaces of $\boldsymbol{A}$ will nevertheless help us select network parameters that provide *near optimal* performance, with guaranteed feasible answers, but we cannot guarantee optimal results.

III. $k \times k$ Grid of Units

To formalize our approach to the AP we begin with $k \times k$ grid of units with the following assumptions:

1) All units in a common row or column are connected by a connection strength of -1, (i.e., mutually inhibitory rows and columns). These connection strengths will remain the same throughout the paper.
2) All other connection strengths are equal to α, except for the self-connections which are all set to s.
3) The maximum activation is set to $\max = +1$; but the minimum activation is treated as a variable $\min \leq 0$. We will actually let max be a positive variable for most of our theorems but all simulations reported at the end of this paper were done with $\max = +1$.

Thus, the variable parameters are min, α and s. Initially, we keep $s = 0$ and $\underline{\mathrm{ext}} = 0$. As the analysis proceeds we introduce a network that has a carefully chosen s. The external inputs will come into play in two different ways: i) we will discuss briefly the use of the same external input to all units to achieve stability of certain corner states; ad ii) we will also discuss the use of external inputs that are proportional to the numbers that define the AP (this lowers the energy of feasible corner states that correspond to better solutions to the AP, as will be described as follows). Finally, we note that the step size, η, is also a variable.

Theorem 2: When $\underline{\mathrm{ext}} = 0$, all the feasible states have the same vector length and the same energy.

Proof: Let $\underline{f}$ be any feasible state. By direct calculation:

$$\|\underline{f}\|^2 = k(\max)^2 + k(k-1)(\min)^2$$

and

$$E(\underline{f}) = -1/2(z_1 + z_2)$$

where

$$z_1 = k[2(k-1)(\max)(\min)(-1) + (k-1)(\max)^2(\alpha) + (k-1)(k-2)(\max)(\min)(\alpha) + s(\max)^2]$$

and

$$z_2 = k(k-1)[2(k-2)(\min)^2(-1) + 2(\min)(\max)(-1) + (k-2)(\min(\max))(\alpha) + \left((k-1)^2 - (k-2)\right)(\min)^2(\alpha) + s(\min)^2].$$

Q.E.D.

Theorem 3: Assume that the initial activation values define a particular AP. The network solves the AP, optimally iff the network converges to the *nearest* feasible corner state.

Proof: Let $\underline{a}$ denote the k^2-dimensional vector of initial activations, and $\underline{f}$ any feasible state. It suffices to show that the optimal $\underline{f}$ maximizes $\underline{a}^T\underline{f}$:

$$\underline{a}^T\underline{f} = \max\left(\sum{}' a_i\right) + \min\left(\sum{}'' a_i\right).$$

The first sum is over those values of $\underline{a}$ that correspond to the positions of the max's in f, and the second sum is over the rest of the values of $\underline{a}$. Let $\sum a_i$ be the sum of *all* the activations $\Rightarrow \sum a_i = \sum' a_i + \sum'' a_i$ and:

$$\underline{a}^T\underline{f} = \max\left(\sum{}' a_i\right) + \min\left(\sum a_i - \sum{}' a_i\right)$$
$$= (\max - \min)\left(\sum{}' a_i\right) - \min\left(\sum a_i\right)$$

This is as linear increasing function of $\sum' a_i$. Q.E.D.

Corollary: If the numbers that define the AP are used as the external inputs, then the feasible state with the lowest energy corresponds to the optimal solution.

Proof: The energy of a feasible state is given by $E = -1/2\underline{f}^T\boldsymbol{A}\underline{f} - \underline{f}^T\underline{\mathrm{ext}}$. From Theorem 2 we know that the first term is the same for all feasible states. By Theorem 3 we know that the second term has its largest magnitude when f is nearest to $\underline{\mathrm{ext}}$. Q.E.D.

If we encode the AP as the initial activations of the network, with external inputs set to zero, then convergence to the nearest feasible state constitutes optimal performance. If we encode the numbers as the external inputs to the network, with initial

activations set to zero, then convergence to the lowest energy feasible state constitutes optimal performance. If we encode the problem in both ways then the network would converge to the nearest and lowest energy feasible state. Later in this paper we provide statistical results for the initial activation formulation (externals set to zero). We will also discuss, however, the results of running networks using lowest energy formulations.

A brief sketch of what follows: we choose min so that certain corner states are orthogonal, then choose α so those corner states are eigenvectors with eigenvalues of the appropriate sign and magnitude, and then choose s so that the feasible states are the only asymptotically stable states.

IV. Choosing $\min = -1/(k-1)$

Let F denote the set of feasible states. Let E_c denote the set of corner states that correspond to arrays with all max's in one column, such as this example for $k = 3$ indicates:

$$\underline{e} = \begin{bmatrix} \mathbf{max} \\ \min \\ \min \\ \mathbf{max} \\ \min \\ \min \\ \mathbf{max} \\ \min \\ \min \end{bmatrix} = \begin{bmatrix} \mathbf{max} & \min & \min \\ \mathbf{max} & \min & \min \\ \mathbf{max} & \min & \min \end{bmatrix}.$$

Also let E_r denote the set of corner states that correspond to arrays with all max's in row, such as this example for $k = 3$ indicates:

$$\underline{e} = \begin{bmatrix} \mathbf{max} \\ \mathbf{max} \\ \mathbf{max} \\ \min \\ \min \\ \min \\ \min \\ \min \\ \min \end{bmatrix} = \begin{bmatrix} \mathbf{max} & \mathbf{max} & \mathbf{max} \\ \min & \min & \min \\ \min & \min & \min \end{bmatrix}.$$

Note that $\#E_c = \#E_r = k$.

Lemma 1 is a direct consequence of discrete Fourier analysis [19].

Lemma 1: Let $\underline{\cos}^m$ and $\underline{\sin}^m$ denote the k-dimensional vectors whose components are discrete samples $(i = 0, \cdots, k-1)$ of the Fourier harmonics:

$$\underline{\cos}^m = \sqrt{2/k} \begin{bmatrix} \cos(2\pi m 0/k) \\ \cos(2\pi m 1/k) \\ \cdot \\ \cdot \\ \cdot \\ \cos(2\pi m(k-1)/k) \end{bmatrix}$$

$$\underline{\sin}^m = \sqrt{2/k} \begin{bmatrix} \sin(2\pi m 0/k) \\ \sin(2\pi m 1/k) \\ \cdot \\ \cdot \\ \cdot \\ \sin(2\pi m(k-1)/k) \end{bmatrix}.$$

For $m = 1, \cdots k/2$ if k is even, or $m = 1, \cdots (k-1)/2$, if k is odd, there are $(k-1)$ such vectors and they form an orthonormal basis for the $(k-1)$-dimensional subspace of $\mathcal{R}^k$ spanned by the k-dimensional vectors x with the property that $\sum_i x_i = 0$.

