ALTERNATIVE ENERGY FUNCTIONS FOR OPTIMIZING NEURAL NETWORKS

A. H. Gee & R. W. Prager

CUED/F-INFENG/TR 95

March 1992

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Email: ahg/rwp @eng.cam.ac.uk

Alternative Energy Functions for Optimizing Neural Networks

Andrew H. Gee and Richard W. Prager [1]

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

March 1992

Abstract

When feedback neural networks are used to solve combinatorial optimization problems, their dynamics perform some sort of descent on a continuous energy function related to the objective of the discrete problem. For any particular discrete problem, there are generally a number of suitable continuous energy functions, and the performance of the network can be expected to depend heavily on the choice of such a function. In this paper, alternative energy functions are employed to modify the dynamics of the network in a predictable manner, and progress is made towards identifying which are well suited to the underlying discrete problems. This is based on a revealing study of a large database of solved problems, in which the optimal solutions are decomposed along the eigenvectors of the network's connection matrix. It is demonstrated that there is a strong correlation between the mean and variance of this decomposition and the ability of the network to find good solutions. A consequence of this is that there may be some problems which neural networks are not well adapted to solve, irrespective of the manner in which the problems are mapped onto the network for solution.

1 Introduction and Outline

Ever since the Hopfield network was first proposed as a means of solving combinatorial optimization problems [15], researchers have frequently reported disappointment at the failure of the network to find valid solutions, let alone high quality ones [16, 24]. More recently, alternative mappings and modified dynamics have been developed which largely overcome the validity difficulties [3, 20, 23], and neural techniques have been applied with some success to the solution of a wide variety of problems.

Notably successful have been the self-organizing networks, which can be applied to problems with a suitable underlying geometric structure [4, 7, 9], though the neural techniques with the widest applicability are based around the original Hopfield network approach. These techniques invariably perform some kind of descent on a continuous energy function, which coincides with the objective of the combinatorial problem at the valid solution points in the network's output space. Neural techniques, like the Hopfield network or the Mean Field Annealing (MFA) algorithm, perform this descent within a highly parallel framework, which naturally suggests extremely fast, parallel implementations. Their dynamics dictate that the continuous energy function be quadratic in the network's state vector, $\mathbf{v}$. Even with this limitation, there are many problems which can be mapped onto this system, including the benchmark travelling salesman [3, 1, 20] and graph partitioning problems [1, 20, 23], along with more useful problems of two-graph matching [10, 11], scheduling [12], load balancing [8], analogue to digital conversion [22], and a novel implementation of the Viterbi algorithm for Hidden Markov Models [2].

An issue which has aroused little discussion so far concerns the choice of the continuous energy function on which the network performs its descent. For many problems there are an infinite number of quadratic functions which satisfy the condition that they coincide with the combinatorial

---

[1] Email: ahg/rwp @eng.cam.ac.uk

objective at the valid solution points. While transformations of objective functions have been previously investigated as a means of reducing network costs and mapping a wider variety of problems [18], little attention has been drawn to how the use of these alternative energy functions may affect the solution quality. This issue is addressed in this paper, with specific application to the two-graph matching problem, of which the travelling salesman problem (TSP) is a special case.

In Section 2 we introduce the notation in use throughout the paper, and define some frequently used matrices. Section 3 reviews much of the background to the subject, including the Hopfield and MFA dynamics, and the notion of a valid subspace on which all the valid solution points lie. In Section 4 we recast the entire scenario relative to the basis formed by the eigenvectors of the network's connection matrix; the use of this basis highlights many features of both the network's dynamics and good solutions which would otherwise remain obscured. Section 5 contains a detailed analysis of the network's dynamics relative to this new basis, while Section 6 examines the effect on these dynamics of the various annealing techniques which are often used to improve the network's performance. Included in Sections 4, 5 and 6 are many results taken from [1], which should be referred to for proofs. In Section 7 we introduce a set of alternative energy functions which are all suitable for the solution of the two-graph matching problem, and examine how the use of these functions, coupled with the usual annealing processes, affects aspects of the network's dynamics. Section 8 draws on large databases of solved Euclidean and random TSPs to examine properties of optimal solutions relative to the network's eigenvector basis (by a *random* TSP, we mean a TSP for which the distance matrix does not correspond to any set of points in a 2D plane, but is an arbitrary, symmetric square matrix). The results suggest how a suitable energy function may be selected from among the alternatives for the Euclidean TSPs, but that none of these functions are well suited for the solution of the random problems. A series of experiments, using both the Hopfield and MFA dynamics on the problem databases, confirms these predictions. Finally, in Section 10 we present a discussion of the issues raised in the paper, and draw our conclusions.

2 Notation and Definitions

2.1 Matrix Notation

Let $\mathbf{A}^T$ denote the transpose of $\mathbf{A}$.
Let $[\mathbf{A}]_{ij}$ refer to the element in the i^{th} row and j^{th} column of the matrix $\mathbf{A}$.
Similarly, let $[\mathbf{a}]_i$ refer to the i^{th} element of the vector $\mathbf{a}$.
Sometimes, where the above notation would appear clumsy, and there is no danger of ambiguity, the same elements will be alternatively denoted A_{ij} and a_i.
Let the modulus of an $n \times m$ matrix $\mathbf{A}$ be defined as follows:

$$||\mathbf{A}||^2 = \left[\sum_{i=1}^{n}\sum_{j=1}^{m}[\mathbf{A}]_{ij}^2\right] = \text{trace}\left(\mathbf{A}^T\mathbf{A}\right) \tag{1}$$

Let $\mathbf{A} \otimes \mathbf{B}$ denote the Kronecker product of two matrices. If $\mathbf{A}$ is an $n \times n$ matrix, and $\mathbf{B}$ is an $m \times m$ matrix, then $\mathbf{A} \otimes \mathbf{B}$ is an $nm \times nm$ matrix given by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{11}\mathbf{B} & A_{12}\mathbf{B} & \dots & A_{1n}\mathbf{B} \\ A_{21}\mathbf{B} & A_{22}\mathbf{B} & \dots & A_{2n}\mathbf{B} \\ \dots & \dots & \dots & \dots \\ A_{n1}\mathbf{B} & A_{n2}\mathbf{B} & \dots & A_{nn}\mathbf{B} \end{bmatrix} \tag{2}$$

Let vec($\mathbf{A}$) be the function which maps the $n \times m$ matrix $\mathbf{A}$ onto the nm-element vector $\mathbf{a}$. This function is defined by:

$$\mathbf{a} = \text{vec}(\mathbf{A}) = [A_{11}, A_{21}, \dots, A_{n1}, A_{12}, A_{22}, \dots, A_{n2}, \dots, A_{1m}, A_{2m}, \dots, A_{nm}]^T \tag{3}$$

The following properties of Kronecker products are used in the course of this paper (see [13] for proofs):

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T \tag{4}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{X} \otimes \mathbf{Y}) = (\mathbf{AX} \otimes \mathbf{BY}) \tag{5}$$

If the $n \times n$ matrix $\mathbf{A}$ has eigenvectors $\{\mathbf{w}_1 \ldots \mathbf{w}_n\}$ with corresponding eigenvalues $\{\gamma_1 \ldots \gamma_n\}$, and the $m \times m$ matrix $\mathbf{B}$ has eigenvectors $\{\mathbf{h}_1 \ldots \mathbf{h}_m\}$ with corresponding eigenvalues $\{\mu_1 \ldots \mu_m\}$, then the $nm \times nm$ matrix $\mathbf{A} \otimes \mathbf{B}$ has eigenvectors $\{\mathbf{w}_i \otimes \mathbf{h}_j\}$ with corresponding eigenvalues $\{\gamma_i \mu_j\}$ $(i \in \{1 \ldots n\}, j \in \{1 \ldots m\})$.

2.2 Some Definitions

Let $\mathbf{I}^n$ be the $n \times n$ identity matrix.
Let $\mathbf{o}^n$ be the n-element column vector of ones:

$$[\mathbf{o}^n]_i = 1 \quad (i \in \{1, \ldots, n\}) \tag{6}$$

Let $\mathbf{O}^n$ be the $n \times n$ matrix of ones:

$$[\mathbf{O}^n]_{ij} = 1 \quad (i, j \in \{1, \ldots, n\}) \tag{7}$$

Similarly, let $\mathbf{O}^{nm}$ be the $n \times m$ matrix of ones.
Let δ_{ij} be the Kronecker impulse function:

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases} \tag{8}$$

Let $\mathbf{R}^n$ be the $n \times n$ matrix given by

$$\mathbf{R}^n = \mathbf{I}^n - \tfrac{1}{n}\mathbf{O}^n \tag{9}$$

Multiplication by $\mathbf{R}^n$ has the effect of setting the column sums of a matrix to zero:

$$\begin{aligned} \sum_{i=1}^{n} [\mathbf{R}^n \mathbf{a}]_i &= \sum_{i=1}^{n} [\mathbf{a} - \tfrac{1}{n}\mathbf{O}^n \mathbf{a}]_i = \mathbf{o}^{nT}[\mathbf{a} - \tfrac{1}{n}\mathbf{O}^n \mathbf{a}] \\ &= \mathbf{o}^{nT}\mathbf{a} - \tfrac{1}{n}(\mathbf{o}^{nT}\mathbf{o}^n\mathbf{o}^{nT})\mathbf{a} = \mathbf{o}^{nT}\mathbf{a} - \mathbf{o}^{nT}\mathbf{a} = 0 \end{aligned} \tag{10}$$

Another way of considering $\mathbf{R}^n$ is as a projection matrix which removes the $\mathbf{o}^n$ component from any vector it pre-multiplies, since

$$\mathbf{R}^n \mathbf{o}^n = (\mathbf{I}^n - \tfrac{1}{n}\mathbf{O}^n)\mathbf{o}^n = \mathbf{o}^n - \tfrac{1}{n} n \mathbf{o}^n = \mathbf{0} \tag{11}$$

Also note that since $\mathbf{R}^n$ is a projection matrix, $\mathbf{R}^n \mathbf{R}^n = \mathbf{R}^n$.

3 Background

3.1 The Hopfield Network

A schematic diagram of the continuous Hopfield network [14] is shown in Figure 1. Neuron i has input $[\mathbf{u}]_i$, output $[\mathbf{v}]_i$, and is connected to neuron j with a weight $[\mathbf{T}]_{ij}$. Associated with each neuron is also an input bias term $[\mathbf{i}^{\mathrm{b}}]_i$. The dynamics of the network are governed by the following equations:

$$\dot{\mathbf{u}} = -\eta \mathbf{u} + \mathbf{T}\mathbf{v} + \mathbf{i}^{\mathrm{b}} \tag{12}$$

$$[\mathbf{v}]_i = g([\mathbf{u}]_i) \tag{13}$$

$[\mathbf{v}]_i$ is a continuous variable in the interval 0 to 1, and $g([\mathbf{u}]_i)$ is a monotonically increasing function which constrains $[\mathbf{v}]_i$ to this interval, usually a hyperbolic tangent of the form

$$g([\mathbf{u}]_i) = \frac{1}{1 + \exp(-[\mathbf{u}]_i / T^p)} \tag{14}$$

The network has a Liapunov function [14]

$$E = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}\mathbf{v} - (\mathbf{i}^{\mathrm{b}})^T\mathbf{v} + \eta\sum\int_0^{[\mathbf{v}]_i} g^{-1}(V)\,dV \tag{15}$$

The network was subsequently proposed as a means of solving combinatorial optimization problems which can somehow be expressed as the constrained minimization of

$$E^{\mathrm{op}} = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{op}}\mathbf{v} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v} \quad ([\mathbf{v}]_i \in \{0,1\}) \tag{16}$$

The idea is that the network's Liapunov function, invariably with $\eta = 0$, is associated with the cost function to be minimized in the combinatorial optimization problem. The network is then run and allowed to converge to a hypercube corner, which is subsequently interpreted as the solution of the problem. Hopfield and Tank showed in [15] how the network output can be used to represent a solution to the travelling salesman problem, and how the interconnection weights and input biases can be programmed appropriately for that problem: this process has subsequently been termed 'mapping' the problem onto the network. The same authors later showed how a variety of other problems can be mapped onto the same network, and reported on the results of using analogue hardware implementations to solve the problems [22].

The difficulties with mapping problems onto the Hopfield network lie with the satisfaction of hard constraints. The network acts to minimize a single Liapunov function, and yet the typical combinatorial optimization problem requires the minimization of a function *subject to* a number of constraints: if any of these constraints are violated then the solution is termed 'invalid'. The early mapping techniques coded the validity constraints as terms in the Liapunov function which were minimized when the constraints were satisfied:

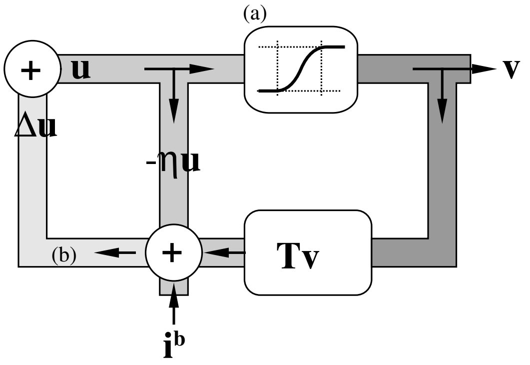
$$E = E^{\mathrm{op}} + c_1 {E^{\mathrm{cns}}}_1 + c_2 {E^{\mathrm{cns}}}_2 + \ldots \tag{17}$$

The c_i parameters in equation (17) are constant weightings given to the various energy terms. The multiplicity of terms in the Liapunov function tend to frustrate one another, and the success of the network is highly sensitive to the relative values of the c_i parameters; it is not surprising, therefore, that the network frequently found invalid solutions, let alone high quality ones [16, 24].

3.2 The Valid Subspace

In [3] an eigenvector and subspace analysis of the network's behaviour revealed how the E^{op} and E^{cns} terms in equation (17) can be effectively decoupled into different subspaces so that they no longer frustrate one another. For a wide variety of problems, it was realized that all the hypercube corners corresponding to valid solutions lie on a particular affine subspace with equation

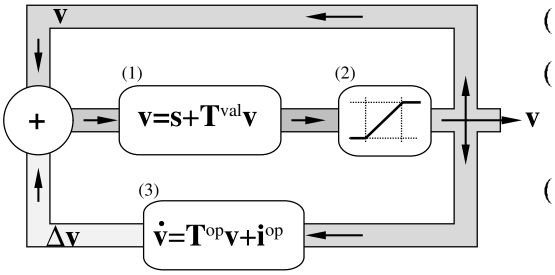
$$\mathbf{v} = \mathbf{T}^{\mathrm{val}}\mathbf{v} + \mathbf{s} \tag{18}$$



(a) Non-linear threshold functions constraining **v** to the unit hypercube.

(b) The change in **u** is specified by the differential equation (12).

Figure 1: Schematic diagram of the continuous Hopfield network.



(1) Projection of $\mathbf{v}$ onto valid subspace

(2) Nonlinear 'symmetric ramp' threshold functions constraining $\mathbf{v}$ to the unit hypercube.