Lemma 2: Let $\underline{v}^j$ denote the k-dimensional vectors that have a single max, in the jth position, and the rest min's:

$$\underline{v}^j = \begin{bmatrix} \min \\ \cdot \\ \cdot \\ \mathbf{max} \leftarrow j\text{th position.} \\ \cdot \\ \cdot \\ \min \end{bmatrix}$$

When $\min = -(\max)/(k-1)$ these vectors span the same $(k-1)$-dimensional subspace as the Fourier harmonics defined in Lemma 1.

Proof: First of all notice that the choice of min ensures that $\sum_i v_i^j = 0$ for any j and thus the $\underline{v}^j$'s are in the subspace spanned by the Fourier harmonics. Furthermore, we also know that any k-dimensional vector $\underline{x}$ with the property that $\sum_i x_i = 0$ can be expressed as a linear combination of the $\underline{v}^j$'s via: $\underline{x} = [1/(\max - \min)]\left[\sum_j x_j \underline{v}^j\right]$. Q.E.D.

Theorem 4: When $\min = -(\max)/(k-1)$ the sets E_c and E_r each span a $(k-1)$ dimensional subspace of $\mathcal{R}^{k^2}$, call them $\mathcal{E}_c$ and $\mathcal{E}_r$, and furthermore, these subspaces are mutually orthogonal.

Proof: We prove that E_c and E_r are mutually orthogonal sets, and therefore the spanned subspaces $\mathcal{E}_c$ and $\mathcal{E}_r$ are as well. The inner product of any vector from E_c with any vector from E_r is given by: $(\max)^2 + 2(k-1)(\max)(\min) + (k-1)^2(\min)^2$ which is equal to zero when $\min = -(\max)/(k-1)$.

Now we demonstrate the dimensionality of $\mathcal{E}_c$ by defining a set of $(k-1)$ orthonormal spanning vectors by using $\underline{\cos}^m$ and $\underline{\sin}^m$ as the repeated rows of a matrix representation:

$$\begin{bmatrix} \cos(2\pi m0/k) & \cos(2\pi m1/k) \cdots \cos(2\pi m(k-1)/k) \\ \cos(2\pi m0/k) & \cos(2\pi m1/k) \cdots \cos(2\pi m(k-1)/k) \\ & \bullet \\ & \bullet \\ & \bullet \\ \cos(2\pi m0/k) & \cos(2\pi m1/k) \cdots \cos(2\pi m(k-1)/k) \end{bmatrix}$$

and

$$\begin{bmatrix} \sin(2\pi m0/k) & \sin(2\pi m1/k) \cdots \sin(2\pi m(k-1)/k) \\ \sin(2\pi m0/k) & \sin(2\pi m1/k) \cdots \sin(2\pi m(k-1)/k) \\ & \bullet \\ & \bullet \\ & \bullet \\ \sin(2\pi m0/k) & \sin(2\pi m1/k) \cdots \sin(2\pi m(k-1)/k) \end{bmatrix}.$$

These vectors each have length $k/\sqrt{2}$, with the exception that when k is even and $m = k/2$, $\underline{\cos}^{k/2}$ has length k, while $\underline{\sin}^{k/2}$ is all zeros. Denote the normalized versions of these vectors as $\underline{\cos}^{m,\mathrm{col}}$ and $\underline{\sin}^{m,\mathrm{col}}$. Using Lemmas 1 and 2 we see that these k^2-dimensional vectors span the same subspace as the vectors from E_c. Notice that the vectors in $\mathcal{E}_c$ have the property that the sum of any *row* is zero. Finally we can use the same cosines and sines as *columns* to prove the analogous result for $\mathcal{E}_r$, noting that the vectors in $\mathcal{E}_r$ have the property that the sum of any *column* is zero. Q.E.D.

Theorem 5: When $\min = -(\max)/(k-1)$ the vectors of F span a $(k-1)^2$-dimensional subspace of $\mathcal{R}^{k^2}$, call it $\mathcal{F}$, that is orthogonal to $\mathcal{E}_c$ and $\mathcal{E}_r$

Proof: First we show that any vector in F is perpendicular to any vector in E_c or E_r. The inner product of any vector from F with any vector from E_c or E_r is given by: $(\max)^2 + 2(k-1)(\max)(\min) + (k-1)^2(\min)^2$ which is equal to zero when $\min = -(\max)/(k-1)$. (Note that this is the same expression as in Theorem 4).

The fact that F spans a $(k-1)^2$-dimensional subspace $(\mathcal{F})$ turns out to be equivalent to the Birkhoff-von Neumann theorem [18] for doubly stochastic matrices (i.e., nonnegative matrices whose sums of rows and columns is 1) To see the connection let $\mathcal{G}$ be the space of all vectors whose sums of rows and columns is zero. Any vector in $\mathcal{G}$ can be translated and scaled into a doubly stochastic matrix which, by the Birkhoff-von Neumann theorem, can be expressed as a linear combination of permutation matrices. Furthermore, any permutation matrix can be expressed as a translation and scaling of a feasible state. Thus any vector in $\mathcal{G}$ can be expressed as a linear combination of feasible states. Therefore $\mathcal{F} = \mathcal{G}$ and the dimension of $\mathcal{G}$ is $(k-1)^2$ because a $(k-1)^2$ block of any matrix representation of a vector in $\mathcal{G}$ can be filled in arbitrarily and the remaining $2k-1$ entries must be used to make the rows and columns sum to zero. Notice that the sum of any column or any row of a vector in $\mathcal{F}$ is zero. Q.E.D.

Let $\mathcal{D}$ be the one-dimensional space spanned by:

$$\underline{d} = 1/k \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} = 1/k \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \cdots & & & \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Theorem 6: When $\min = -(\max)/(k-1)$, $\mathcal{R}^{k^2}$ is the direct sum:

$$\mathcal{R}^{k^2} = \mathcal{F} \oplus \mathcal{E}_c \oplus \mathcal{E}_r \oplus \mathcal{D}.$$

Proof: Since the sum of any row or column of any vector in $\mathcal{F}$ is zero, $\mathcal{F}$ is orthogonal to $\mathcal{D}$. Similarly, since the sum of any row of a vector in $\mathcal{E}_c$ is zero and any column of a vector in $\mathcal{E}$ is zero they are also orthogonal to $\mathcal{D}$. Q.E.D.