(3) Change in $\mathbf{v}$ given by the gradient of the optimization energy term E^{op}, with $\Delta\mathbf{v} = \Delta t\dot{\mathbf{v}}$

Figure 2: Schematic diagram of the modified network implementation.

where $\mathbf{T}^{\mathrm{val}}$ is a projection matrix (ie. $\mathbf{T}^{\mathrm{val}}\mathbf{T}^{\mathrm{val}} = \mathbf{T}^{\mathrm{val}}$) and $\mathbf{T}^{\mathrm{val}}\mathbf{s} = \mathbf{0}$. This subspace was termed the **valid subspace**. The constrained optimization can now be re-expressed using only a single constraint term [1]:

$$E = E^{\mathrm{op}} + \tfrac{1}{2}c_0 \left\| \mathbf{v} - (\mathbf{T}^{\mathrm{val}}\mathbf{v} + \mathbf{s}) \right\|^2 \tag{19}$$

Expanding equation (19) we obtain, to within a constant

$$E = E^{\mathrm{op}} - c_0 \left(\tfrac{1}{2}\mathbf{v}^T(\mathbf{T}^{\mathrm{val}} - \mathbf{I})\mathbf{v} + \mathbf{s}^T\mathbf{v} \right) \tag{20}$$

from which we see that the Hopfield network parameters must be set as follows:

$$\mathbf{T} = \mathbf{T}^{\mathrm{op}} + c_0(\mathbf{T}^{\mathrm{val}} - \mathbf{I}) \tag{21}$$

$$\mathbf{i}^{\mathrm{b}} = \mathbf{i}^{\mathrm{op}} + c_0\mathbf{s} \tag{22}$$

In the limit of large c_0, $\mathbf{v}$ will be pinned to the valid subspace throughout convergence, and the network dynamics will minimize $E = E^{\mathrm{op}}$ as required. This approach suggests a way in which the Hopfield network may be approximately simulated with high efficiency and without the need for penalty terms — see Figure 2. In addition to steepest descent dynamics, we see an extra loop which continually projects $\mathbf{v}$ onto the valid subspace, directly enforcing the validity constraints. Piecewise linear transfer functions are employed to keep $\mathbf{v}$ within the unit hypercube, while allowing elements of $\mathbf{v}$ to fully saturate at 0 or 1. The descent dynamics are now free to minimize E^{op} alone, without the need for penalty functions. This model has been successful in solving the benchmark travelling salesman problem [1], along with the more useful problems of two-graph matching for invariant pattern recognition [10, 11] and implementing the Viterbi algorithm for Hidden Markov Models [1, 2]: we shall use it throughout this paper to approximately simulate the behaviour of the Hopfield network.

3.3 Mean Field Annealing

While the mapping of problems is generally easier to visualize with $\eta = 0$, in practice the Hopfield network dynamics may usually be implemented with any value of η, so long as all hypercube corners corresponding to valid solutions lie at the same distance from the origin. This is because the η term in the network Liapunov function (15) has the same value at all such hypercube corners, so the relative energy of all valid hypercube corners is unchanged whatever the value of η. If we choose to simulate the Hopfield network dynamics using an Euler approximation with a time step of Δt, we obtain

$$\mathbf{u}_{(k+1)\Delta t} = \mathbf{u}_{k\Delta t} + \Delta t(-\eta\mathbf{u}_{k\Delta t} + \mathbf{T}\mathbf{v}_{k\Delta t} + \mathbf{i}^{\mathrm{b}}) \tag{23}$$

Furthermore, if we choose $\Delta t = 1$ and $\eta = 1$, then the Euler approximation becomes

$$\mathbf{u}_{(k+1)\Delta t} = \mathbf{T}\mathbf{v}_{k\Delta t} + \mathbf{i}^{\mathrm{b}} \tag{24}$$

This equation, together with (14), describes the update rule for the Mean Field Annealing (MFA) technique for solving combinatorial optimization problems. It is possible to derive the MFA equations as an approximation of simulated annealing [1, 20, 21], though for the purposes of analyzing MFA and the Hopfield network from a common standpoint, they are better viewed as an Euler approximation of the Hopfield network dynamics with $\Delta t = 1$ and $\eta = 1$. Viewing MFA in this way also allows us to assess the detrimental effect of the approximation which lies between MFA and simulated annealing, since the latter is guaranteed to find the optimal solution in the (impractical) limit of an infinitely slow cooling schedule. The MFA equations can also be derived in a way that makes the satisfaction of one constraint explicit within the update rule [1, 20, 23], allowing many problems to be mapped so that one of the constraints is enforced without the need for a penalty term. This approach, which we shall refer to as MFA with neuron normalization, has been successful when solving the benchmark travelling salesman [20] and graph partitioning [20, 23] problems, along with some more practical scheduling [12] and load balancing [8] problems.

3.4 Mapping some common problems

Throughout this paper we shall consider a class of combinatorial optimization problem which can be formulated as

$$\text{minimize} \quad E^{\mathrm{op}} = -\tfrac{1}{2}\,\mathrm{trace}\left(\mathbf{V}^T\mathbf{P}\mathbf{V}\mathbf{Q}\right) \tag{25}$$

$$\text{subject to} \quad [\mathbf{V}]_{ij} \in \{0,1\} \tag{26}$$

$$\text{and} \quad \mathbf{V} = \mathbf{R}^n\mathbf{V}\mathbf{R}^m + \tfrac{1}{m}\mathbf{O}^{nm} \tag{27}$$

where $\mathbf{V}$ is an $n \times m$ matrix, $\mathbf{P}$ is a symmetric $n \times n$ matrix and $\mathbf{Q}$ is a symmetric $m \times m$ matrix. For the case $n = m$, the conditions in equations (26) and (27) ensure that $\mathbf{V}$ takes the form of a valid permutation matrix, suitable for representing the solutions to the travelling salesman, Hamilton path and two-graph matching problems. For $n > m$, the framework can be used to solve the graph partitioning problem, where a graph of n nodes is to be divided into m equally sized partitions. For such mappings it is necessary that the ratio of n to m is an integer. The solution is represented by $\mathbf{V}$ in the following manner: $[\mathbf{V}]_{ij}$ equals 1 if node i is to be in partition j, and 0 otherwise. The conditions in equations (26) and (27) now ensure that a node is associated with only one partition, and that the partitions are of equal size. Table 1 shows how the $\mathbf{P}$ and $\mathbf{Q}$ matrices can be set so that the minimization of the objective function (25) solves the required problem: for a fuller description of these mappings, see [1, 10]. It should be noted that the travelling salesman and Hamilton path problems are merely special cases of the two-graph matching problem.

Such problems may be readily mapped onto the Hopfield network for solution as follows [1, 10]:

$$\mathbf{v} = \mathrm{vec}(\mathbf{V}^T) \tag{28}$$

$$\mathbf{T}^{\mathrm{op}} = \mathbf{P} \otimes \mathbf{Q} \tag{29}$$

$$\mathbf{i}^{\mathrm{op}} = \mathbf{0} \tag{30}$$

$$\mathbf{T}^{\mathrm{val}} = \mathbf{R}^n \otimes \mathbf{R}^m \tag{31}$$

$$\mathbf{s} = \tfrac{1}{m}(\mathbf{o}^n \otimes \mathbf{o}^m) \tag{32}$$

If the network parameters are set as described above, then the network's objective function (16) corresponds to the problem's cost function (25), whilst confinement to the valid subspace (18) ensures satisfaction of the problem constraint (27). If the network is forced to converge to a hypercube corner, then the problem constraint (26) is also satisfied.

3.5 The effective objective function

If it is assumed that $\mathbf{v}$ is continually confined to the valid subspace, so that $\mathbf{v} = \mathbf{T}^{\mathrm{val}}\mathbf{v} + \mathbf{s}$, it is possible to express E^{op} as follows:

$$E^{\mathrm{op}} = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{val}}\mathbf{T}^{\mathrm{op}}\mathbf{T}^{\mathrm{val}}\mathbf{v} - \left(\mathbf{T}^{\mathrm{val}}(\mathbf{T}^{\mathrm{op}}\mathbf{s} + \mathbf{i}^{\mathrm{op}})\right)^T\mathbf{v} + \text{ terms independent of } \mathbf{v}$$

Problem	$[\mathbf{P}]_{xy}$	$[\mathbf{Q}]_{ij}$	n	m
Two-Graph Matching	e^p_{xy}	e^q_{ij}	Number of nodes	Number of nodes
TSP	$-d_{xy}$	$\delta_{i,j\oplus 1} + \delta_{i,j\ominus 1}$	Number of cities	Number of cities
Hamilton Path	$-e_{xy}$	$\delta_{i,j+1} + \delta_{i,j-1}$	Number of nodes	Number of nodes
GPP	e_{xy}	δ_{ij}	Number of nodes	Number of partitions

e_{xy} is the edge weight between nodes x and y
d_{xy} is the distance between cities x and y
$j \oplus 1 = j + 1$ except $m \oplus 1 = 1$
$j \ominus 1 = j - 1$ except $1 \ominus 1 = m$

Table 1: Some possible problem mappings.

Hence, for $\mathbf{v}$ confined to the valid subspace, the objective function becomes

$$
\begin{aligned}
E^{\mathrm{opr}} &= -\tfrac{1}{2}\mathbf{v}^T \mathbf{T}^{\mathrm{opr}} \mathbf{v} - (\mathbf{i}^{\mathrm{opr}})^T \mathbf{v} && (33)\\
\text{where } \mathbf{T}^{\mathrm{opr}} &= \mathbf{T}^{\mathrm{val}} \mathbf{T}^{\mathrm{op}} \mathbf{T}^{\mathrm{val}} && (34)\\
\text{and } \mathbf{i}^{\mathrm{opr}} &= \mathbf{T}^{\mathrm{val}} (\mathbf{T}^{\mathrm{op}} \mathbf{s} + \mathbf{i}^{\mathrm{op}}) && (35)
\end{aligned}
$$

It is E^{opr} which the network effectively minimizes, so long as the evolving state vector $\mathbf{v}$ remains pinned to the valid subspace. For the mappings of Section 3.4, $\mathbf{T}^{\mathrm{opr}}$ may be expressed as follows:

$$
\begin{aligned}
\mathbf{T}^{\mathrm{opr}} &= \mathbf{P}^{\mathrm{r}} \otimes \mathbf{Q}^{\mathrm{r}} && (36)\\
\text{where } \mathbf{P}^{\mathrm{r}} &= \mathbf{R}^n \mathbf{P} \mathbf{R}^n && (37)\\
\text{and } \mathbf{Q}^{\mathrm{r}} &= \mathbf{R}^m \mathbf{Q} \mathbf{R}^m && (38)
\end{aligned}
$$

Likewise, $\mathbf{i}^{\mathrm{opr}}$ becomes

$$
\mathbf{i}^{\mathrm{opr}} = \tfrac{1}{m} (\mathbf{R}^n \mathbf{P} \mathbf{o}^n \otimes \mathbf{R}^m \mathbf{Q} \mathbf{o}^m) \qquad (39)
$$

4 An alternative set of basis vectors for network analysis

In this section we shall examine the eigenvectors and eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ in some detail, since they shall emerge as the driving force behind the network dynamics. Subsequently, we shall use them as a set of basis vectors for any further analysis, since in so doing various features of the optimization problems are naturally cast in the context of the network dynamics.

We shall start our examination by considering the eigenvectors and eigenvalues of $\mathbf{P}^{\mathrm{r}}$ and $\mathbf{Q}^{\mathrm{r}}$. The null spaces of $\mathbf{P}^{\mathrm{r}}$ and $\mathbf{Q}^{\mathrm{r}}$ are readily apparent since

$$
\begin{aligned}
\mathbf{P}^{\mathrm{r}} \mathbf{o}^n &= \mathbf{R}^n \mathbf{P} \mathbf{R}^n \mathbf{o}^n = \mathbf{0}\\
\text{and } \mathbf{Q}^{\mathrm{r}} \mathbf{o}^m &= \mathbf{R}^m \mathbf{Q} \mathbf{R}^m \mathbf{o}^m = \mathbf{0}
\end{aligned}
$$

So let us define $\mathbf{w}^1$ and $\mathbf{h}^1$ to be eigenvectors of $\mathbf{P}^{\mathrm{r}}$ and $\mathbf{Q}^{\mathrm{r}}$ respectively, with associated eigenvalues λ_1 and γ_1, where

$$
\begin{aligned}
\mathbf{w}^1 &= \tfrac{1}{\sqrt{n}} \mathbf{o}^n \quad \text{and} \quad \lambda_1 = 0 && (40)\\
\mathbf{h}^1 &= \tfrac{1}{\sqrt{m}} \mathbf{o}^m \quad \text{and} \quad \gamma_1 = 0 && (41)
\end{aligned}
$$

Let the remaining eigenvectors of $\mathbf{P}^{\mathrm{r}}$ be $\mathbf{w}^2 \ldots \mathbf{w}^n$ with eigenvalues $\lambda_2 \ldots \lambda_n$, and those of $\mathbf{Q}^{\mathrm{r}}$ be $\mathbf{h}^2 \ldots \mathbf{h}^m$ with eigenvalues $\gamma_2 \ldots \gamma_m$. Let us further assume that the eigenvalues are ordered as follows:

$$\lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \tag{42}$$

$$\gamma_2 \geq \gamma_3 \geq \ldots \geq \gamma_m \tag{43}$$

The result in Section 2 concerning the eigenvectors and eigenvalues of Kronecker product matrices allows us to express the eigenvectors and eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ in terms of those of $\mathbf{P}^{\mathrm{r}}$ and $\mathbf{Q}^{\mathrm{r}}$. If $\mathbf{T}^{\mathrm{opr}}$ has eigenvectors and eigenvalues $\mathbf{x}^{kl}$ and χ_{kl} respectively, then

$$\mathbf{x}^{kl} = \mathbf{w}^k \otimes \mathbf{h}^l \tag{44}$$

$$\text{and} \quad \chi_{kl} = \lambda_k \gamma_l \tag{45}$$

for $1 \leq k \leq n$ and $1 \leq l \leq m$. Note that

$$\begin{aligned} \mathbf{T}^{\mathrm{val}}\mathbf{x}^{kl} &= (\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{w}^k \otimes \mathbf{h}^l) \\ &= (\mathbf{R}^n \mathbf{w}^k \otimes \mathbf{R}^m \mathbf{h}^l) \\ \Rightarrow \mathbf{T}^{\mathrm{val}}\mathbf{x}^{kl} &= \begin{cases} \mathbf{0} & \text{for } k = 1 \text{ or } l = 1 \\ \mathbf{x}^{kl} & \text{for } k \geq 2 \text{ and } l \geq 2 \end{cases} \end{aligned} \tag{46}$$

Hence those $\mathbf{x}^{kl}$ for which $k = 1$ or $l = 1$ are eigenvectors of $\mathbf{T}^{\mathrm{opr}}$ lying outside the valid subspace, while those $\mathbf{x}^{kl}$ for which $k \geq 2$ and $l \geq 2$ lie within the valid subspace.

4.1 The eigenvector component matrices A and B

As has already been pointed out, the eigenvectors and eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ will emerge as the driving force behind the network dynamics. It will therefore be very useful to examine the decompositions of $\mathbf{v}$ and $\mathbf{i}^{\mathrm{opr}}$ along these eigenvectors. We shall express the decompositions in terms of matrices $\mathbf{A}$ and $\mathbf{B}$, such that

$$\mathbf{v} = \sum_{k=1}^{n}\sum_{l=1}^{m} [\mathbf{A}]_{kl}\mathbf{x}^{kl} = \sum_{k=1}^{n}\sum_{l=1}^{m} [\mathbf{A}]_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \tag{47}$$

$$\mathbf{i}^{\mathrm{opr}} = \sum_{k=1}^{n}\sum_{l=1}^{m} [\mathbf{B}]_{kl}\mathbf{x}^{kl} = \sum_{k=1}^{n}\sum_{l=1}^{m} [\mathbf{B}]_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \tag{48}$$

Referring to equation (46), we see that the first row and column of $\mathbf{A}$ correspond to components of $\mathbf{v}$ outside the valid subspace, so the component of $\mathbf{v}$ which lies within the valid subspace may be expressed as

$$\mathbf{v}^{\mathrm{val}} = \mathbf{T}^{\mathrm{val}}\mathbf{v} = \sum_{k=2}^{n}\sum_{l=2}^{m} [\mathbf{A}]_{kl}\mathbf{x}^{kl} = \sum_{k=2}^{n}\sum_{l=2}^{m} [\mathbf{A}]_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \tag{49}$$

If it is assumed that $\mathbf{v}$ is continually confined to the valid subspace, then the elements in the first row and column of $\mathbf{A}$ are fixed as follows:

$$\begin{aligned} [\mathbf{A}]_{kl} = \mathbf{v}^T\mathbf{x}^{kl} &= (\mathbf{T}^{\mathrm{val}}\mathbf{v} + \mathbf{s})^T(\mathbf{w}^k \otimes \mathbf{h}^l) \\ &= \tfrac{1}{m}(\mathbf{o}^n \otimes \mathbf{o}^m)^T(\mathbf{w}^k \otimes \mathbf{h}^l) + \mathbf{v}^T(\mathbf{R}^n\mathbf{w}^k \otimes \mathbf{R}^m\mathbf{h}^l) \\ &= \tfrac{1}{m}(\mathbf{o}^{nT}\mathbf{w}^k \otimes \mathbf{o}^{mT}\mathbf{h}^l) \quad \text{for } k = 1 \text{ or } l = 1 \\ \Rightarrow [\mathbf{A}]_{kl} &= \begin{cases} \sqrt{\frac{n}{m}} & \text{if } k = l = 1 \\ 0 & \text{for } k = 1 \text{ and } l \geq 2, \text{ or } l = 1 \text{ and } k \geq 2 \end{cases} \end{aligned} \tag{50}$$

Hence $[\mathbf{A}]_{11} = \sqrt{n/m}$, while all the other elements in the first row and column of $\mathbf{A}$ are zero. Considering now the matrix $\mathbf{B}$, we see that

$$\begin{aligned} [\mathbf{B}]_{kl} = \mathbf{i}^{\mathrm{opr}T}\mathbf{x}^{kl} &= \left(\mathbf{T}^{\mathrm{val}}(\mathbf{T}^{\mathrm{op}}\mathbf{s} + \mathbf{i}^{\mathrm{op}})\right)^T \mathbf{x}^{kl} \\ &= (\mathbf{T}^{\mathrm{op}}\mathbf{s} + \mathbf{i}^{\mathrm{op}})^T \mathbf{T}^{\mathrm{val}}\mathbf{x}^{kl} \\ \Rightarrow [\mathbf{B}]_{kl} &= 0 \quad \text{for } k = 1 \text{ or } l = 1 \end{aligned} \tag{51}$$

Thus the elements in the first row and column of $\mathbf{B}$ are zero.