V. Projecting onto the Feasible Subspace

We now have the following othonormal basis for the subspace $\mathcal{E}_c \oplus \mathcal{E}_r \oplus \mathcal{D}$:

$$\{\underline{\cos}^{m,\mathrm{col}}, \underline{\sin}^{m,\mathrm{col}}, \underline{\cos}^{m,\mathrm{row}}, \underline{\sin}^{m,\mathrm{row}}, \underline{d}\}$$

$$(m = 1, \cdots, [k/2]).$$

(Here $[k/2]$ stands for the "largest integer less than or equal to $k/2$"). From Theorem 6 we know that any vector $\underline{a} \in \mathcal{R}^{k^2}$ can be expressed as:

$$\begin{aligned} \underline{a} = \underline{a}_F &+ \sum\nolimits_m \left(\gamma_m^{\cos,\mathrm{col}}\right)\left(\underline{\cos}^{m,\mathrm{col}}\right) \\ &+ \sum\nolimits_m \left(\gamma_m^{\sin,\mathrm{col}}\right)\left(\underline{\sin}^{m,\mathrm{col}}\right) \\ &+ \sum\nolimits_m \left(\gamma_m^{\cos,\mathrm{row}}\right)\left(\underline{\cos}^{m,\mathrm{row}}\right) \\ &+ \sum\nolimits_m \left(\gamma_m^{\sin,\mathrm{row}}\right)\left(\underline{\sin}^{m,\mathrm{row}}\right) + \gamma \underline{d}. \end{aligned}$$

Where $\underline{a}_F$ is the orthogonal projection of $\underline{a}$ onto $\mathcal{F}$. The coefficients (γ's) are obtained via inner products:

$$\gamma_m^{\cos,\mathrm{col}} = \underline{a}^T \underline{\cos}^{m,\mathrm{col}}$$

$$\gamma_m^{\sin,\mathrm{col}} = \underline{a}^T \underline{\sin}^{m,\mathrm{col}}$$

$$\gamma_m^{\cos,\mathrm{row}} = \underline{a}^T \underline{\cos}^{m,\mathrm{row}}$$

$$\gamma_m^{\sin,\mathrm{row}} = \underline{a}^T \underline{\sin}^{m,\mathrm{row}}$$

$$\gamma = \underline{a}^T \underline{d}.$$

If we rearrange the sums implied by these inner products and

let:

$$\text{colsum}_i = \text{ the sum of the } i\text{th column of } \underline{a}$$

$$\text{rowsum}_i = \text{ the sum of the } i\text{th row of } \underline{a}$$

then

$$\gamma_m^{\cos,\text{col}} = \left(2/\sqrt{k}\right)\left\{\sum_i \text{colsum}_i \cos(2\pi mi/k)\right\}$$

$$\gamma_m^{\sin,\text{col}} = \left(2/\sqrt{k}\right)\left\{\sum_i \text{colsum}_i \sin(2\pi mi/k)\right\}$$

$$\gamma_m^{\cos,\text{row}} = \left(2/\sqrt{k}\right)\left\{\sum_i \text{rowsum}_i \cos(2\pi mi/k)\right\}$$

$$\gamma_m^{\sin,\text{row}} = \left(2/\sqrt{k}\right)\left\{\sum_i \text{rowsum}_i \sin(2\pi mi/k)\right\}.$$

Thus the coefficients are discrete Fourier transforms (without the $m = 0$ term) of the row and column sums, viewed as k-dimensional vectors. Also notice that $\gamma = \underline{a}^T\underline{d} = (1/k)(\sum_i a_i) = (k)(\text{avg})$, where $\text{avg} = (1/k)^2(\sum_i a_i)$ and hence $\gamma\underline{d}$ is just the vector with avg for every component. Finally, by subtracting the associated terms from $\underline{a}$ we attain the projection $\underline{a}_F$.

As an example, consider the 10×10 array of random numbers displayed at the beginning of this paper. Applying this projection gives the array shown at the bottom of the page.

The sum of any row or column of this matrix is zero, as expected. Furthermore it is important to notice that the values may no longer lie within the n-cube defined by min and max, a fact that will cause us to "clip" some activations when applying this projection.

VI. Eigenspaces of the Connection Strength Matrix ($\boldsymbol{A}$)

We now show that $\mathcal{F}$, $\mathcal{E}_c$, $\mathcal{E}_r$ and $\mathcal{D}$ are the degenerate eigenspaces of the connection strength matrix $\boldsymbol{A}$.

Theorem 7: $\mathcal{F}$ is an eigenspace of $\boldsymbol{A}$ with eigenvalue $\lambda_{\boldsymbol{F}} = (\alpha + s + 2)$.

Proof: Let $\underline{f}$ be any feasible state. With $\underline{\text{ext}} = 0$, $\boldsymbol{A}\underline{f} = \underline{\text{Net}}$ and we need only compute the net input to a max unit and a min unit, recalling that $\min = -(\max)/(k-1)$:

$$\begin{aligned}\text{Net}_{\max} &= 2(k-1)(\min)(-1) + (k-1)(\alpha)(\max) \\ &\quad + (k-1)(k-2)(\min)(\alpha) + s(\min) \\ &= (\max)(\alpha + s + 2).\end{aligned}$$

$$\begin{aligned}\text{Net}_{\min} &= 2(k-2)(\min)(-1) + 2(\max)(-1) \\ &\quad + (k-2)(\max)(\alpha) \\ &\quad + \left((k-1)^2 - (k-2)\right)(\min)(\alpha) + s(\min). \\ &= (-(\max)/(k-1))(\alpha + s + 2) \\ &= (\min)(\alpha + s + 2).\end{aligned}$$

Since the feasible states span $\mathcal{F}$ we now see that $\mathcal{F}$ is a $(k-1)^2$-dimensional degenerate eigenspace of $\boldsymbol{A}$ with eigenvalue $\lambda_{\boldsymbol{F}} = \alpha + s + 2$. Q.E.D.

Theorem 8: $\mathcal{E}_c$ and $\mathcal{E}_r$ are eigenspaces of A with the same eigenvalue $\lambda_{\boldsymbol{E}} = -(k-1)(\alpha) + s - (k-2)$.

Proof: We need only do the following two computations, applicable to any vector in E_c or E_r, again using $\min = -(\max)/(k-1)$:

$$\begin{aligned}\text{Net}_{\max} &= (k-1)(\min)(-1) + (k-1)(\max)(-1) \\ &\quad + (k-1)^2(\min)(\alpha) + s(\max) \\ &= (\max)(-(k-1)(\alpha) + s - (k-2)).\end{aligned}$$

$$\begin{aligned}\text{Net}_{\min} &= (2k-3)(\min)(-1) + (\max)(-1) \\ &\quad + (k-1)(\max)(\alpha) \\ &\quad + ((k-1)(k-2))(\min)(\alpha) + s(\min). \\ &= (\min)(-(k-1)(\alpha) + s - (k-2)).\end{aligned}$$

Since E_c and E_r span $\mathcal{E}_c$ and $\mathcal{E}_r$ we see that they are $(k-1)$-dimensional degenerate eigenspaces of $\boldsymbol{A}$. Q.E.D.

Theorem 9: $\mathcal{D}$ is a one-dimensional eigenspace of A with eigenvalue $\lambda_{\boldsymbol{D}} = (k-1)^2\alpha + s - 2(k-1)$.

+0.21	+0.21	+0.32	+0.21	−0.34	−0.22	−0.12	−0.19	+0.35	−0.42
+0.30	+0.38	+0.06	−0.24	−0.44	+0.29	−0.27	+0.16	−0.23	−0.01
−0.11	+0.14	−0.18	−0.24	+0.04	+0.37	−0.27	+0.51	−0.09	−0.16
−0.14	−0.10	+0.19	−0.34	+0.49	−0.14	−0.20	−0.04	−0.12	+0.40
+0.11	+0.19	−0.17	+0.07	−0.35	+0.35	+0.25	+0.00	−0.19	−0.25
+0.46	−0.29	−0.24	−0.35	+0.23	+0.25	+0.47	−0.21	−0.06	−0.24
−0.43	−0.04	−0.51	−0.12	+0.52	+0.30	+0.03	−0.16	+0.35	+0.05
−0.03	−0.36	+0.09	+0.42	+0.01	−0.57	+0.06	−0.02	+0.27	+0.13
−0.20	+0.27	+0.20	+0.17	+0.21	−0.27	+0.13	−0.43	−0.28	+0.20
−0.16	−0.39	+0.23	+0.43	−0.36	−0.37	−0.38	+0.38	+0.01	+0.31

Proof: Let $\underline{d}$ be the diagonal vector defined earlier. The net input to any unit is independent of the choice of min and given by

$$\text{Net} = 2(k-1)(1/k)(-1) + (k-1)^2(1/k)(\alpha)s(1/k).$$

$$= (1/k)\Big((k-1)^2\alpha + s - 2(k-1)\Big).$$

Q.E.D.

With regard to the eigenspaces developed in [13] we see there that they are analyzing the analogous problem en route to representing the TSP. To correlate our results with theirs we need to display their notation:

$$\lambda_1 = -Ck^2 - 2A(k-1)$$

$$\lambda_2 = 2A$$

$$\lambda_3 = -A(k-1).$$

This "A, B, C, D" notation refers to parameters in Hopfield's original formulation of the TSP [2]. These eigenvalues correspond to our $\mathcal{D}$, $\mathcal{F}$, and $\mathcal{E}$ spaces, in that order. In fact, when $A = 1$ and $C = 0$ we see that λ_1, λ_2 and λ_3 are exactly equal to $\lambda_{\boldsymbol{D}}$, $\lambda_{\boldsymbol{F}}$ and $\lambda_{\boldsymbol{E}}$ for the case when $\alpha = 0$. Since they use $\min = 0$ and $\max = +1$, it appears that the need for the C factor is offset in our formulation by the use of $\min = -(\max)/(k-1)$, a fact that also allows us to derive the eigenspaces in a cleaner fashion. Furthermore, we have generalized to the case where α is nonzero.

VII. Stability Analysis

A corner state is stable if the net input to any max unit is positive and the net input to any min unit is negative. This is a direct consequence of the Interactive Activation update equations.

Theorem 10: For $s = 0$, $\underline{\text{ext}} = 0$, and α in the range: $-1 \le \alpha \le 1/(k-2)$, the feasible states are stable if:

$$\frac{(k-2)(\alpha) - 2}{2(k-2) - \Big((k-1)^2 - (k-2)\Big)(\alpha)} < \min$$

$$< \frac{\alpha}{2 - (k-2)(\alpha)}.$$

Proof: From Theorem 7 we know that the net input to a max unit in a feasible state is given by:

$$\text{Net}_{\max} = 2(k-1)(\min)(-1) + (k-1)(\alpha)(\max)$$

$$+ (k-1)(k-2)(\min)(\alpha).$$

For stability we need $\text{Net}_{\max} > 0$. We solve the inequality for min, using the fact that the restriction on α causes the inequality to reverse orientation after division by the coefficient of min, to get

$$\min < \frac{\alpha}{2 - (k-2)(\alpha)}.$$

Similarly we know that

$$\text{Net}_{\min} = (2k-3)(\min)(-1) + (\max)(-1)$$

$$+ (k-1)(\max)(\alpha) + ((k-1)(k-2))(\min)(\alpha).$$

To achieve $\text{Net}_{\min} < 0$, we solve this inequality for min to get

$$\min > \frac{(k-2)(\alpha) - 2}{2(k-2) - \Big((k-1)^2 - (k-2)\Big)(\alpha)}.$$

Q.E.D.

Although these conditions provide stable feasible states, they do not preclude other corner states from also being stable. In particular we have observed that the following three types of corner states must be made unstable to ensure that the feasible states are the *only* stable corners of the system.

Type 1: The state consisting of all min's:

$$\begin{bmatrix} \min & \min & \cdots & \min \\ \min & \min & \cdots & \min \\ \cdots & & & \\ \min & \min & \cdots & \min \end{bmatrix}.$$

This state is stable iff $\min > 0$. We have already restricted $\min \le 0$ in our original parameter definitions so this state will be unstable if we add the slightly stricter condition: $\min < 0$

Type 2: Corner states with $k+1$ max's (the rest min's) and having the property that if we chose any max in the matrix representation of the state there would be at most one other max in the same row and column. Here is an example for $k = 4$:

$$\begin{bmatrix} \mathbf{max} & \min & \min & \mathbf{max} \\ \min & \mathbf{max} & \min & \min \\ \min & \min & \mathbf{max} & \min \\ \min & \min & \mathbf{max} & \min \end{bmatrix}$$

A similar analysis of the net input to particular max's and min's leads to the following condition on min for the stability of these states:

$$\frac{(k-1)(\alpha) - 2}{2(k-2) - (k-1)(k-2)(\alpha)} < \min$$

$$< \frac{(k-1)(\alpha) - 1}{(2k-3) - (k-1)(k-2)(\alpha)}.$$

This overlaps with Theorem 10, but allows choices of α and min that make these states unstable while retaining the stability of the feasible states (see Fig. 1).