4.2 Eigenvalue degeneracy

For many problems it transpires that the eigenvalues of $\mathbf{P}^{\mathrm{r}}$ or $\mathbf{Q}^{\mathrm{r}}$ are degenerate; this degeneracy propagates through to the eigenvalues of $\mathbf{T}^{\mathrm{opr}}$. As an example, let us consider the $\mathbf{Q}$ matrix for the travelling salesman problem (see Table 1). The (unordered) eigenvalues of the corresponding $\mathbf{Q}^{\mathrm{r}}$ matrix are given by [1]

$$\gamma_l = \begin{cases} 0 & \text{for } l = 1 \\ 2\cos\left(\frac{2\pi}{n}(l-1)\right) & \text{for } l \geq 2 \end{cases} \tag{52}$$

It follows that many of the eigenvalues of $\mathbf{Q}^{\mathrm{r}}$ are degenerate, since if we define $\tilde{l} = n - l + 2$, then $\gamma_l = \gamma_{\tilde{l}}$. For example, if $n = 10$ then the (ordered) eigenvalues of $\mathbf{Q}^{\mathrm{r}}$ are approximately

$$\gamma = [\ 0 \quad 1.6 \quad 1.6 \quad 0.6 \quad 0.6 \quad -0.6 \quad -0.6 \quad -1.6 \quad -1.6 \quad -2.0\ ]^T$$

and it is clear that there are four pairs of degenerate eigenvalues. We should really be examining the component of $\mathbf{v}$ in the eigenplanes of $\mathbf{T}^{\mathrm{opr}}$ instead of along each degenerate eigenvector of $\mathbf{T}^{\mathrm{opr}}$. Since we shall make extensive use of the 10 city travelling salesman problem as an illustrative example throughout this paper, it would be beneficial at this stage to introduce some notation to cope with the eigenvalue degeneracy of this particular problem. Let us define a 10×6 matrix $\mathbf{Z}$ of complex elements such that

$$\mathrm{Re}([\mathbf{Z}]_{kl}) = \begin{cases} [\mathbf{A}]_{kl} & \text{for } l = 1 \\ [\mathbf{A}]_{k,2l-2} & \text{for } 2 \leq l \leq 6 \end{cases} \tag{53}$$

$$\mathrm{Im}([\mathbf{Z}]_{kl}) = \begin{cases} 0 & \text{for } l = 1 \text{ or } l = 6 \\ [\mathbf{A}]_{k,2l-1} & \text{for } 2 \leq l \leq 5 \end{cases} \tag{54}$$

Hence, for the degenerate eigenvalues, the magnitude of $[\mathbf{Z}]_{kl}$ gives the magnitude of $\mathbf{v}$ in the associated eigenplane, while the phase of $[\mathbf{Z}]_{kl}$ gives the direction in that eigenplane. For the simple eigenvalues, the corresponding $[\mathbf{Z}]_{kl}$ is real and gives the component of $\mathbf{v}$ along the corresponding eigenvector. Let us also introduce a reduced 10×6 matrix ζ of eigenvalues of $\mathbf{T}^{\mathrm{opr}}$, such that

$$\zeta_{kl} = \begin{cases} \chi_{kl} & \text{for } l = 1 \\ \chi_{k,2l-2} & \text{for } 2 \leq l \leq 6 \end{cases} \tag{55}$$

The eigenvalues ζ_{kl} correspond to the eigenvectors and eigenplanes in the decomposition $\mathbf{Z}$.

4.3 Validity of solutions and bounds on A

In Section 4.1, it was demonstrated how confinement to the valid subspace fixed the elements in the first row and column of $\mathbf{A}$. We shall now go on to show how bounds on the other elements of $\mathbf{A}$ can be derived by considering the form of a valid solution. It is difficult to analyze the effect on $\mathbf{A}$ of the precise validity condition $[\mathbf{V}]_{ij} \in \{0,1\}$. However, some progress can be made by considering a slightly relaxed condition. For the two-graph matching problem (of which the travelling salesman and Hamilton path problems are special cases) the final $\mathbf{V}$ is a permutation matrix, and so

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}^n \tag{56}$$

Equation (56) is a necessary but not sufficient condition for $\mathbf{V}$ to be a valid solution. The combination of equation (56) and

$$[\mathbf{V}]_{ij} \geq 0 \tag{57}$$

is, however, sufficient [1]. Defining a matrix $\mathbf{A}^{\mathrm{s}}$ such that $[\mathbf{A}^{\mathrm{s}}]_{kl} = [\mathbf{A}]_{kl}^2$, and assuming that (56) (but not necessarily (57)) holds, it can be shown that [1]

$$\sum_{k=1}^{n} [\mathbf{A}^{\mathrm{s}}]_{kl} = \sum_{l=1}^{n} [\mathbf{A}^{\mathrm{s}}]_{kl} = 1 \tag{58}$$

Moreover, these conditions, which are valid for the fully converged network output, can be relaxed for the unconverged state vector to [1]

$$\sum_{k=1}^{n} [\mathbf{A}^{\mathrm{s}}]_{kl} \leq 1 \tag{59}$$

$$\sum_{l=1}^{n} [\mathbf{A}^{\mathrm{s}}]_{kl} \leq 1 \tag{60}$$

Hence the sum of the squares of the elements in any row or column of $\mathbf{A}$ cannot exceed one at any time, and must reach this value when the network has converged to a valid solution. The same conditions can be expressed in terms of the matrix $\mathbf{Z}$ for the 10 city travelling salesman problem. Defining the matrix $\mathbf{Z}^{\mathrm{s}}$ such that $[\mathbf{Z}^{\mathrm{s}}]_{kl} = |[\mathbf{Z}]_{kl}|^2$, the conditions are

$$\sum_{l=1}^{6} [\mathbf{Z}^{\mathrm{s}}]_{kl} \leq 1 \tag{61}$$

$$\sum_{k=1}^{10} [\mathbf{Z}^{\mathrm{s}}]_{kl} \leq \begin{cases} 1 & \text{for } l = 1 \text{ or } l = 6 \\ 2 & \text{for } 2 \leq l \leq 5 \end{cases} \tag{62}$$

with the strict equalities holding when $\mathbf{v}$ has converged to a valid solution. A similar result can be derived for the graph partitioning problem, for which

$$\mathbf{V}^T\mathbf{V} = \tfrac{n}{m}\mathbf{I}^m \tag{63}$$

In this case, following the approach in [1], it is straightforward to show that

$$\sum_{k=1}^{n} [\mathbf{A}^{\mathrm{s}}]_{kl} \leq \tfrac{n}{m} \tag{64}$$

with the strict equalities holding when $\mathbf{v}$ has converged to a valid solution. Hence, for the graph partitioning problem, we obtain limits on the sum of the squares of the elements in the columns of $\mathbf{A}$, but not in the rows.

5 Examining the behaviour of the network

5.1 Linearization of the network dynamics

The Hopfield network's dynamics are described by the following equations:

$$\dot{\mathbf{u}} = -\eta\mathbf{u} + \mathbf{T}\mathbf{v} + \mathbf{i}^{\mathrm{b}} \tag{65}$$

$$[\mathbf{v}]_i = g([\mathbf{u}]_i) \tag{66}$$

Let us assume that the network is initialized with $\mathbf{v}$ near the centre of the valid subspace, ie. with $\mathbf{v} \approx \mathbf{s}$. If we also assume that $\mathbf{v}$ is continually confined to the valid subspace, then it is only necessary to consider the dynamics of $\mathbf{v}^{\mathrm{val}}$ (49), the component of $\mathbf{v}$ in the valid subspace. For small perturbations of $\mathbf{u}$ and $\mathbf{v}$ around the initial condition, it is possible to linearize the network dynamics by writing $\mathbf{u}^{\mathrm{val}} = \alpha T^p \mathbf{v}^{\mathrm{val}}$, leading to the following dynamic equation for $\mathbf{v}^{\mathrm{val}}$ [1]:

$$\dot{\mathbf{v}}^{\mathrm{val}} = \left(\frac{1}{\alpha T^p}\mathbf{T}^{\mathrm{opr}} - \eta\mathbf{I}\right)\mathbf{v}^{\mathrm{val}} + \frac{1}{\alpha T^p}\mathbf{i}^{\mathrm{opr}} \tag{67}$$

Using equations (48) and (49), we can rewrite these dynamics in terms of $\mathbf{A}$ and $\mathbf{B}$ [1]:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{A}]_{kl} &= \tilde{\chi}_{kl}[\mathbf{A}]_{kl} + \frac{1}{\alpha T^p}[\mathbf{B}]_{kl} & (68) \\ \text{where} \quad \tilde{\chi}_{kl} &= \frac{\chi_{kl}}{\alpha T^p} - \eta & (69) \end{aligned}$$

This equation is valid for $2 \leq k \leq n$ and $2 \leq l \leq m$, since we know that the elements in the first row and column of $\mathbf{A}$ are fixed by confinement to the valid subspace. If we assume an initial condition $\mathbf{A} = \mathbf{A}^0$, then (68) can be integrated to give [1]

$$[\mathbf{A}]_{kl}(t) = \begin{cases} \left[\mathbf{A}^0\right]_{kl} + \frac{1}{\alpha T^p}[\mathbf{B}]_{kl}t & \text{for} \quad \tilde{\chi}_{kl} = 0 \\ \left(\left[\mathbf{A}^0\right]_{kl} + \frac{[\mathbf{B}]_{kl}}{\alpha T^p \tilde{\chi}_{kl}}\right)\exp(\tilde{\chi}_{kl}t) - \frac{[\mathbf{B}]_{kl}}{\alpha T^p \tilde{\chi}_{kl}} & \text{for} \quad \tilde{\chi}_{kl} \neq 0 \end{cases} \tag{70}$$

If we are considering a network with piecewise linear transfer functions, as in Figure 2, then equations (70) are an exact description of the network dynamics until $\mathbf{v}$ reaches a hypercube face.

5.2 The effect of the linear bias term $\mathbf{i}^{\mathrm{opr}}$

If we assume that $\mathbf{i}^{\mathrm{opr}} = \mathbf{0}$, then the linearized dynamics become

$$[\mathbf{A}]_{kl}(t) = \left[\mathbf{A}^0\right]_{kl}\exp(\tilde{\chi}_{kl}t) \tag{71}$$

Equation (71) indicates that the sign of $[\mathbf{A}]_{kl}$ is totally dependent on its initial value $\left[\mathbf{A}^0\right]_{kl}$, which is generally random for $k \geq 2$ and $l \geq 2$. In other words, in the absence of a linear bias term $\mathbf{i}^{\mathrm{opr}}$, the state vector will develop along the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$ in an unpredictable direction, and we would expect the final solution to be highly dependent on the initial state $\left[\mathbf{A}^0\right]_{kl}$. This means that for most problems, unless there is a multiplicity of optimal solutions located evenly around the valid subspace (as in the case of the travelling salesman problem [1]), the presence of the linear bias term is crucial if the network is to converge to a unique solution, let alone the optimal one.

However, the effect of the linear bias term on the network dynamics is not always beneficial. To start with, it is often the case that $\mathbf{i}^{\mathrm{opr}}$ is virtually orthogonal to many of the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$, and so the evolution of $\mathbf{v}$ along these eigenvectors remains indeterminate in terms of direction [10]. As a result, the network cannot be relied upon to converge to a unique solution, even though $\mathbf{i}^{\mathrm{op}} \neq \mathbf{0}$, and the performance of the network is variable. Even if $\mathbf{i}^{\mathrm{opr}}$ is capable of guiding the state vector towards a unique solution, its effect can still sometimes be detrimental. To see why this is so, first consider substituting a truncated Taylor series for the exponential in (70), and assume that the $\left[\mathbf{A}^0\right]_{kl}$ are small for $k \geq 2$ and $l \geq 2$. It subsequently transpires that for small t [10]

$$\begin{aligned} [\mathbf{A}]_{kl}(t) &\approx \left[\mathbf{A}^0\right]_{kl} + \frac{1}{\alpha T^p}[\mathbf{B}]_{kl}t \\ \Leftrightarrow \mathbf{v}(t) &\approx \mathbf{v}(0) + \frac{1}{\alpha T^p}\mathbf{i}^{\mathrm{opr}}t & (72) \end{aligned}$$

Hence $\mathbf{i}^{\mathrm{opr}}$ is the driving force behind the initial evolution of $\mathbf{v}$. It is useful to note that the linear bias term can be associated with a certain auxiliary linear problem [10], namely the minimization of

$$E^{\mathrm{alp}} = -\mathbf{v}^T\mathbf{i}^{\mathrm{opr}} \tag{73}$$

In some cases the auxiliary linear problem can be poorly related to the parent problem, inasmuch as the optimal solution to the auxiliary linear problem lies in a completely different region of state space to that of the parent problem. Given that the network dynamics are initially concerned with minimizing E^{alp}, it is not surprising that the initial behaviour of the network can often be responsible for its eventual failure to find a good solution to the parent problem. In such cases, it may be advantageous to reformulate the problem such that $\mathbf{i}^{\mathrm{opr}} = \mathbf{0}$, in so doing removing the initial dependence on the auxiliary linear problem; we shall see how this can be achieved in Section 7.1. Of course, this means that the network stands no chance of converging to a unique solution, and so for problems with no solution degeneracy, it will be necessary to run the network several times, with different random starting vectors, to ensure that the best possible solution has been observed. Fortunately, an effect of annealing, which will be illustrated in Section 6.1, is to limit the number of solutions a network running with $\mathbf{i}^{\text{opr}} = \mathbf{0}$ can find. Experimental evidence suggests that, for cases where $\mathbf{T}^{\text{opr}}$ exhibits no eigenvalue degeneracy, the network can generally converge to one of two solutions. Moreover, if an initial condition $\mathbf{v}^{\text{val}} = \mathbf{v}_{\text{o}}^{\text{val}}$ leads to one such solution, then the initial condition $\mathbf{v}^{\text{val}} = -\mathbf{v}_{\text{o}}^{\text{val}}$ leads to the other. Hence the network need be run at most twice to be sure of observing the best network performance on any particular problem, and only once if there exists a suitable multiplicity of optimal solutions, as with the travelling salesman problem.

Dispensing with the $\mathbf{i}^{\text{opr}}$ term also simplifies the mathematical analysis considerably, and since all problems may be cast into this form, we shall from this point on assume $\mathbf{i}^{\text{opr}} = \mathbf{0}$, and refer to the simplified dynamics (71).

5.3 Evolution of v along each eigenvector

Examining equation (71), it is clear that if the network state vector $\mathbf{v}$ is going to evolve along a particular eigenvector $\mathbf{x}^{kl}$, it is necessary that $\tilde{\chi}_{kl} \geq 0$. If, on the other hand, $\tilde{\chi}_{kl} < 0$, then $[\mathbf{A}]_{kl} \to 0$ and the evolution of $\mathbf{v}$ along $\mathbf{x}^{kl}$ is blocked. Hence, those $[\mathbf{A}]_{kl}$ with $\tilde{\chi}_{kl} < 0$ are not responsible for the progress of $\mathbf{v}$ towards a hypercube corner. Unfortunately the situation is not quite so simple for the MFA dynamics, which perform an Euler approximation of the continuous Hopfield network dynamics with $\eta = 1$ and a time step $\Delta t = 1$. Since this is a rather large time step, it is feasible that the simulation may fail to stabilize those $[\mathbf{A}]_{kl}$ with $\tilde{\chi}_{kl} \ll 0$, over-correcting at each time step and leading to an underdamped oscillation of the $[\mathbf{A}]_{kl}$'s. Hence the presence of negative $\tilde{\chi}_{kl}$'s can result in instability of the MFA dynamics, especially if the MFA equations are updated synchronously [20].

While the above results were arrived at through a linearized analysis, valid only at the start of convergence when $\mathbf{v}^{\text{val}} \approx \mathbf{0}$, it is possible to use an energy based argument to extend their applicability for the case $\eta = 0$. The effective objective function, E^{opr}, may be expressed relative to the eigenvector basis as follows [1]:

$$E^{\text{opr}} \quad = \quad -\tfrac{1}{2} \sum_{k=2}^{n} \sum_{l=2}^{m} \chi_{kl} \, [\mathbf{A}]_{kl}^2 \qquad \text{for } \mathbf{i}^{\text{opr}} = \mathbf{0} \tag{74}$$

If we view the network dynamics as continuously reducing the objective function E^{opr}, then by inspection of equation (74), it is clear that for $\chi_{kl} < 0$ it is desirable that $[\mathbf{A}]_{kl} \to 0$. When the nonlinearities of the network are taken into account, the $[\mathbf{A}]_{kl}$'s are no longer independent, so it is not the case that the network dynamics will necessarily enforce $[\mathbf{A}]_{kl} \to 0$ for $\chi_{kl} < 0$, if in so doing the augmentation of an $[\mathbf{A}]_{kl}$ with a large positive χ_{kl} is prevented. However, it is reasonable to assume that those $[\mathbf{A}]_{kl}$ with $\chi_{kl} < 0$ will not evolve to any considerable degree, since this is unlikely to result in a reduction of E^{opr}.

To extend the applicability of the energy argument to cover MFA, we must consider the effect of setting $\eta = 1$. Since the η term in the network's Liapunov function (15) is convex and minimized at $\mathbf{v} = \mathbf{0}$, its effect is roughly to force the network to evolve $[\mathbf{A}]_{kl}$ to a value somewhat smaller that would be expected given the minimization of E^{opr} alone. Hence the result that $\chi_{kl} < 0$ prevents the significant evolution of $[\mathbf{A}]_{kl}$ throughout convergence is certainly true for MFA as well, with the same proviso concerning instability for $\tilde{\chi}_{kl} \ll 0$.

6 Annealing

Some form of annealing has been commonly employed by most researchers in the field to improve the quality of solutions attained by optimizing networks. In this section we shall examine two different annealing techniques for the Hopfield network and MFA, concluding that they both have much the same effect on the evolution of $\mathbf{A}$.

6.1 Hysteretic annealing for the Hopfield network

Hysteretic annealing [6] has proved particularly successful when applied to the Hopfield network system. Similar schemes have been independently examined by several researchers: see, for example, convex relaxation [19] and matrix graduated non-convexity [1]. The idea is to incorporate a variable amount of self-feedback into the network dynamics, so that

$$\dot{\mathbf{u}} = \mathbf{T}\mathbf{v} + \beta\mathbf{v} + \mathbf{i}^{\rm b} \tag{75}$$

The consequences of the self-feedback can be most conveniently analyzed by replacing $\mathbf{T}^{\rm op}$ with $\mathbf{T}^{\rm op} + \beta\mathbf{I}$ in the foregoing analysis. The problem objective function $E^{\rm op}$ becomes

$$E^{\rm op} = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\rm op}\mathbf{v} - (\mathbf{i}^{\rm op})^T\mathbf{v} - \beta||\mathbf{v}||^2$$

which is still a valid objective for the problem, provided $||\mathbf{v}||^2$ is the same for all valid solutions (as is the case for all the problems considered in this paper). The eigenvectors of $\mathbf{T}^{\rm opr}$ are unchanged, but the eigenvalues become [1, 10]

$$\chi_{kl} = \begin{cases} \lambda_k\gamma_l + \beta & \text{for } k \geq 2 \text{ and } l \geq 2 \\ 0 & \text{for } k = 1 \text{ or } l = 1 \end{cases} \tag{76}$$

Hence

$$\tilde{\chi}_{kl} = \begin{cases} \dfrac{1}{\alpha T^p}\left(\lambda_k\gamma_l + \beta\right) & \text{for } k \geq 2 \text{ and } l \geq 2 \\ 0 & \text{for } k = 1 \text{ or } l = 1 \end{cases} \tag{77}$$

The annealing starts with a sufficiently negative value of β such that all the $\tilde{\chi}_{kl}$ are negative; the network therefore stabilizes at the centre of the valid subspace, since $[\mathbf{A}]_{kl} \rightarrow 0$ for all $k \geq 2, l \geq 2$. The parameter β is then gradually increased, so that one of the $\tilde{\chi}_{kl}$ becomes positive, at which point the corresponding $[\mathbf{A}]_{kl}$ is free to increase in magnitude. Only when β is increased further is it possible for other $[\mathbf{A}]_{kl}$'s to depart significantly from their stable values of zero. Thus the action of annealing is to control the evolution of $\mathbf{A}$ so that the $[\mathbf{A}]_{kl}$'s are introduced roughly in the order of the corresponding $\tilde{\chi}_{kl}$'s. In other words, $\mathbf{v}$ evolves along those $\mathbf{x}^{kl}$ with the largest positive eigenvalues first, and then along those with the more negative eigenvalues later. By making β sufficiently positive, all the $\tilde{\chi}_{kl}$'s can become positive, at which point the $||\mathbf{v}||^2$ term dominates the Liapunov function, ensuring convergence to a hypercube corner as the network drives $\mathbf{v}$ outwards in order to maximize $||\mathbf{v}||^2$.

Figure 3 illustrates the effects of annealing and using different random starting vectors on a simple, hypothetical problem. In Figure 3(a) we see a 2-dimensional valid subspace within a 3-dimensional unit cube; there are three valid solution points at the cube corners labelled A, B and C, all equidistant from the origin O. In Figure 3(b) the contours of the effective objective function $E^{\rm opr}$ have been marked on the valid subspace. $E^{\rm opr}$ is a pure quadratic form with $\mathbf{i}^{\rm opr} = \mathbf{0}$, so there is a stationary point at the centre of the valid subspace, equidistant from the three valid solution points. The eigenvectors of $\mathbf{T}^{\rm opr}$, $\mathbf{x}^1$ and $\mathbf{x}^2$, are the axes of the conic sections which make up the contours of $E^{\rm opr}$. We have assumed that both of the eigenvalues of $\mathbf{T}^{\rm opr}$, χ_1 and χ_2, are positive, so that the contours are ellipses. We have also assumed that $\chi_1 > \chi_2$, so the descent is steepest in the $\mathbf{x}^1$ direction. Examining the contours, it is clear that the corner C represents the unique global optimum to this problem. As discussed in Section 5.2, since $\mathbf{i}^{\rm opr} = \mathbf{0}$ and there is no solution degeneracy, we would expect to have to anneal the network twice to be sure of observing the best possible solution, starting once with $\mathbf{v}^{\rm val} = \mathbf{v}^{\rm val}_{\rm o}$, and then again with $\mathbf{v}^{\rm val} = -\mathbf{v}^{\rm val}_{\rm o}$. Looking at Figure 3(b), we see that, in the absence of an annealing process, any of the three corners A, B and C can reached by descent dynamics starting from positions near the centre of the valid subspace.

The effect of annealing is to make the network behaviour less dependent on the random starting position and increase the chances of the network finding the optimal solution. Figure 3(c) shows the contours at the start of an annealing process, when only χ_1 is positive. The stationary point at the centre of the valid subspace has been turned into a saddle point, and any move away from the starting position which increases the magnitude of the $\mathbf{x}^2$ component also increases E^{opr}. Let us consider a network with piecewise linear transfer functions, as in Figure 2, so that the linearized analysis is valid until the state vector reaches a hypercube face. The subsequent trajectory of the state vector will therefore reduce the $\mathbf{x}^2$ component towards zero and increase the magnitude of the $\mathbf{x}^1$ component: such a trajectory, starting from a position $\mathbf{v}^{\text{val}} = \mathbf{v}_{\text{o}}^{\text{val}}$, is shown in Figure 3(c). When the state vector reaches a hypercube face, its components along $\mathbf{x}^1$ and $\mathbf{x}^2$ are no longer independent, and so it is possible that the component along $\mathbf{x}^2$ can now increase, even though $\chi_2 < 0$. In Figure 3(d) we see that this is indeed what happens in this case, and the optimal solution is attained. Figure 3(e) shows the effect of starting from an initial position $\mathbf{v}^{\text{val}} = -\mathbf{v}_{\text{o}}^{\text{val}}$. This time the $\mathbf{x}^1$ component is introduced with the opposite sign, leading to a suboptimal solution in Figure 3(f). Note that, for this second starting position, it was necessary to advance the annealing so that $\chi_2 > 0$, in order to force convergence to a hypercube corner.

This example illustrates the necessity of running the network twice, starting once with $\mathbf{v}^{\text{val}} = \mathbf{v}_{\text{o}}^{\text{val}}$, and then again with $\mathbf{v}^{\text{val}} = -\mathbf{v}_{\text{o}}^{\text{val}}$, unless there is a multiplicity of equivalent solutions located at regular intervals around the valid subspace, as with the TSP. Furthermore, we see that the annealing has made it necessary to run the network *only* twice, whereas with no annealing it would be necessary to try many random starting vectors to be sure of finding the optimal solution at least once. Remember that it may be necessary to run the network twice even if $\mathbf{i}^{\text{opr}} \neq \mathbf{0}$, since it is possible that $\mathbf{i}^{\text{opr}}$ may be orthogonal to $\mathbf{x}^1$, in which case $\mathbf{v}^{\text{val}}$ will still evolve in one of two directions along $\mathbf{x}^1$, depending on the sign of the random starting vector.

6.2 Temperature annealing for MFA

A more familiar form of annealing is often employed in conjunction with the MFA equations. In a direct parallel with simulated annealing, the 'temperature' parameter T^p is gradually reduced from its initial value as the network converges to a hypercube corner. Recalling that for MFA

$$\tilde{\chi}_{kl} = \frac{\chi_{kl}}{\alpha T^p} - 1$$

it is apparent that $[\mathbf{A}]_{kl}$ can depart from its stable value of zero only when $\chi_{kl} \geq \alpha T^p$. Hence, the effect of reducing T^p is to gradually free more of the $[\mathbf{A}]_{kl}$, again in the order of the corresponding χ_{kl}'s. Unlike the case of hysteretic annealing, those $\tilde{\chi}_{kl}$ with associated $\chi_{kl} < 0$ will never become positive in the course of annealing, and so evolution of $\mathbf{v}$ along these eigenvectors cannot take place, except in the unstable sense as $T^p \to 0$ and the $\tilde{\chi}_{kl}$ become very negative. Instability of the MFA equations can be avoided by ensuring that none of the $\tilde{\chi}_{kl}$ become excessively negative. This is usually achieved by mixing hysteretic annealing with temperature annealing, using a fixed value of β to make the χ_{kl}'s more positive. The stabilizing effect of β has been previously noted in [20].

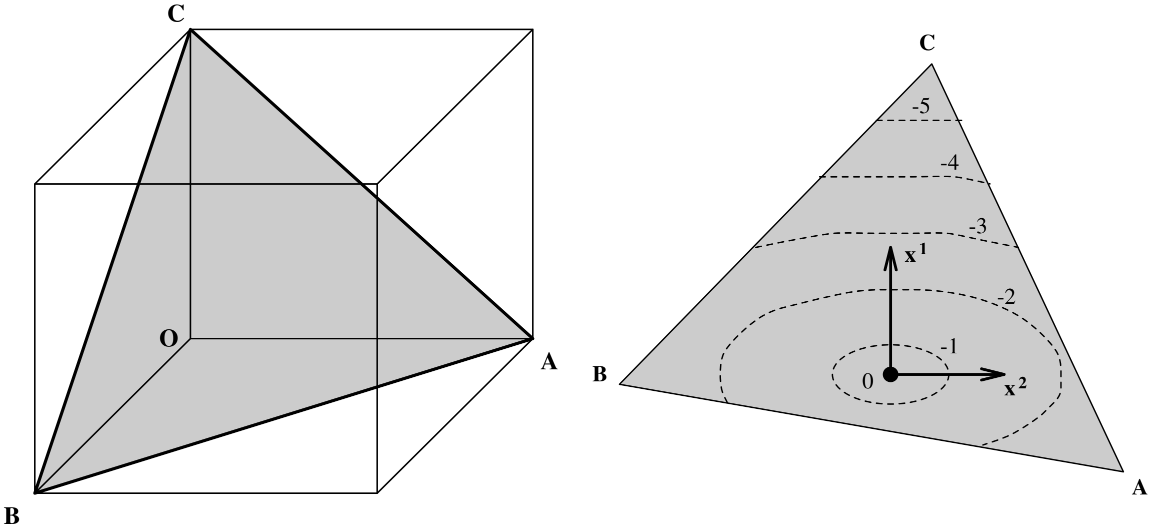
6.3 Annealing for the two-graph matching problem

The likely result of such annealing processes can be examined in more detail for the two-graph matching problem, for which $n = m$ and so $\mathbf{A}$ is square. Remember that the travelling salesman and Hamilton path problems are both special cases of the two-graph matching problem. In what follows, hysteretic annealing will be used as our example, though it should be realized that temperature annealing has approximately the same effect.

To begin with, let us assume that the λ_k and γ_l are all positive for $k \geq 2$ and $l \geq 2$. Recalling that the eigenvalues of $\mathbf{T}^{\text{opr}}$ are given by

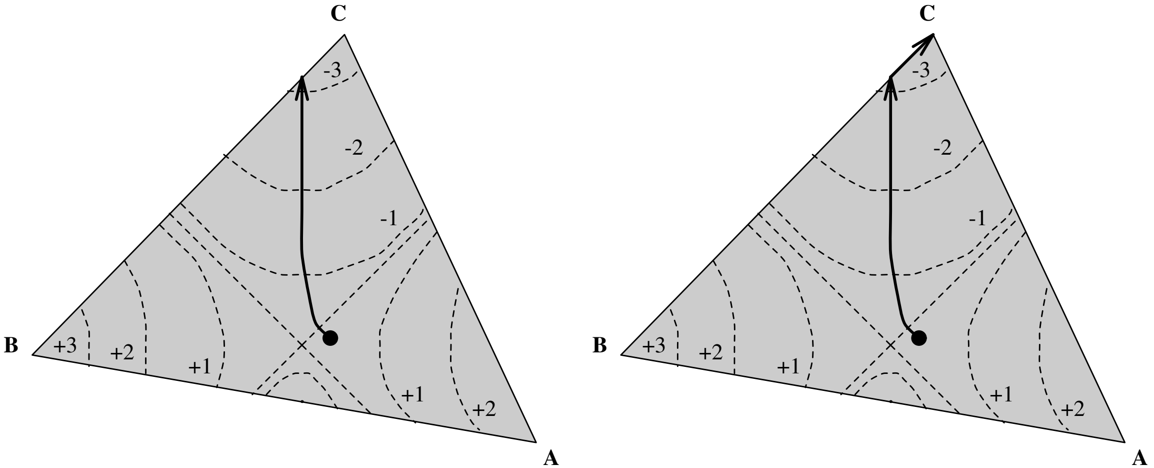
$$\chi_{kl} = \lambda_k \gamma_l + \beta \tag{78}$$

and that the λ_k's and γ_l's are ordered as in (42) and (43), it is apparent that the most positive eigenvalue of $\mathbf{T}^{\text{opr}}$ is χ_{22}. The network will therefore introduce $[\mathbf{A}]_{22}$ first, until $[\mathbf{A}^{\text{s}}]_{22} = 1$, at which point the magnitude of $[\mathbf{A}]_{22}$ can increase no further without violating conditions (59)