Type 3: Corner states with k max's but exactly one complete row or column of min's, such as this example for $k = 4$:

$$\begin{bmatrix} \mathbf{max} & \min & \min & \min \\ \min & \mathbf{max} & \min & \min \\ \min & \min & \mathbf{max} & \min \\ \min & \min & \mathbf{max} & \min \end{bmatrix}.$$

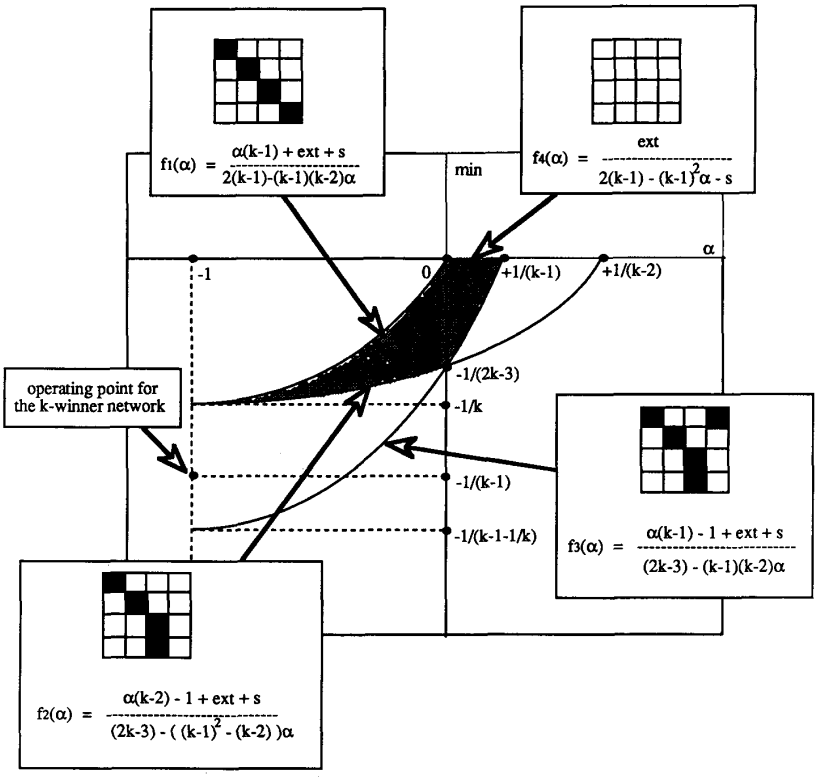


Fig. 1. A sketch of the $(\alpha, \min)$ parameter space for $s = 0$ and $\underline{\text{ext}} = 0$. The shaded region provides values that guarantee that the feasible states are the only stable corner states of the network. Each boundary curve is derived from a stability analysis of a type of potentially stable corner state, shown in the sketch as a 4×4. (Note that the sketch is not drawn to scale).

Analyzing the net inputs to max's and min's leads to

$$\frac{(k-1)(\alpha)-1}{(2k-3)-(k-1)(k-2)(\alpha)} < \min < \frac{(k-2)(\alpha)-1}{(2k-3)-\left((k-1)^2-(k-2)\right)(\alpha)}.$$

This stability region also overlaps with Theorem 10, but again can be avoided while maintaining the stability of the feasible states.

Fig. 1 summarizes these results. The boundary curves are sketched as though $s = 0$ and $\underline{\text{ext}} = 0$, but the functions that define these curves are expressed in terms of variable s and $\underline{\text{ext}}$. Notice the operating point for the k-winner networks derived in [17]. That is, if $\alpha = -1$ and $\min = -1/(k-1)$, the network would converge to the largest k initial values, ignoring the one per row, one per column constraint. Values of α and min chosen from the shaded region provide networks for which the feasible states are the only stable corner states of the system.

VIII. Three Networks

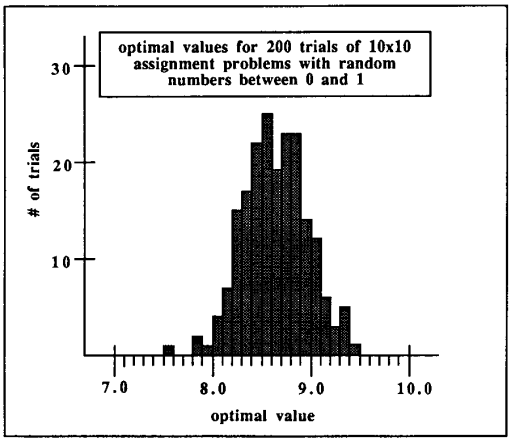
We are now in a position to select specific network parameters and compare the results to optimal AP answers. We will define three specific networks, all with $\max = +1$, and give a justification for the parameter selections. In these simulations we encode the numbers that define the AP as the initial activations of the network, with $\underline{\text{ext}} = 0$. We want networks that always provide feasible answers.

A. Parameters for Network "A"

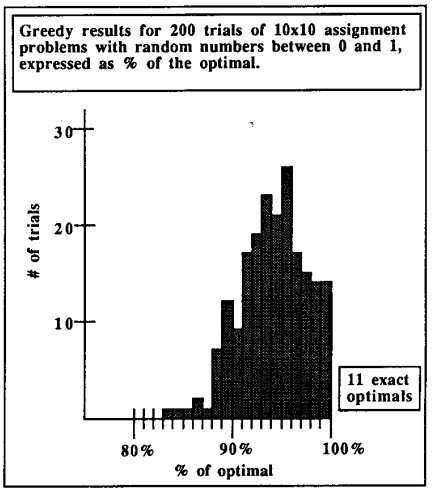
The basic strategy in choosing the parameters for this first network is to make it as simple as possible. Thus we would like to have $\alpha = 0$ simply because that drastically reduces the number of nonzero connections and by scanning Fig. 1 we see that there is a range of min's available when $\alpha = 0$: $0 > \min > -1/(2k-3)$. We prefer $\min = -1/(k-1)$ since the feasible states would then be eigenvectors of $\boldsymbol{A}$, but that choice is not available within the shaded region. So, we compromise and choose $\min = -1/(2(k-1))$. Thus the parameters for Network "A" are: $s = 0$, $\alpha = 0$, and $\min = -1/(2(k-1))$.

B. Parameters for Network "B"

Still trying to keep it simple, we keep $s = 0$. From Fig. 1 we see that the largest we can make α is $+1/(k-1)$, corresponding to $\min = 0$. This choice of α has an additional property: when $\alpha = +1/(k-1)$ the eigenvalues of the nonfeasible subspaces, $\mathcal{E}_c$, $\mathcal{E}_r$, and $\mathcal{D}$, are negative and



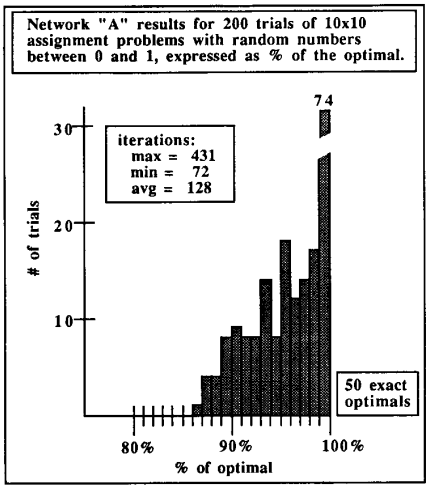
(a)



(b)

K = 10	Network "A"	Network "B"	Network "C"
min	-1/(2(K-1))	0.0	-1/(K-1)
α	0.0	+1/(K-1)	+1/(K-1)
s	0.0	0.0	-1-1/(K-1)
η	0.1	0.1	0.1

(c)



(d)

Fig. 2. A comparison of Networks, "A", "B", and "C" with greedy (G) results.

equal:

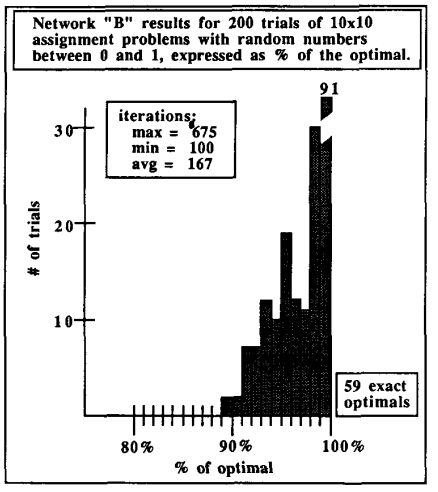
$$\lambda_{\boldsymbol{E}} = -(k-1)\alpha + s - (k-2) = -1 + s - (k-2)$$
$$= -(k-1) + s.$$
$$\lambda_{\boldsymbol{D}} = -(k-1)^2\alpha + s - 2(k-1)$$
$$= (k-1) + s - 2(k-1)$$
$$= -(k-1) + s.$$

It is our intuition that this condition will cause the network dynamics to flow more uniformly out of the nonfeasible subspace as it converges to a corner in the feasible subspace. This conjecture is born out to some extent by the improved performance of Network "B" over Network "A" but it is difficult to separate the effects of the other parameter differences.

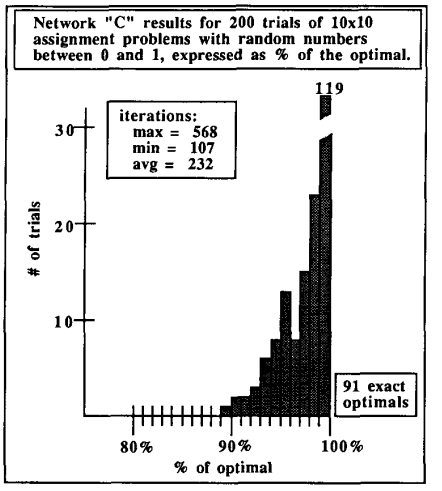
C. Parameters for Network "C"

We now forgo simplicity to achieve a theoretically satisfying selection of parameters. We prefer to have both $\min = -1/(k-1)$ and $\alpha = +1/(k-1)$. Recall that Fig. 1 was constructed under the assumption the $s = 0$. We now choose s so that the feasible states are the only asymptotically stable states of the system when $\alpha = +1/(k-1)$ and $\min = -1/(k-1)$. This involves shifting the boundary curves in Fig. 1 by making s more negative, until the operating point $(+1/(k-1), -1/(k-1))$ is in the feasible region. When $s = -1$ the desired point is on the lower boundary curve and when $s = -2 - 1/(k-1)$ it is on the upper boundary curve. A reasonable choice is $s = -1 - 1/(k-1)$. Thus the parameters for Network "C" are $\min = -1/(k-1)$, $\alpha = +1/(k-1)$, and $s = -1 - 1/(k-1)$.

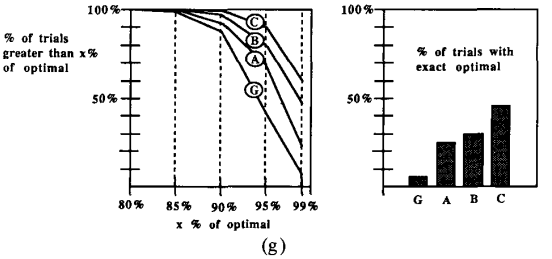
Before closing this section it is worthwhile to point out that similar analysis of the parameter space in Fig. 1 leads



(e)



(f)



(g)

Fig. 2. *(Continued).*

to regions where other types of corner state are exclusively stable. In particular, empirical evidence indicates that keeping $\min = 0$ while gradually increasing α, leads to networks for which there are exactly two winners in every row and column, three winners, etc. This may be useful for extending this work to more general task scheduling and transportation problems.

IX. Stastistical Results

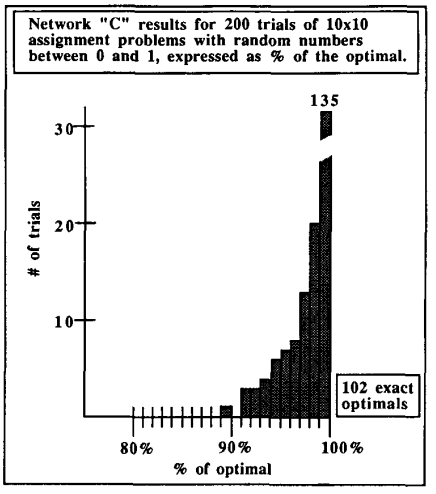
Experiment #1:

The first data set is shown in Fig. 2. The three networks described in the previous section, Networks "A", "B", and "C", were used simultaneously on 200 different initializations of random numbers between 0 and 1. For each trial a 10 by 10 array of random numbers between 0 and 1 was generated and used as the same initial activations for all three networks. Before running the networks however, the optimal solution was calculated using linear programming, allowing us to represent our results as percents of the optimal solution. For comparison purposes the results of applying the "greedy" algorithm to the same data are also shown. The "greedy" algorithm is a simple heuristic that selects the largest element in the array, deletes the corresponding row and column from further consideration, then selects the largest of the remaining elements, and so on. All three networks always provide feasible answers, confirming the validity of Fig. 1. Furthermore, the networks appear statistically better than the greedy results and the networks progress in improved performance from "A" to "C," as predicted by the eigenspace analysis.

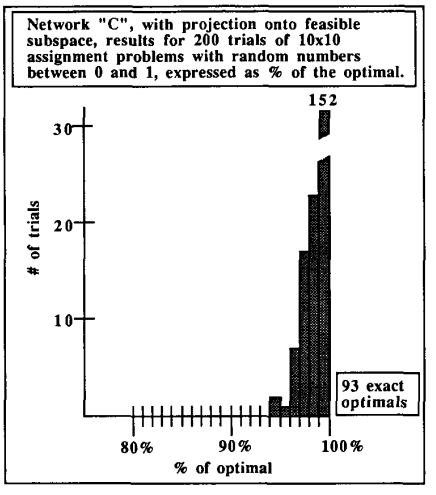
Experiment #2:

The second data set also consists of 200 trials using only Network "C". But this time the activation vector is orthogonally projected onto the feasible space after each neural update. Since the projection can take the activation vector outside of the n-cube it was necessary to "clip" some activation values to min or max. This system performed better than the original Network "C": 76% of the trials were within 99% of the optimal answer (see Fig. 3). This confirms the conjecture that projecting onto the feasible space helps the network converge closer to the optimal. (Technical note: The initial random vectors for these 200 trials, were normalized to unit length at the start; omitting this normalization appears to have no effect on the overall statistics but makes it a little bit more difficult to pick a reliable step size).