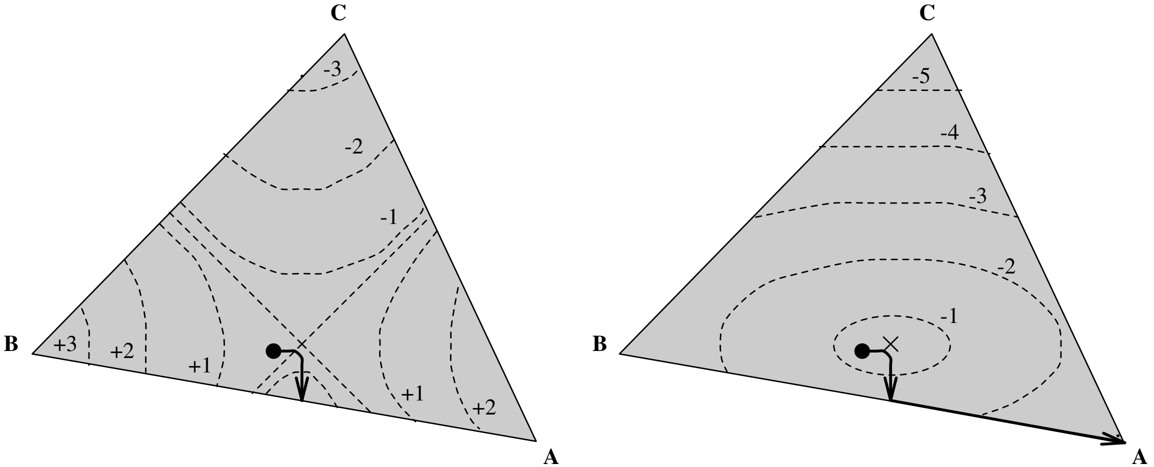
(a) Valid subspace
within unit hypercube

(b) Contours of E^{opr} and
eigenvectors of $\mathbf{T}^{\mathrm{opr}}$



(c) Start of annealing,
initial $\mathbf{v}^{\mathrm{val}} = \mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$

(d) End of annealing,
initial $\mathbf{v}^{\mathrm{val}} = \mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$



(e) Start of annealing,
initial $\mathbf{v}^{\mathrm{val}} = -\mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$

(f) End of annealing,
initial $\mathbf{v}^{\mathrm{val}} = -\mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$

Figure 3: The effects of annealing and different random starting positions.

and (60). The dynamics therefore stabilize at this point until β is increased further so that another $\tilde{\chi}_{kl}$ becomes positive. Note that the presence of $[\mathbf{A}^{\mathrm{s}}]_{22} = 1$ means that no further elements in the second row or column of $\mathbf{A}$ can be introduced without violating conditions (59) and (60). The next most positive $\tilde{\chi}_{kl}$ with $k \neq 2$ and $l \neq 2$ is χ_{33}. The dynamics will therefore introduce $[\mathbf{A}]_{33}$ until its magnitude equals one, and then stabilize again. The presence of $[\mathbf{A}]_{33}$ blocks the other elements in the third row and column of $\mathbf{A}$, so the next element to be introduced will be $[\mathbf{A}]_{44}$. This process continues, and it is apparent that the network ideally reaches a state for which $\mathbf{A}^{\mathrm{s}} = \mathbf{I}^{n}$. Now, in the above analysis, the condition $[\mathbf{V}]_{ij} \geq 0$ (57) was ignored, and it is highly unlikely that the state $\mathbf{A}^{\mathrm{s}} = \mathbf{I}^{n}$ can be achieved without violating this condition. However, it is reasonable to propose that the network will attain a state not unlike $\mathbf{A}^{\mathrm{s}} = \mathbf{I}^{n}$, without violating (57).

The above reasoning can be extended to cover cases in which $\mathbf{P}^{\mathrm{r}}$ and $\mathbf{Q}^{\mathrm{r}}$ have both positive and negative eigenvalues [1], with the same result. The corresponding result for the 10 city travelling salesman problem is that the network will find a solution for which $\mathbf{Z}^{\mathrm{s}}$ is not unlike the matrix $\mathbf{Z}^{10}$, where

$$\mathbf{Z}^{10} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

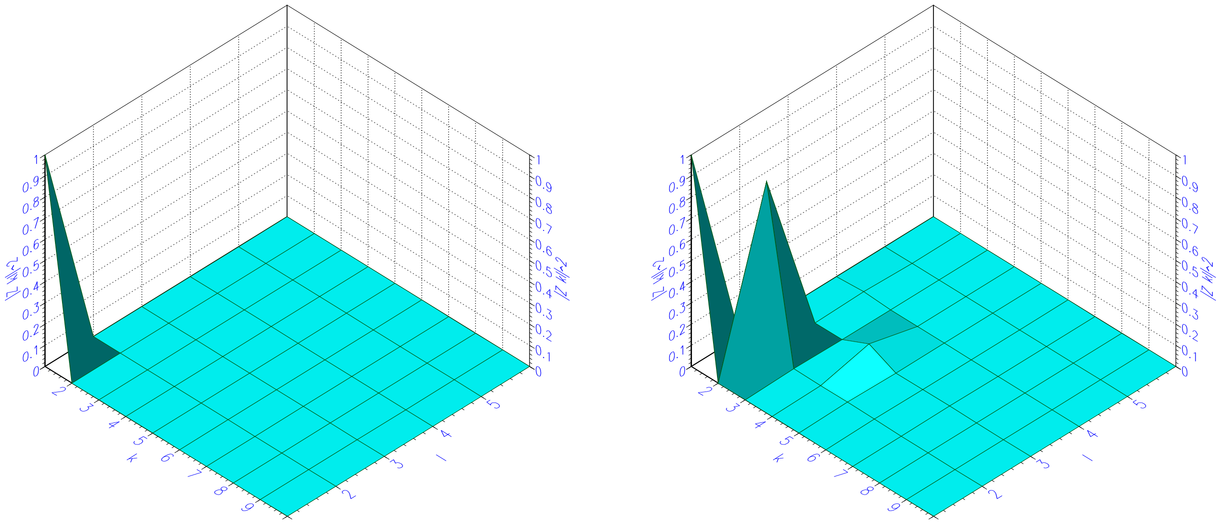
To illustrate the annealing process, let us consider a particular 10 city Euclidean travelling salesman problem, where the cities have been randomly placed within a unit square. The $\mathbf{P}$ and $\mathbf{Q}$ matrices are set according to the mappings in Table 1, and it transpires that the reduced eigenvalue matrix ζ for this problem is

$$\zeta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.58 & 1.37 & -1.37 & -3.58 & -4.42 \\ 0 & 1.69 & 0.64 & -0.64 & -1.69 & -2.08 \\ 0 & 0.87 & 0.33 & -0.33 & -0.87 & -1.08 \\ 0 & 0.62 & 0.24 & -0.24 & -0.62 & -0.76 \\ 0 & 0.39 & 0.15 & -0.15 & -0.39 & -0.48 \\ 0 & 0.24 & 0.09 & -0.09 & -0.24 & -0.29 \\ 0 & 0.19 & 0.07 & -0.07 & -0.19 & -0.24 \\ 0 & 0.18 & 0.07 & -0.07 & -0.18 & -0.22 \\ 0 & 0.16 & 0.06 & -0.06 & -0.16 & -0.20 \end{bmatrix} \qquad \mathrm{order}(\zeta) = \begin{bmatrix} - & - & - & - & - & - \\ - & 1 & 3 & 41 & 44 & 45 \\ - & 2 & 5 & 37 & 42 & 43 \\ - & 4 & 8 & 33 & 39 & 40 \\ - & 6 & 10 & 29 & 36 & 38 \\ - & 7 & 14 & 23 & 34 & 35 \\ - & 9 & 15 & 22 & 30 & 32 \\ - & 11 & 16 & 21 & 26 & 31 \\ - & 12 & 17 & 20 & 25 & 28 \\ - & 13 & 18 & 19 & 24 & 27 \end{bmatrix}$$

where the right hand matrix shows the ordering of the elements of ζ; the dashes correspond to eigenvalues outside the valid subspace, which play no part in the network's dynamics.

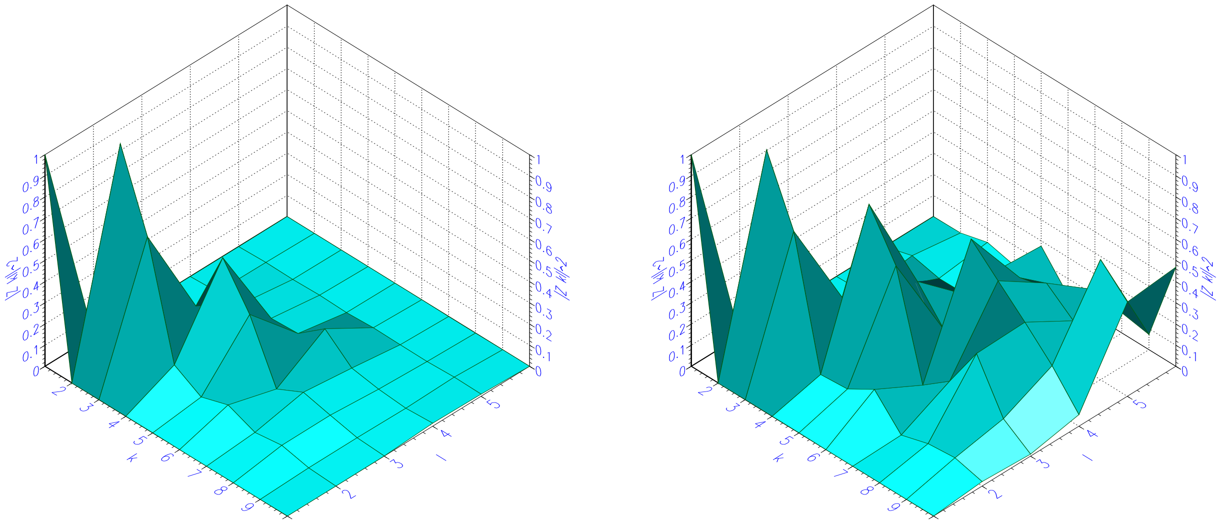
Figure 4 shows the matrix $\mathbf{Z}^{\mathrm{s}}$ at various values of β as a Hopfield network converges towards a valid solution. For $\beta = -4.5$, all the $\tilde{\zeta}_{kl}$'s are negative, and so the network stabilizes at a point near the centre of the valid subspace. For $\beta = -4.0$ only $\tilde{\zeta}_{22}$ is positive, and the network has increased the magnitude of $[\mathbf{Z}]_{22}$ as predicted. By the time $\beta = 0.2$, the $\mathbf{Z}^{\mathrm{s}}$ matrix has evolved to a form not unlike the expected $\mathbf{Z}^{10}$ defined above, though those $[\mathbf{Z}]_{kl}$ with negative $\tilde{\zeta}_{kl}$'s have yet to be introduced. Finally, when $\beta = 0.4$ the network has successfully converged to a valid hypercube corner, and the final form of $\mathbf{Z}^{\mathrm{s}}$ is similar to $\mathbf{Z}^{10}$. So this illustrative experiment is in good agreement with the theory presented above. The tour length for this solution is 2.96 units, which is in fact the global optimum for this problem.

In order to demonstrate the equivalence between hysteretic annealing and temperature annealing, Figure 5 shows how the $\mathbf{Z}^{\mathrm{s}}$ matrix evolves for the same problem being solved by MFA with neuron normalization [1, 20, 23]. It is apparent that the state vector follows a similar trajectory to that of the Hopfield network, leading to an identical final solution with a tour length of 2.96 units.



(a) $\beta = -4.0$

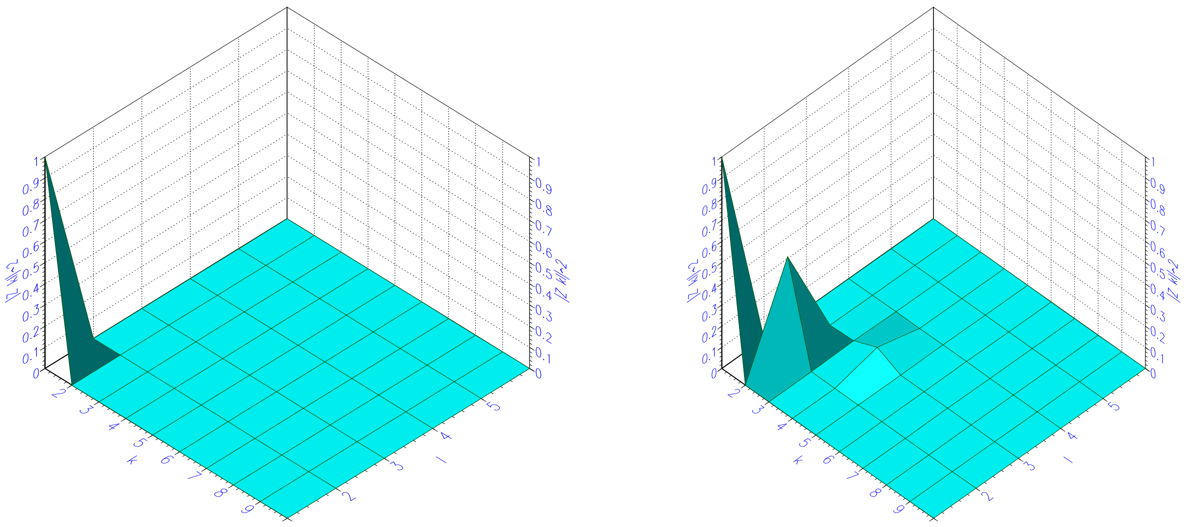
(b) $\beta = -1.9$



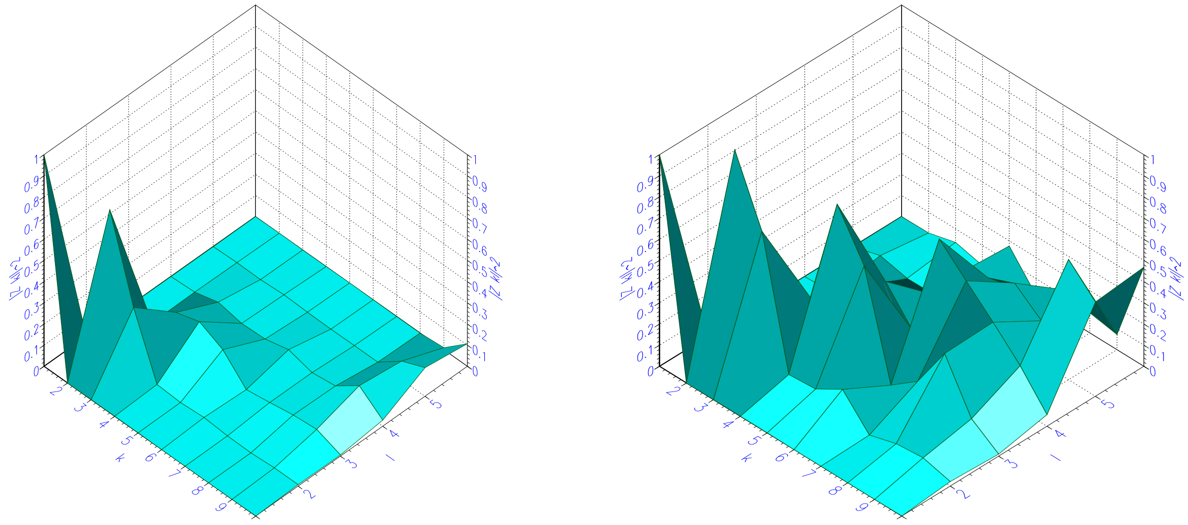
(c) $\beta = -0.1$

(d) $\beta = 1.5$ (Fully converged)

Figure 4: $\mathbf{Z}^s$ matrix for the 10 city TSP being solved by a Hopfield network with hysteretic annealing. The solution, seen decomposed in (d), is optimal.



(a) $T^p = 0.5$ (b) $T^p = 0.35$



(c) $T^p = 0.30$ (d) $T^p = 0.20$ (Fully converged)

Figure 5: $\mathbf{Z}^s$ matrix for the 10 city TSP being solved by MFA with temperature annealing. The solution, seen decomposed in (d), is optimal.