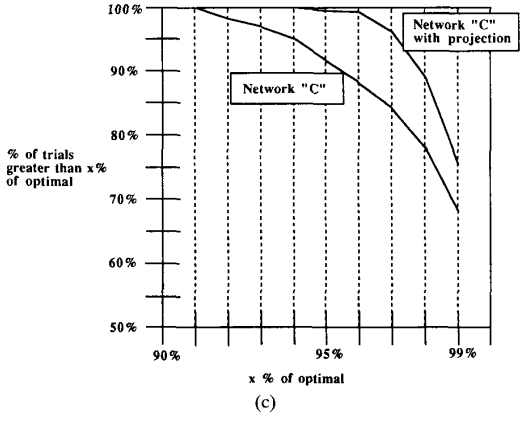
Experiment #3: The third data set again consists of 200 trials but this time we orthogonally project Network "A" onto the feasible subspace on every neural iteration. The addition of the projection to this network actually creates stable points that are inside the hypercube but directly associated with the original feasible states. That is, the network converges to states that have the correct value of min but do not reach



(a)



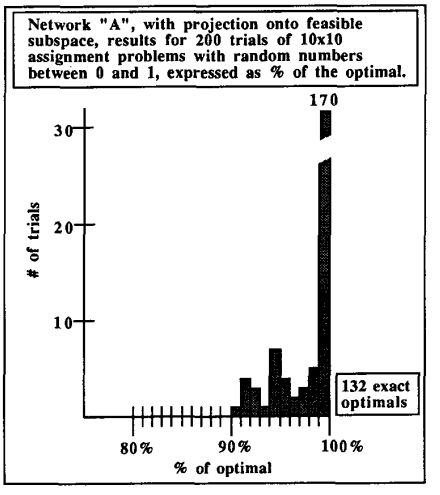
(b)



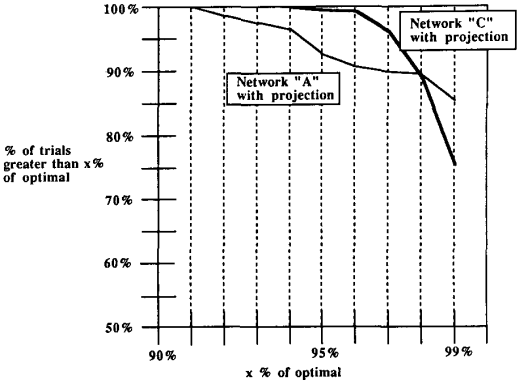
(c)

Fig. 3. Another 200 trials, comparing the results of running Network "C" as in the previous figure, with running Network "C" by projecting onto the feasible subspace on every iteration.

the full maximum value of $+1$ (they reached 0.65 for this $k = 10$ case). Fig. 4 shows the results of Network "A" with



(a)



(b)

Fig. 4. 200 trials running Network "A" by projecting onto the feasible subspace after each neural update. The lower part of the figure compares the analogous results for Network "C" with Network "A".

projection and a comparison with the results of Network "C" with projection.

It is not reasonable to apply the projection to Network "B" because $\min = 0$ forces *all* such projections to land outside of the n-cube. We tried several variations in addition to those described above and are confident that the data presented are "representative."

All of the experiments used $\underline{\text{ext}} = 0$. However, we did explore variations using the same external input to all units. For example, by analyzing the curves in Fig. 1, we see that they shift downward for a negative external input. In particular we tried the following network parameters: i) $\underline{\text{ext}} = -1$, $\min = -1/(k-1)$, $\alpha = 0$, $s = 0$; and ii) $\text{ext} = -1 - 1/(2(k-1))$, $\min = -1/(k-1)$, $\alpha = 1/(k-1)$, $s = 0$. These networks performed similar to Networks "A" and "B."

In additional experiments we added an external input to each unit that was proportional to the initial activation of that unit. As mentioned earlier, this should improve performance since better solutions now have lower energy. By analyzing the curves in Fig. 1 again, we see that a positive external input

will shift these curves upward, and if the inputs are too large the exclusive stability of feasible states comes into question. For Network "A" we see that a proportionality constant of $p_{\text{ext}} = +1/(2(k-1))$ puts the network right on the boundary of the feasible region, so we chose a safer setting: $p^a_{\text{ext}} = +1/(2k-1)$. This network performed slightly better than the original Network "A." For Network "B" we tried $p^b_{\text{ext}} = +1/2$, but in this case we had to reduce α to $1/(2(k-1))$ because the parameters of Network "B" are already on the boundary of the feasible region. Again, this network performed only slightly better than the original Network "B." For Network "C" we used $p^c_{\text{ext}} = +0.1/(k-1)$ and again the new network was only a slight improvement over the old.

Although adding external inputs proportional to the initial activations did not provide a significant improvement, it was noted that in *all* cases the addition of the orthogonal projection dramatically improved performance.

X. Conclusions

We have demonstrated how to choose the parameters of this type of network using eigenspaces and stability analysis. We confirmed most of our conjectures with statistical results. Unfortunately we do not arrive at 100% optimal performance. This must be attributed to a complex dynamic and geometric aspect of the interaction of the n-cube geometry and the update equations. In particular we originally desired convergence to the nearest feasible state, a requirement that can be analyzed by investigating the "boundaries" between basins of attraction for each feasible state. It appears that if these boundaries were invariant manifolds then the network would always pick the optimal choice, but this is not the case. The inherent nonlinearity of the update equations seems to be the underlying cause of the suboptimal results. The results can be improved slightly by using external inputs that are proportional to the initial activations, causing the optimal feasible state to be the lowest energy feasible state as well as the nearest. But if the external inputs are too large, nonfeasible states can become stable. In all cases, however, the introduction of the orthogonal projection onto the feasible subspace dramatically improved performance.

We have performed some simulations of the more traditional approach to this problem using the Hopfield-Tank model, and are confident that since our analysis is based on the underlying gradient descent, a common feature of both models, there will be similar statistics for that model.

Acknowledgment

The authors extend their thanks to the IEEE editors, as well as to D. Michaels and T. Altman of the University of Colorado, and J. Shaw of Hughes A. C., Denver, for their helpful technical comments. Thanks are also due to M. McCandless, J. Clark, and J. Thomas, of the University of Colorado at Denver, for their administrative support during this work.