7 Alternative Energy Functions

There are an infinite number of continuous functions which are suitable objectives for any particular combinatorial optimization problem. Moreover, it is often the case that an infinite number of suitable quadratic functions exist, any of which can be used in conjunction with the Hopfield or MFA dynamics. We have already come across a class of such functions when we considered hysteretic annealing, for which it was demonstrated that any function of the form

$$E^{\mathrm{op}} = -\tfrac{1}{2}\mathbf{v}^T \mathbf{T}^{\mathrm{op}}\mathbf{v} - (\mathbf{i}^{\mathrm{op}})^T \mathbf{v} - \beta \|\mathbf{v}\|^2$$

was a suitable objective for a particular problem, so long as all valid solution points have the same $\|\mathbf{v}\|^2$. It is readily apparent that when we are attempting to solve a discrete problem by descent on a continuous energy function, the positioning of the discrete problem's solutions relative to important features of the continuous energy function is going to be critical to the success of the descent procedure. Given that there are many applicable continuous energy functions, we would expect some to be more suitably related to the underlying discrete problem than others, and give correspondingly better solutions with the Hopfield or MFA dynamics.

There are several levels at which it is possible to derive alternative energy functions for the same underlying discrete problem. At the highest level, we might choose to modify the problem representation, so that, taking the two-graph matching problem as an example, we are no longer searching for a permutation matrix $\mathbf{V}$, but for some other form by which a solution may be uniquely represented. At this level of abstraction the possibilities for alternative energy functions are vast. We shall limit ourselves in this discussion to retaining the usual solution representations, and investigating alternative functions compatible with these representations. As has previously been mentioned, there are an infinite number of such functions, though identifying analytical expressions for them is far from straightforward.

One such class of functions may be derived for the unlabelled two-graph matching problem. By 'unlabelled', we mean that there are no features identified with any single node in isolation, so that the edge weights obey $e_{xx} = 0$ for both graphs (the travelling salesman and Hamilton path problems both exhibit this property). The usual objective function for this problem (see Table 1) is obtained by setting $\mathbf{P}$ to be the edge weight matrix for the first graph, and $\mathbf{Q}$ to be the edge weight matrix for the second graph, and minimizing

$$E^{\mathrm{op}}_{\mathrm{o}} = -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{V}^T \mathbf{P} \mathbf{V} \mathbf{Q}\right) \quad (\text{where } [\mathbf{P}]_{ii} = [\mathbf{Q}]_{ii} = 0)$$

subject to $\mathbf{V}$ being a valid permutation matrix. We shall now propose a class of alternative functions

$$E^{\mathrm{op}} = -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{V}^T (\mathbf{P} + \phi \mathbf{I}^n)\mathbf{V}(\mathbf{Q} + \mathbf{D}^{\mathrm{q}})\right) \tag{79}$$

where $\mathbf{D}^{\mathrm{q}}$ is an $n \times n$ diagonal matrix and ϕ is a scalar constant. Using the fact that $\mathbf{V}$ is a permutation matrix for valid solutions, we obtain

$$\begin{aligned} E^{\mathrm{op}} &= E^{\mathrm{op}}_{\mathrm{o}} - \tfrac{1}{2}\left[\mathrm{trace}\left(\mathbf{V}^T \mathbf{P} \mathbf{V} \mathbf{D}^{\mathrm{q}}\right) + \phi\ \mathrm{trace}\left(\mathbf{V}^T \mathbf{I}^n \mathbf{V} \mathbf{Q}\right) + \phi\ \mathrm{trace}\left(\mathbf{V}^T \mathbf{I}^n \mathbf{V} \mathbf{D}^{\mathrm{q}}\right)\right] \\ &= E^{\mathrm{op}}_{\mathrm{o}} - \tfrac{1}{2}\left[\sum_{i=1}^{n} \left[\mathbf{V}\mathbf{D}^{\mathrm{q}}\mathbf{V}^T\right]_{ii} [\mathbf{P}]_{ii} + \phi \sum_{i=1}^{n} [\mathbf{Q}]_{ii} + \phi \sum_{i=1}^{n} [\mathbf{D}^{\mathrm{q}}]_{ii}\right] \\ &= E^{\mathrm{op}}_{\mathrm{o}} - \tfrac{1}{2}\phi\ \mathrm{trace}(\mathbf{D}^{\mathrm{q}}) \end{aligned} \tag{80}$$

Hence E^{op} is a valid objective function for any $\mathbf{D}^{\mathrm{q}}$ or ϕ, since, to within a constant, the energy at the valid hypercube corners is the same as that obtained using the original function $E^{\mathrm{op}}_{\mathrm{o}}$. The symmetry of the formulation indicates that it is equally valid to add an arbitrary diagonal matrix $\mathbf{D}^{\mathrm{p}}$ to $\mathbf{P}$, and a multiple of the identity matrix to $\mathbf{Q}$. The alternative energy functions so obtained result in $\mathbf{T}^{\mathrm{opr}}$ having different eigenvectors and eigenvalues, and so we would expect the performance of descent procedures on these functions to be quite variable. We shall now go on to consider how to exploit this flexibility to our advantage.

7.1 Removing the linear bias term

As mentioned earlier, the linear bias term $\mathbf{i}^{\mathrm{opr}}$ can be a nuisance, as it dominates the network dynamics at the start of convergence, and can guide $\mathbf{v}$ into a poor region of state space on the basis of an inappropriate linear objective function E^{alp} (73). In this section, we shall consider how an alternative objective function may be selected for any unlabelled two-graph matching problem such that the linear bias term $\mathbf{i}^{\mathrm{opr}}$ vanishes. Recall that for $n = m$ (39)

$$\mathbf{i}^{\mathrm{opr}} = \tfrac{1}{n}(\mathbf{R}^n\mathbf{P}\mathbf{o}^n \otimes \mathbf{R}^n\mathbf{Q}\mathbf{o}^n)$$

Now consider adding a diagonal matrix $\mathbf{D}^{\mathrm{q}}$ to $\mathbf{Q}$, where

$$[\mathbf{D}^{\mathrm{q}}]_{ii} = \theta - [\mathbf{Q}\mathbf{o}^n]_i = \theta - \sum_{j=1}^{n} [\mathbf{Q}]_{ij} \tag{81}$$

and θ is an arbitrary constant. This results in all the row sums of $(\mathbf{Q} + \mathbf{D}^{\mathrm{q}})$ equalling θ, in which case $(\mathbf{Q} + \mathbf{D}^{\mathrm{q}})\mathbf{o}^n = \theta\mathbf{o}^n$. The linear bias term subsequently becomes

$$\begin{aligned} \mathbf{i}^{\mathrm{opr}} &= \tfrac{1}{n}(\mathbf{R}^n\mathbf{P}\mathbf{o}^n \otimes \mathbf{R}^n(\mathbf{Q} + \mathbf{D}^{\mathrm{q}})\mathbf{o}^n) \\ &= \tfrac{1}{n}(\mathbf{R}^n\mathbf{P}\mathbf{o}^n \otimes \theta\mathbf{R}^n\mathbf{o}^n) \\ \Leftrightarrow \mathbf{i}^{\mathrm{opr}} &= \mathbf{0} \end{aligned} \tag{82}$$

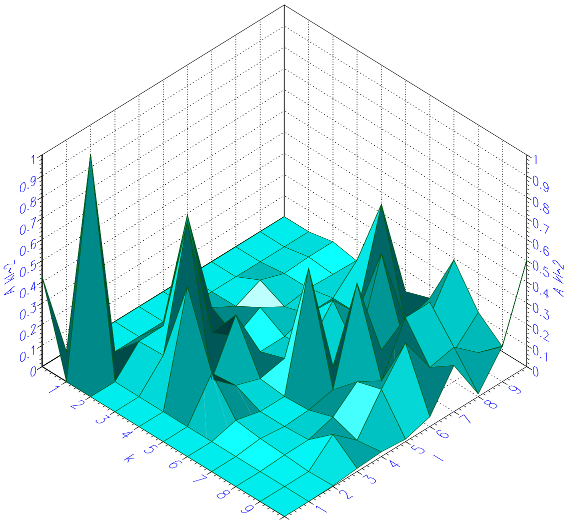
In this manner, any unlabelled two-graph matching problem can be formulated with no linear bias term. The elements of $\mathbf{D}^{\mathrm{q}}$ may be made, on average, as small as possible by setting

$$\theta = \tfrac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} [\mathbf{Q}]_{ij} \tag{83}$$

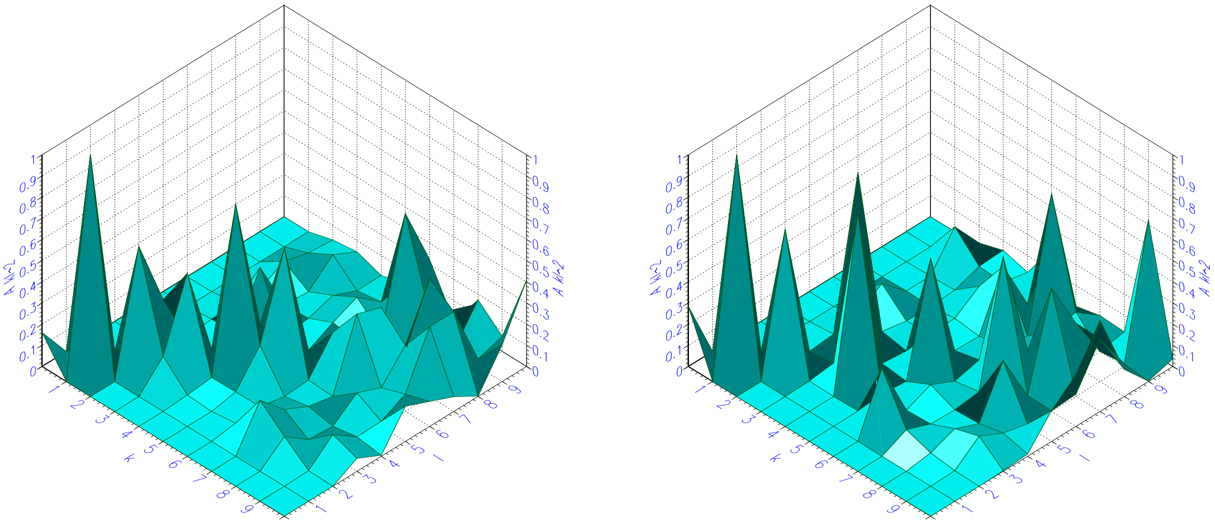
which ensures that the eigenvalues of the modified $\mathbf{Q}$ are not uniformly shifted by any significant amount; this may be advantageous when attempting to modify the annealing process, as described in Section 7.2.

While this technique is of no relevance to the travelling salesman problem, for which $\mathbf{i}^{\mathrm{opr}} = \mathbf{0}$ without any modifications to $\mathbf{Q}$, it may be useful for more general two-graph matching problems in which the linear bias term is having a detrimental effect. Such a problem is illustrated in the following example, in which two random 10-node graphs are to be matched. Figure 6(a) shows the decomposition of the solution found by a Hopfield network using the original objective function $E^{\mathrm{op}}_{\mathrm{o}}$. The values plotted in rows 1 to 10 give the elements of $\mathbf{A}^{\mathrm{s}}$, while the extra value plotted at position (0,0) gives the component of $\mathbf{v}$ along $\mathbf{i}^{\mathrm{opr}}$: ie. it measures $(\mathbf{v}^T\mathbf{i}^{\mathrm{opr}})/\,||\mathbf{i}^{\mathrm{opr}}||$. It is apparent that the solution obtained with the original formulation contains a significant component along $\mathbf{i}^{\mathrm{opr}}$, which is not surprising since it is $\mathbf{i}^{\mathrm{opr}}$ which guides $\mathbf{v}$ in the early stages of convergence. Figures 6(b) and 6(c) show the solutions obtained using the alternative objective function with θ set as in (83), starting once with $\mathbf{v}^{\mathrm{val}} = \mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$ and then again with $\mathbf{v}^{\mathrm{val}} = -\mathbf{v}^{\mathrm{val}}_{\mathrm{o}}$. The values plotted at (0,0) give the component of $\mathbf{v}$ along the $\mathbf{i}^{\mathrm{opr}}$ of the *original* mapping, though there is no actual linear bias for this alternative mapping. Both alternative solutions are superior to the original solution, and both exhibit smaller components along the original linear bias term $\mathbf{i}^{\mathrm{opr}}$. In particular, the best solution, seen in Figure 6(b), has a component along $\mathbf{i}^{\mathrm{opr}}$ less than half that of the original solution in Figure 6(a), and it is most unlikely that this solution could have been found using the original objective function with $\mathbf{i}^{\mathrm{opr}} \neq \mathbf{0}$.

However, it is by no means certain that removing $\mathbf{i}^{\mathrm{opr}}$ will always improve the solution quality; an improvement can be expected only when the auxiliary linear problem is misleading in the context of the parent problem. Removing $\mathbf{i}^{\mathrm{opr}}$ can indeed have a detrimental effect, since there is no guarantee that the alternative objective will be any more suited to the underlying discrete problem than the original objective. In an experiment, 1000 pairs of 10 node random graphs were generated, and then matched using a Hopfield network running with both the original and alternative objective functions. The results are summarized in Table 2: for the alternative objective



(a) Original objective,
$E_{\mathrm{o}}^{\mathrm{op}} = -51.16$



(b) Alternative objective,
initial $\mathbf{v}^{\mathrm{val}} = \mathbf{v}_{\mathrm{o}}^{\mathrm{val}}$,
$E_{\mathrm{o}}^{\mathrm{op}} = -51.75$

(c) Alternative objective
initial $\mathbf{v}^{\mathrm{val}} = -\mathbf{v}_{\mathrm{o}}^{\mathrm{val}}$,
$E_{\mathrm{o}}^{\mathrm{op}} = -51.42$

Figure 6: $\mathbf{A}^{\mathrm{s}}$ matrices and $\mathbf{i}^{\mathrm{opr}}$ components for the alternative solutions to the two-graph matching problem.

Problem subset	**Frequency**	**Mean $E_{\mathrm{o}}^{\mathrm{op}}$ with $\mathbf{i}^{\mathrm{opr}}$**	**Mean $E_{\mathrm{o}}^{\mathrm{op}}$ without $\mathbf{i}^{\mathrm{opr}}$**
Removing $\mathbf{i}^{\mathrm{opr}}$ detrimental	599	-13.5	-13.2
Removing $\mathbf{i}^{\mathrm{opr}}$ beneficial	328	-13.1	-13.4
Removing $\mathbf{i}^{\mathrm{opr}}$ has no effect	73	-13.1	-13.1
All problems	1000	-13.4	-13.3

Table 2: Effect of removing $\mathbf{i}^{\mathrm{opr}}$ in 1000 random two-graph matching problems.

Problem subset	**Frequency**	**Mean $E_{\mathrm{o}}^{\mathrm{op}}$ with $\mathbf{i}^{\mathrm{opr}}$**	**Mean $E_{\mathrm{o}}^{\mathrm{op}}$ without $\mathbf{i}^{\mathrm{opr}}$**
Removing $\mathbf{i}^{\mathrm{opr}}$ detrimental	645	-14.0	-13.8
Removing $\mathbf{i}^{\mathrm{opr}}$ beneficial	197	-13.5	-14.0
Removing $\mathbf{i}^{\mathrm{opr}}$ has no effect	158	-14.1	-14.1
All problems	1000	-13.9	-13.9

Table 3: Effect of removing $\mathbf{i}^{\mathrm{opr}}$ in 1000 Euclidean two-graph matching problems.

function, the best of the two possible solutions is represented in the statistics. We see that there is a subset of 599 problems for which the original objective is better, a subset of 328 problems for which the alternative objective is better, and a subset of 73 problems for which both objectives perform equally well. On average, the original objective outperforms the alternative objective by a narrow margin. However, the fact remains that in about 33% of the problems, the removal of $\mathbf{i}^{\mathrm{opr}}$ was beneficial. The same experiment was carried out with 1000 Euclidean graphs, where the edge weights represent the Euclidean distance between the two nodes in a plane (such a graph has applications in invariant pattern recognition [10, 11, 5, 17]). The results, summarized in Table 3, show that while the removal of $\mathbf{i}^{\mathrm{opr}}$ was beneficial in only 20% of the cases, the improvement in solution quality for these cases was more significant than for the random graphs, resulting in the average performance of the two objective functions being virtually identical. While it is unfortunate that there is no clear first choice of objective function, it is important to bear in mind that alternative objectives exist, and that the alternatives must be fully explored to be sure of finding the best possible solution.

7.2 Influencing the annealing

If we are to reserve modifications to $\mathbf{Q}$ for the purpose of removing linear bias terms, then the alternative objective functions (79) still offer us one more degree of freedom, that being to add a multiple of the identity matrix to the leading diagonal of $\mathbf{P}$. While this has no effect on the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$, it does modify the eigenvalues. Recalling from Section 6 that the course of an annealing process relies heavily on the ordering of these eigenvalues, it is clear that we can use this remaining degree of freedom to influence the annealing.

As an example, let us consider the Euclidean travelling salesman problem of Section 6.3. By adding the matrix $\mathbf{D}^{\mathrm{p}} = -\mathbf{I}^n$ to $\mathbf{P}$, the reduced eigenvalue matrix becomes

$$\zeta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.96 & 0.75 & -0.75 & -1.96 & -2.42 \\ 0 & 0.07 & 0.03 & -0.03 & -0.07 & -0.08 \\ 0 & -0.74 & -0.28 & 0.28 & 0.74 & 0.92 \\ 0 & -1.00 & -0.38 & 0.38 & 1.00 & 1.24 \\ 0 & -1.23 & -0.47 & 0.47 & 1.23 & 1.52 \\ 0 & -1.38 & -0.53 & 0.53 & 1.38 & 1.71 \\ 0 & -1.42 & -0.54 & 0.54 & 1.42 & 1.76 \\ 0 & -1.44 & -0.55 & 0.55 & 1.44 & 1.78 \\ 0 & -1.46 & -0.56 & 0.56 & 1.46 & 1.80 \end{bmatrix} \qquad \mathrm{order}(\zeta) = \begin{bmatrix} - & - & - & - & - & - \\ - & 1 & 15 & 37 & 44 & 45 \\ - & 24 & 25 & 26 & 27 & 28 \\ - & 36 & 29 & 23 & 16 & 14 \\ - & 38 & 30 & 22 & 13 & 11 \\ - & 39 & 31 & 21 & 12 & 6 \\ - & 40 & 32 & 20 & 10 & 5 \\ - & 41 & 33 & 19 & 9 & 4 \\ - & 42 & 34 & 18 & 8 & 3 \\ - & 43 & 35 & 17 & 7 & 2 \end{bmatrix}$$

Note that this gives a very different eigenvalue ordering compared with the original objective function in Section 6.3, and so we would expect the annealing to follow a correspondingly different course. Figure 7 shows the result of annealing on this energy function. It is apparent that a different solution has been achieved: in fact, this solution is poorer, with a tour length of 3.07, compared with 2.96 for the original objective.

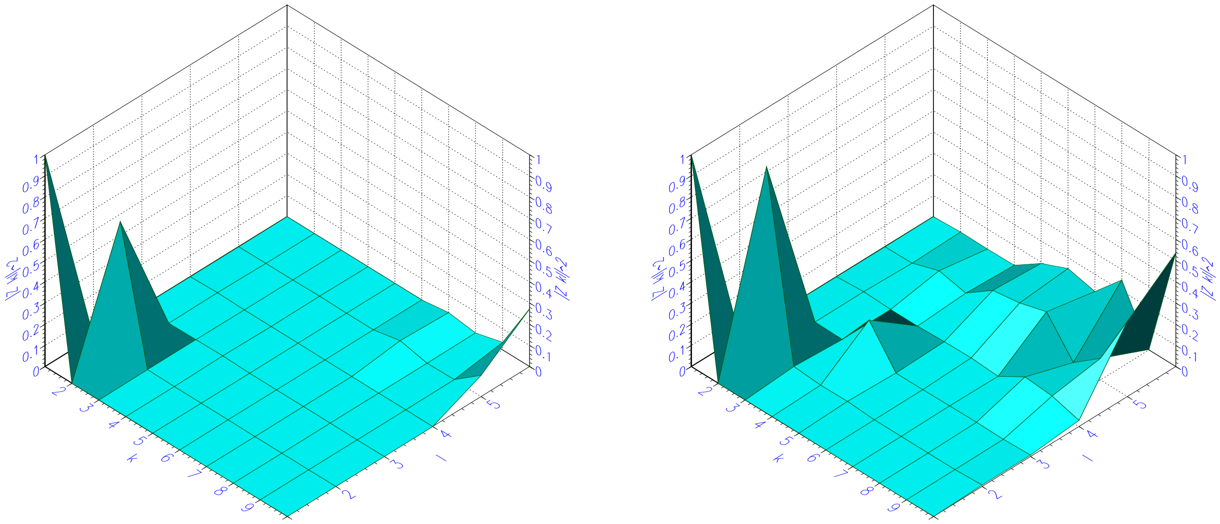
Given that by adding different multiples of $\mathbf{I}^n$ to $\mathbf{P}$ we can realize an infinite number of alternative objective functions, the obvious question is which one is it best to use? Of the two we have investigated so far, one took the network to the optimal solution, while the other achieved a poorer solution. We are really asking *which* continuous energy function leads most directly to the global optimum of the discrete problem. To answer this question, we require some knowledge of the likely structure of the optimal solution relative to the eigenvector basis of $\mathbf{T}^{\mathrm{opr}}$. Given the mathematical intractability of the combinatorial optimization problems in question, it is highly unlikely that this information can be obtained analytically. Instead we must appeal to experimental means.

8 Investigating properties of the optimal solution

While exhaustive search for the optimal solutions to combinatorial optimization problems is generally impractical, it *is* a reasonable undertaking for small problems, and if we take the properties of the small problems to be typical of larger problems of the same class, then the results should be very useful. To this end 1000 Euclidean 10 city TSPs were generated, and exhaustive search was applied to each problem to find the optimal solution. The same process was carried out for 1000 random 10 city TSPs (by a *random* TSP, we mean a TSP for which the distance matrix does not correspond to any set of points in a 2D plane, but is an arbitrary, symmetric square matrix). Furthermore, the $\mathbf{Z}^{\mathrm{s}}$ matrices for all the optimal solutions were calculated, along with their mean and standard deviation across the ensemble of 1000 problems; the results are summarized in Figures 8 and 9.

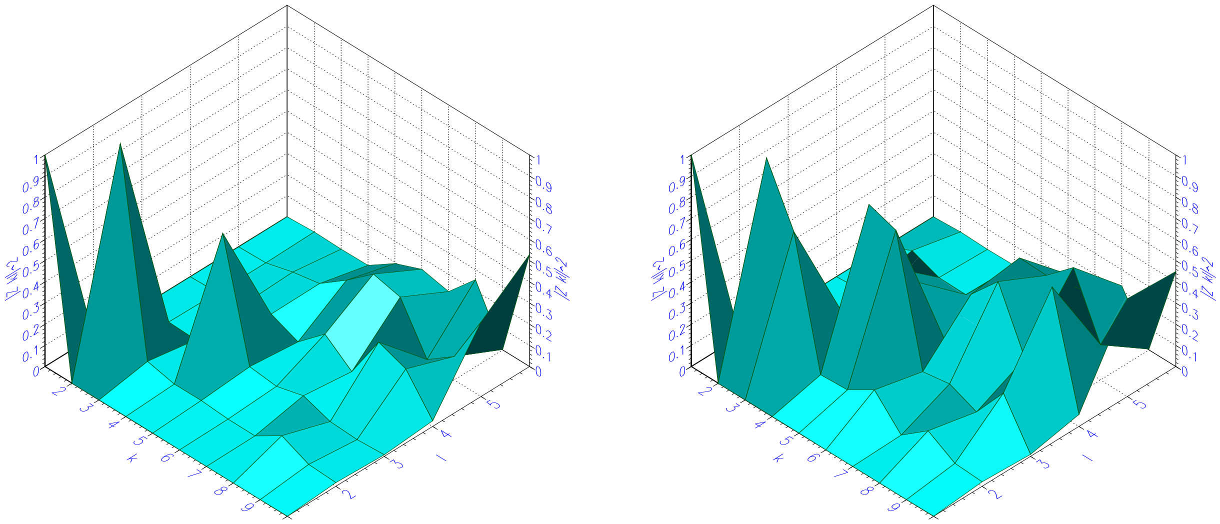
Looking first at the Euclidean problems (Figure 8), we see that the optimal solutions' decompositions, on average, bear a close resemblance to the $\mathbf{Z}^{10}$ matrix. The resemblance is closest around the largest elements, to be found in the (2,2) corner, where the standard deviations are also relatively low. So it seems that the optimal solution *consistently* has large components $[\mathbf{Z}^{\mathrm{s}}]_{22}$ and $[\mathbf{Z}^{\mathrm{s}}]_{32}$, while the activity towards the (10,6) corner of $\mathbf{Z}^{\mathrm{s}}$ becomes less predictable. Hence, if we were to arrange the annealing so that the elements in the (2,2) corner were introduced first, and those in the (10,6) corner later, we would expect an optimizing network to perform very well on these problems. By the time the network has introduced the $[\mathbf{Z}^{\mathrm{s}}]_{22}$ and $[\mathbf{Z}^{\mathrm{s}}]_{32}$ components, $\mathbf{v}$ would already be in a very good region of state space, with a good chance of finding the optimal hypercube corner by local descent. Conversely, is we arranged the annealing so that the elements in the (10,6) corner were introduced first, then $\mathbf{v}$ is likely to enter a poor region of state space from the outset, since the optimal solution has an unpredictable structure in the directions along which $\mathbf{v}$ initially evolves: the resulting solution is likely to be poor. These arguments are in good agreement with the experimental findings of Sections 6.3 and 7.2.

Turning now to the random problems (Figure 9), we see a more diffuse matrix $\mathbf{Z}^{\mathrm{s}}$, though still not wholly unlike the $\mathbf{Z}^{10}$ matrix. The standard deviations of all the significant elements of $\mathbf{Z}^{\mathrm{s}}$



(a) $\beta = -1.5$

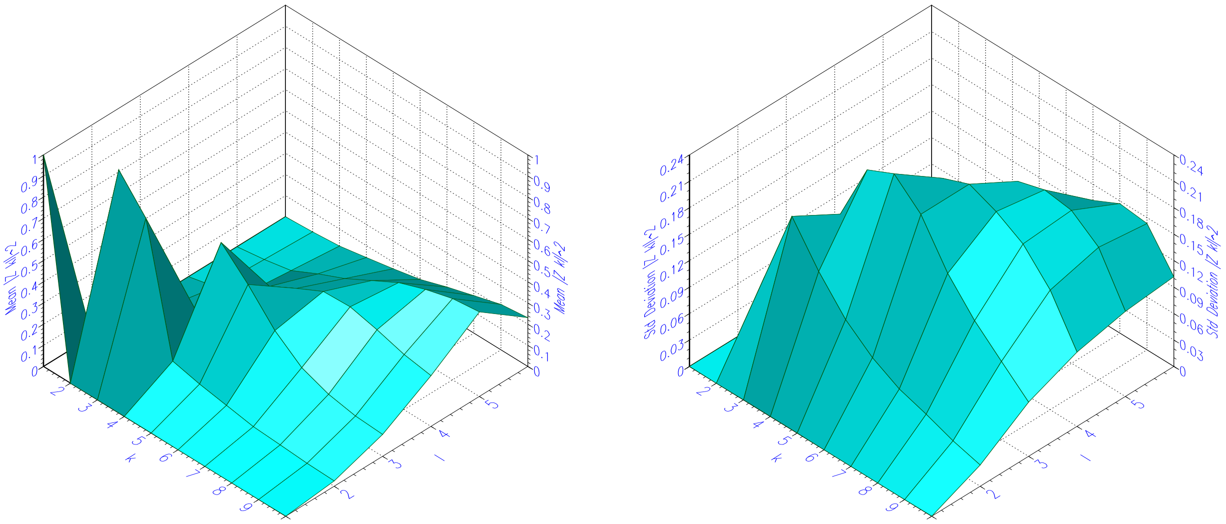
(b) $\beta = -0.3$



(c) $\beta = -0.1$

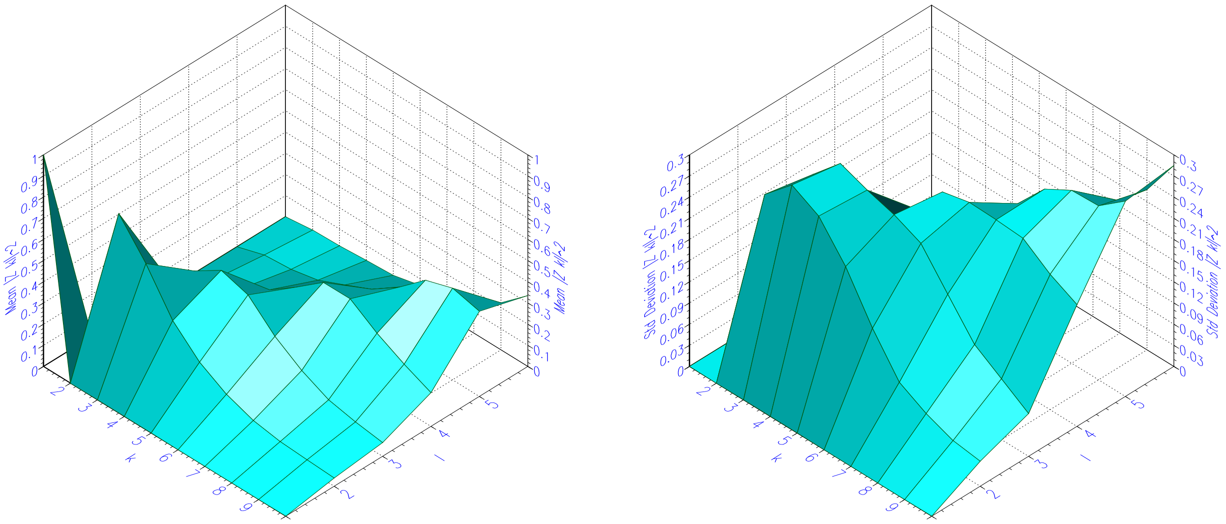
(d) $\beta = 0.0$ (Fully converged)

Figure 7: $\mathbf{Z}^s$ matrix for the 10 city travelling salesman problem at several values of β with the alternative energy function.



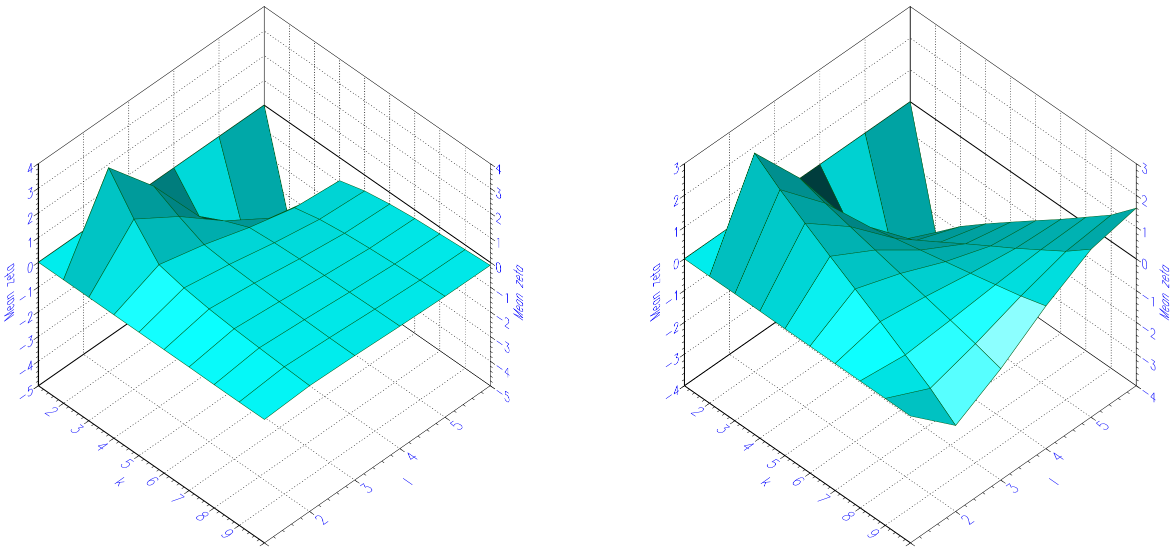
(a) Mean of $\mathbf{Z}^s$ (b) Standard deviation of $\mathbf{Z}^s$

Figure 8: Mean and standard deviation of the $\mathbf{Z}^s$ matrix for the optimal solutions to 1000 Euclidean TSPs.