References

[1] D. W. Tank and J. J. Hopfield, "Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 533–541, May 1986.

[2] J. J. Hopfield and D. W. Tank, "Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[3] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088–3092, May 1984.

[4] J. Ramanujam and P. Sadayuppan, "Optimization by neural networks," in *IEEE ICNN*, San Diego, CA, July 24–27, 1988, vol. II, pp. 325–332.

[5] R. D. Brandt, Y. Wang, A. Laub, and S. K. Mitra, "Alternative networks for solving the TSP and list-matching problem," in *IEEE ICNN*, San Diego, CA, July 24–27, 1988, vol. II, pp. 333–340.

[6] A. Rodriguez-Vasques, R. Dominguez-Castro, A. Rueda, J. Huertas, and E. Sanchez-Sinencio, "Nonlinear switched-capacitor 'neural' networks for optimization problems," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 384–397, Mar. 1990.

[7] D. E. Van den Bout and T. K. Miller," A travelling salesman objective function that works," in *IEEE ICNN*, San Diego, CA, July 24–27, 1988, vol. II, pp. 299–303.

[8] S. U. Hegde, J. L. Sweet, and W. B. Levy, "Determination of parameters in a Hopfield/Tank computational network," in *IEEE ICNN*, San Diego, CA, July 24–27, 1988, vol. II, pp. 291–298.

[9] L. O. Chau and G. N. Lin, "Nonlinear programming without computation," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 182–188, Feb. 1984.

[10] M. P. Kennedy and L. O. Chau, "Unifying the tank and Hopfield linear programming network and the canonical nonlinear programming circuit of Chau and Lin," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 210–214, Feb. 1987.

[11] M. P. Kennedy and L. O. Chau, "Unifying the tank and Hopfield linear programming network and the canonical nonlinear programming circuit of Chau and Lin," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 554–562, May 1988.

[12] J. L. McClelland and D. E. Rumelhart, "An interactive activation model of context effects in letter perception: Part 1. An account of basic findings," *Psych. Rev.*, vol. 88, pp. 375–407, 1981.

[13] S. V. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield Model," *IEEE Trans. Neural Networks*, vol. 1, pp. 204–215, June 1990.

[14] J. Li, A. N. Michel, and W. Porod, "Qualitative analysis and synthesis of a class of neural networks," *IEEE Trans. Circuits Syst.*, vol. 35, Aug. 1988.

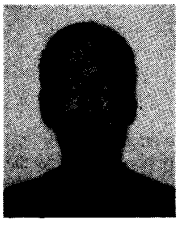
[15] J. Li, A. N. Michel, and W. Porod, "Analysis and synchysis of a class of neural networks: Variable structure systems with infinite gain," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 713–731, May 1989.

[16] J. Li, A. N. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1405–1422, Nov. 1989.

[17] W. J. Wolfe, D. Mathis, C. Anderson, J. Rothman, M. Gottler, G. Brady, R. Walker, G. Duane, and G. Alaghband, "K-winner networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 310–315, Mar. 1991.

[18] R. A. Horn and C. R. Johnson, Matrix analysis. Cambridge, U.K.: Cambridge University Press, 1985.

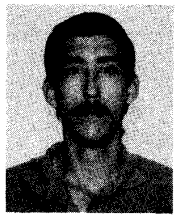
[19] A. V. Oppenheim and A. S. Wilsky, *Signals and Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1983.



William J. Wolfe (M'82) received the Ph.D. degree in mathematics from the City University of New York in 1976.

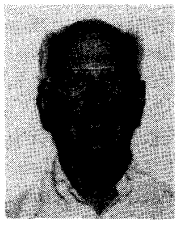
He worked for several years in the robotics and artificial intelligence groups at Martin Marietta in Denver.

He is currently an associate professor of computer science at the University of Colorado at Denver where he teaches graduate courses in artificial intelligence and neural networks. He has also taught several industrial courses in robotics and computer vision and has cochaired the SPIE Mobile Robots conference for the past five years with Wendell H. Chun.



James M. MacMillan received the B.S. degree in electrical engineering from the University of Nebraska at Lincoln in May 1980, and the M.S. degree in electrical engineering from the University of Colorado at Denver in May 1992.

For the past two years, he has researched the application of neural networks to optimization problems, particularly scheduling problems. His effort culminated in his Master's thesis, "Solving the task scheduling problem using neural networks," which described a novel network architecture to solve the knapsack problem. Although far from completing his research into scheduling problems, his interests have expanded to include the application of neural networks to consumer electronic products. After serving four years in the U.S. Army, he joined Hughes Aircraft Company in 1984. Still employed with Hughes, he is responsible for hardware design of satellite ground station equipment.



George Brady was born on February 15, 1939 in San Francisco, California. He graduated from San Jose State College with the B.S. degree in geology in 1969 and from the University of Colorado, Denver, with the M.S. degree in computer science in 1992. His interests include artificial intelligence, neural networks, and software reliability.

Robert Mathews is a Ph.D. student in mathematics at the University of Colorado at Denver.

He has had 20 years of software development experience working for several aerospace companies.

Jay Alan Rothman received a Ph.D. degree in mathematics from the Courant Institute of Mathematical Sciences, New York University, in 1974, and is currently an assistant professor of Computer Science at the University of Colorado at Denver where he teaches a wide variety of graduate level computer science classes.

Donald Mathis is a Ph.D. student in computer science at the University of Colorado at Boulder, where he works with Professor Michael Mozer. He received the B.S. degree in computer science from Carnegie Mellon University, Pittsburg, PA, in 1985.

He worked on a variety of artificial intelligence projects at Martin Marietta from 1985–1988.

Michael Donald Orosz received the M.S. degree in computer science from the University of Colorado at Denver, 1991, where he researched neural networks.

He is currently a software engineer at TRW in Aurora, Colorado.

Charlie Anderson after 20 years of teaching high school in the Denver area, is currently finishing the Ph.D. degree in applied mathematics at the University of Colorado at Denver.

His current research interests include optimization heuristics and competition graph theory, and he has coauthored papers on genetic algorithms and niche graphs.

Gita Alaghband received the B.S. degree in physics in 1977 form Tehran University, the M.S. degree in computer science in 1980, and the Ph.D. degree in electrical engineering in 1986, all from the University of Colorado, Boulder.

Currently she is an assistant professor in the Department of Electrical Engineering and Computer Science at the University of Colorado, Denver. She was also a visiting assistant professor at Rensselaer Polytechnic Institute (R.P.I.) in 1987. Her research interests in parallel processing includes, computer architecture, performance evaluation, simulation, application programs, and algorithm design. Dr. Alaghband is a member of the IEEE Computer Society.