(a) Mean of $\mathbf{Z}^s$ (b) Standard deviation of $\mathbf{Z}^s$

Figure 9: Mean and standard deviation of the $\mathbf{Z}^s$ matrix for the optimal solutions to 1000 random TSPs.



(a) Mean ζ for Euclidean problems

(b) Mean ζ for random problems

Figure 10: Mean reduced eigenvalue matrices ζ for the TSPs without modification to $\mathbf{P}$.

are relatively high, indicating that the optimal solutions have a far less predictable structure than those of the Euclidean problems. Indeed, the variability of the optimal solutions' decompositions along the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$ can be taken as a good indicator of the difficulty of solving the problems with optimizing networks, since such networks will always converge to a solution with a fairly predictable decomposition along these eigenvectors (in this case the $\mathbf{Z}^{10}$ matrix). So it seems that the random TSP is going to be an altogether more difficult problem to tackle. In particular, there is no direction along which the optimal solution has a large component with low variance, and so there is no corresponding criterion for selecting a suitable annealing process. Nevertheless, since the variances in the (2,2) corner are slightly lower than those in the (10,6) corner, we will attempt to arrange the annealing so that these elements are introduced first.

For the planar Euclidean travelling salesman problem, there is strong experimental evidence that if the elements on the leading diagonal of $\mathbf{P}$ are all set to zero, then the λ_k are all positive [1, 10]; this is presumably a consequence of the constraining effect of the Euclidean geometry on the distance matrix $\mathbf{P}$. This ensures that, without any modifications to the leading diagonal of $\mathbf{P}$, the elements of $\mathbf{Z}^{\mathrm{s}}$ will be introduced in the desired order, that is (2,2) corner first. This is confirmed in Figure 10(a), where we see the mean of the reduced eigenvalue matrices ζ for the 1000 Euclidean problems with $[\mathbf{P}]_{ii} = 0$: the ordering of the eigenvalues is suitable for introducing the $[\mathbf{Z}^{\mathrm{s}}]_{kl}$ in the required order. Figure 10(b) shows the mean of the reduced eigenvalue matrices for the random problems, again with all the elements on the leading diagonal of $\mathbf{P}$ set to zero. While this is suitable for introducing the elements in the (2,2) corner of $\mathbf{Z}^{\mathrm{s}}$ first, considering the high variance of the optimal solutions' decompositions in these directions, some experimentation with the eigenvalue ordering (via the leading diagonal of $\mathbf{P}$) might be beneficial.

9 Testing the theory

The results of experiments on the TSP problem sets are displayed in Table 4. Each problem was solved using the Hopfield network and MFA, and an alternative energy function was tried out for the random TSPs. The Hopfield network was simulated using direct projection onto the valid

Problem	$[\mathbf{P}]_{ii}$	**Mean HN error**	**Std. dev. of HN error**	**Mean MFA error**	**Std. dev. of MFA error**	**No. valid for MFA**
Euclidean	0.0	0.71%	1.64%	1.36%	2.39%	987
Random	0.0	13.5%	13.7%	9.85%	11.1%	988
Random	-0.2	11.1%	13.2%	10.5%	11.2%	985

Table 4: Results of experiments on the Euclidean and random problem sets.

subspace, as in Figure 2, thus ensuring that valid solutions were obtained 100% of the time. Neuron normalization [1, 20, 23] was used in conjunction with the MFA algorithm to directly enforce one of the two validity conditions. This left two parameters to set: the weighting given to the remaining penalty term in the energy function, and the amount of self-feedback used to prevent instability. These parameters were manually adjusted to extract the best possible performance from the MFA algorithm, without obtaining an unacceptably large number of invalid solutions. Annealing was started at a sufficiently high value of T^p (or low value of β) to ensure that the network initially stabilized at the centre of the valid subspace. The ensuing annealing schedule was made deliberately slow, in order to tilt the balance of the compromise towards high quality solutions, and away from rapid convergence rates.

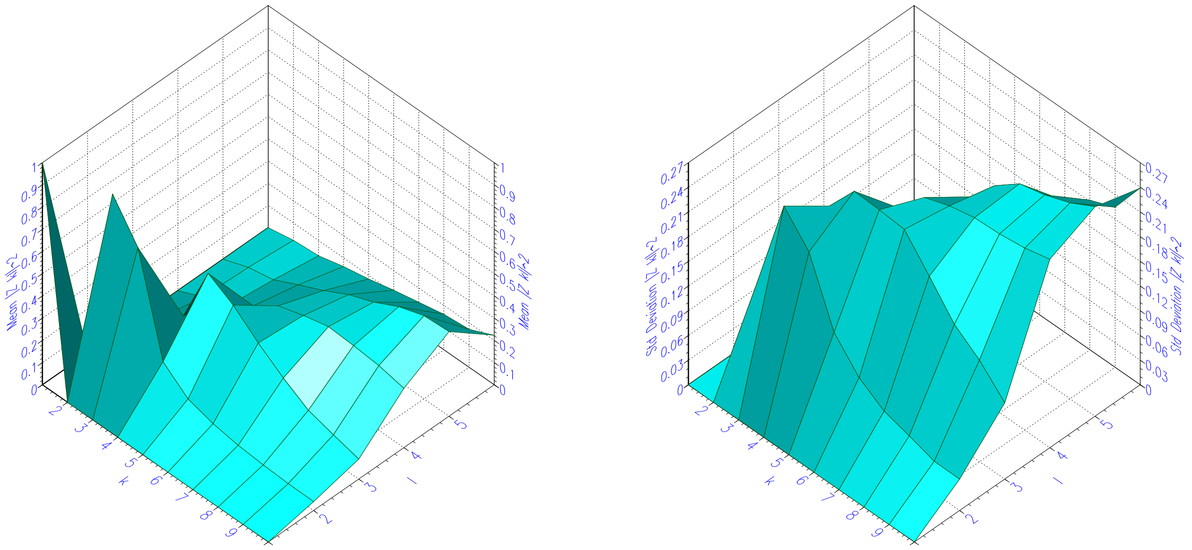
As expected, performance on the Euclidean problems was excellent, with the average solution quality being about 1% suboptimal. Performance on the random problems was significantly poorer, with average tour lengths being about 10% longer than the optimal tour. Note that the use of an alternative energy function to influence the annealing considerably improved the performance of the Hopfield network, though slightly degraded the MFA results.

In order to verify the cause of failure for the random TSPs, it would be useful to study the solutions which the optimizing networks found to these problems. Figures 11 and 12 show the mean and standard deviations of the $\mathbf{Z}^{\mathrm{s}}$ matrices for the solutions obtained by the networks to the random TSPs, using the original energy functions with $[\mathbf{P}]_{ii} = 0$. Both mean decompositions resemble the $\mathbf{Z}^{10}$ matrix much more closely than the decomposition of the optimal solutions (Figure 9(a)), indicating that the networks are not capable of finding solutions which deviate from this form by a large degree. The eigenvalue ordering for the random problem was such that the elements in the (2,2) corner of $\mathbf{Z}^{\mathrm{s}}$ were introduced first, and we see that the standard deviations in these directions are correspondingly low, especially when compared to those of the optimal solutions (Figure 9(b)). What is somewhat obscured in the plots is the value of the standard deviation of $[\mathbf{Z}^{\mathrm{s}}]_{22}$: 0.23 for the optimal solutions, 0.11 for the Hopfield network solutions and 0.14 for the MFA solutions. Hence both networks consistently (ie. with low variance) introduce a large component $[\mathbf{Z}^{\mathrm{s}}]_{22}$, as expected from considerations of the dynamics, whereas the optimal solutions exhibit considerable variance in this direction. As this is the first component introduced by the network, any error will take the state vector into a poor region of state space from the start, leading most probably to a poor solution. Hence, it is likely that this constitutes the main cause of failure on the random problems. We could not expect to do any better by arranging the annealing so that some other element of $\mathbf{Z}^{\mathrm{s}}$ is introduced first, since the optimal solutions exhibit even greater variance along any such alternative direction.

10 Discussion and Conclusions

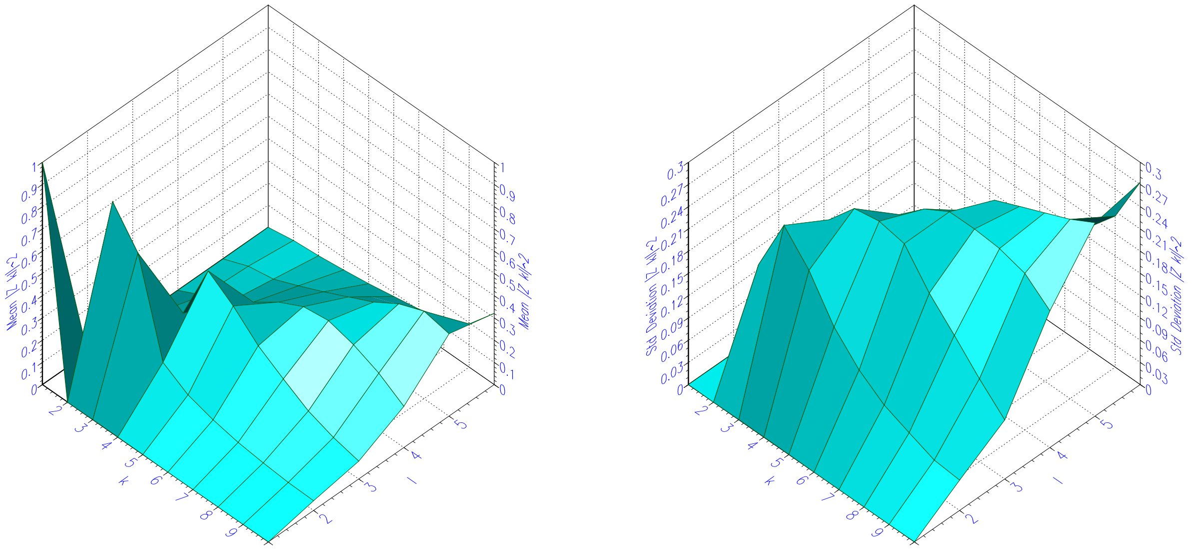
While recent trends have been towards using optimizing networks to solve ever more useful and complex problems, it is most important to note the network's failure to perform well on a problem as small as a 10 city random TSP. There is nothing particularly special about the random TSP; we can expect the solutions to other problems to have just as undesirable properties relative to the eigenvector basis of $\mathbf{T}^{\mathrm{opr}}$.

The use of alternative energy functions can potentially improve the network's performance on such problems. In this paper, we have demonstrated that such alternative functions exist for the



(a) Mean of $\mathbf{Z}^s$ (b) Standard deviation of $\mathbf{Z}^s$

Figure 11: Mean and standard deviation of the $\mathbf{Z}^s$ matrix for the Hopfield network solutions to 1000 random TSPs.



(a) Mean of $\mathbf{Z}^s$ (b) Standard deviation of $\mathbf{Z}^s$

Figure 12: Mean and standard deviation of the $\mathbf{Z}^s$ matrix for the MFA solutions to 1000 random TSPs.

two-graph matching problem, and we have described how the structure of the network's solution can be predicted by examining the effect of the specific function being employed. We have also demonstrated how alternative energy functions *can* improve the solution quality of the network, but it has seemed difficult to predict *which* energy function will perform best without some prior knowledge of the structure of the optimal solution.

Even with such knowledge about the optimal solution, we have seen for the random TSPs that none of the alternative energy functions could dramatically improve the performance, since the optimal solutions exhibit considerable variance when decomposed along the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$. We could in this way distinguish between 'hard' problems, like the random TSP, and 'easy' problems, like the Euclidean TSP. 'Easy' problems have optimal solutions with a predictable structure relative to the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$, at least along those eigenvectors which are most prominent in the decompositions. For 'hard' problems, however, there are no eigenvectors along which the optimal solutions have predictably large components, and so there is no way in which we can arrange the annealing to rapidly guide $\mathbf{v}$ into a good region of state space.

In conclusion, the main contributions of this paper have been to identify a significant cause of poor performance in optimizing neural networks, and to draw attention to degrees of freedom in the problem mappings which have not been previously examined. Bearing in mind the networks' performance on the random TSPs, it would seem a little rash to use such networks to solve more complex problems for which a high quality solution is required, without first performing a detailed study of these problems' solutions to demonstrate the applicability of the network technique. However, optimizing neural networks would seem to present a sensible compromise between solution quality and speed when applied to a wide range of problems, especially when the potential for fast, parallel implementations is considered.

References

[1] S. V. B. Aiyer. Solving combinatorial optimization problems using neural networks. Technical Report CUED/F-INFENG/TR 89, Cambridge University Engineering Department, October 1991.

[2] S. V. B. Aiyer and F. Fallside. A Hopfield network implementation of the Viterbi algorithm for Hidden Markov Models. In *Proceedings of the International Joint Conference on Neural Networks*, pages 827–832, Seattle, July 1991.

[3] S. V. B. Aiyer, M. Niranjan, and F. Fallside. A theoretical investigation into the performance of the Hopfield model. *IEEE Transactions on Neural Networks*, 1(2), June 1990.

[4] B. Angéniol, G. de la Croix Vaubois, and J. le Texier. Self-organizing feature maps and the travelling salesman problem. *Neural Networks*, 1:289–293, 1988.

[5] E. Bienenstock and R. Doursat. Elastic matching and pattern recognition in neural networks. In L. Personnaz and G. Dreyfus, editors, *Neural Networks: From Models to Applications.* IDSET, Paris, 1989.

[6] S. P. Eberhart, D. Daud, D. A. Kerns, T. X. Brown, and A. P. Thakoor. Competitive neural architecture for hardware solution to the assignment problem. *Neural Networks*, 4:431–442, 1991.

[7] F. Favata and R. Walker. A study of the application of Kohonen-type neural networks to the travelling salesman problem. *Biological Cybernetics*, 64:463–468, 1991.

[8] C. G. Fox and W. Furmanski. Load balancing loosely synchronous problems with a neural network. In *Proceedings of the 3rd Conference on Hypercube Concurrent Computers and Applications*, Pasedena, 1988.

[9] B. Fritzke and P. Wilke. FLEXMAP — a neural network for the traveling salesman problem with linear time and space complexity. In *Proceedings of the International Joint Conference on Neural Networks*, Singapore, November 1991.

[10] A. H. Gee, S. V. B. Aiyer, and R. W. Prager. Neural networks and combinatorial optimization problems — the key to a successful mapping. Technical Report CUED/F-INFENG/TR 77, Cambridge University Engineering Department, July 1991.

[11] A. H. Gee, S. V. B. Aiyer, and R. W. Prager. A subspace approach to invariant pattern recognition using Hopfield networks. In *Proceedings of the International Joint Conference on Neural Networks*, Singapore, November 1991.

[12] L. Gislen, C. Peterson, and Söderberg B. 'Teachers and Classes' with neural networks. *International Journal of Neural Systems*, 1(2), 1989.

[13] A. Graham. *Kronecker Products and Matrix Calculus, with Applications.* Ellis Horwood Ltd., Chichester, 1981.

[14] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. USA*, 81:3088–3092, May 1984.

[15] J. J. Hopfield and D. W. Tank. 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[16] B. Kamgar-Parsi and B. Kamgar-Parsi. On problem solving with Hopfield neural networks. *Biological Cybernetics*, 62:415–423, 1990.

[17] W. Li and N. M. Nasrabadi. Object recognition based on graph matching implemented by a Hopfield-style neural network. In *Proceedings of the International Joint Conference on Neural Networks*, Washington DC, 1989.

[18] E. Mjolsness and C. Garrett. Algebraic transformations of objective functions. *Neural Networks*, 3:651–669, 1990.

[19] R. G. Ogier and D. A. Beyer. Neural network solution to the link scheduling problem using convex relaxation. In *Proceedings of the IEEE Global Telecommunications Conference*, San Diego, 1990.

[20] C. Peterson and B. Söderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1), 1989.

[21] P. D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network*, 1:89–103, 1990.

[22] D. W. Tank and J. J. Hopfield. Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 33(5), May 1986.

[23] D. E. Van den Bout and T. K. Miller III. Graph partitioning using annealed neural networks. *IEEE Transactions on Neural Networks*, 1(2), June 1990.

[24] V. Wilson and G. S. Pawley. On the stability of the TSP problem algorithm of Hopfield and Tank. *Biological Cybernetics*, 58:63–70, 1988.