SOLVING COMBINATORIAL OPTIMIZATION PROBLEMS USING NEURAL NETWORKS WITH APPLICATIONS IN SPEECH RECOGNITION

Sreeram V. Balakrishnan-Aiyer

Trinity College
Cambridge
CUED/F-INFENG/TR.89

This dissertation is submitted for consideration towards the degree of Doctor of Philosophy at the University of Cambridge.

October 1st, 1991

Abstract

Combinatorial optimization problems arise naturally in many areas of science and engineering. Unfortunately, the accurate solution of a large class of these problems requires an amount of computation which increases exponentially with the problem size. The extensive research into ways of avoiding this difficulty has recently been given even further impetus by the widespread occurrence of these problems in the field of Artificial Intelligence, especially in the pattern recognition problems associated with speech and vision processing. However, the only computationally tractable methods that have so far been developed all rely on some form of heuristic: essentially an intelligent guess which offers a reasonable compromise between solution quality and computational complexity.

Neural networks offer a novel and potentially powerful heuristic for solving these problems. They are also intrinsically parallel systems, with significant potential for fast hardware implementation. Out of recent research, two related families of neural network for solving combinatorial optimization problems have emerged: Hopfield networks [23, 24, 25], and Mean Field Annealing networks [43, 46]. A number of researchers has applied these networks to a wide variety of problems [48, 43, 37, 26, 46, 32, 16], and there now exists a large body of rather inconclusive experimental results, with several researchers [48, 26] reporting very poor performances. No theoretical framework has yet been developed which can either account for these inconclusive results in a systematic way, or offer a robust method of correcting the root causes of these poor performances. Consequently, some researchers have concluded that these networks are fatally flawed [48].

This thesis is aimed at developing such a theoretical framework. By doing this, it will be possible to explain and correct the poor network performances that have emerged from the experimental results. This framework also has the further advantage of making it possible to analyze how these networks achieve their solutions, and to characterize which types of problems these networks can be expected to solve accurately. Based on this analysis, a series of modifications will then be described which greatly improve the efficiency and final solution quality of these networks. Finally a mapping of the Viterbi algorithm onto these network will be developed, which will be used to demonstrate that combinatorial optimization problems of practical significance to speech recognition can be solved using neural networks.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Frank Fallside, for giving me the support and freedom to develop my own ideas and pursue them to a fruitful conclusion. To my parents I owe an enormous debt, for without their encouragement and enthusiasm, life would have been a very different story. I wish to thank Andrew Gee for his numerous stimulating discussions and especially for his Herculean efforts in proof reading this thesis. I am also indebted for all the help, large and small, that I have received from members (it would be unfair to name individuals) of the Speech, Vision and Robotics group. Lastly I wish to thank the Science and Engineering Research Council and Trinity College for their financial support.

Declaration

Apart from chapter 1, the rest of this thesis is, unless other authors are specifically cited, my own original work done between October 1989 and September 1991. This thesis does not include any work done in collaboration, and no part of it has been submitted to any other institution towards obtaining a degree. Although chapter 1 draws on the work of several authors, these works have been combined in an original way to develop a unified theoretical approach. Significant parts of chapters 2, 3, 4, 5 and 6 are taken from the following papers of mine that have been published, or have been submitted for publication, over the last two years.

S. V. Balakrishnan Aiyer, M. Niranjan, & F. Fallside, *A Theoretical Investigation into the Performance of the Hopfield Model*, IEEE Transactions on Neural Networks, Vol. 1, No. 2, June 1990.

S. V. Balakrishnan Aiyer, M. Niranjan, & F. Fallside, *On the Performance of the Hopfield Model*, Proc. INNS Paris, July 1990.

S. V. Balakrishnan Aiyer & F. Fallside, *A Subspace Approach to Solving Combinatorial Optimization Problems with Hopfield Networks*, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR 55, December 1990.

S. V. Balakrishnan Aiyer & F. Fallside, *A Hopfield Network Implementation of the Viterbi Algorithm for Hidden Markov Models*, Proc. IJCNN-91 Seattle, July 1991.

Contents

List of Figures

List of Tables

List of Notation

General Conventions

Matrices are denoted using upper case bold letters, and vectors are denoted using lower case bold letters. All superscripts in lower case italics denote the dimension of a vector or matrix or the number of an eigenvector. Superscripts in normal lower case denote a label for a vector or matrix, and subscripts in lower case italics are used to pick out the elements of a vector or matrix. Eigenvalues are denoted using lower case greek letters. Where appropriate, the section in which the detailed definition of a particular symbol is given, will be listed in the third column of the following tables.

General Definitions

Symbol	Description	Refer to
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$	
H	Hamiltonian of a Mean Field Annealing network	1.3
E	Energy of a Hopfield network	1.3
$\langle\ \rangle$	Operator which denotes expected value	1.3
vec()	Function which maps a matrix onto a vector	2.1
λ_k	Eigenvalue of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$	3.1
γ_l	Eigenvalue of $\mathbf{R}^m\mathbf{Q}\mathbf{R}^m$	3.1
$\otimes$	Kronecker Product	2.1
$\nabla_{\mathbf{v}}$	Gradient operator w.r.t. $\mathbf{v}$. Equivalent to $[\frac{\partial}{\partial v_1}, \frac{\partial}{\partial v_2}, \ldots, \frac{\partial}{\partial v_n}]$	
T^p	Temperature in Mean Field Annealing networks	1.3
χ_{kl}	Eigenvalue of $\mathbf{T}^{\mathrm{opr}}$	3.1
$\tilde{\chi}_{kl}$	Eigenvalue of $\mathbf{T}^{\mathrm{lin}}$	3.3.2
κ	Matrix Graduated Non-Convexity scalar	5.1.2
θ	Constraint energy term scalar	2.4

Acronyms

Acronym	Full meaning	Refer to
EM	Energy-Minimization	1
GPP	Graph Partitioning Problem	1.1
MFA	Mean Field Annealing	1.3
MGNC	Matrix Graduated Non-Convexity	5.1.2
TSP	Travelling Salesman Problem	

Superscripts

Superscript	Generally used to represent:
op	Optimization term
cn	Constraint term
lin	Linearized dynamics term
opr	Optimization term accounting for valid subspace confinement
zs	Zerosum subspace component
nz	Nonzero subspace component
nzo	Nonzero-one subspace component
o	Component along vector of ones

Vectors

Vector	Description	Refer to
$\mathbf{h}^l$	Eigenvector of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$	3.1
$\hat{\mathbf{h}}$	Complex Eigenvector of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$ (TSP only)	4.1.2
$\mathbf{i}^{\mathrm{b}}$	Network input bias vector	1.3
$\mathbf{i}^{\mathrm{cn}}$	Constraint input bias vector term	1.4
$\mathbf{i}^{\mathrm{lin}}$	Linearized input bias vector term	3.3.2
$\mathbf{i}^{\mathrm{op}}$	Optimization input bias vector term	1.4
$\mathbf{i}^{\mathrm{opr}}$	Valid subspace optimization input bias vector term	2.5.2
$\mathbf{o}$	Vector of ones - dimension given as superscript	2.1
$\mathbf{p}$	Permutation vector	1.2
$\mathbf{s}$	Valid subpace origin offset term	2.2
$\mathbf{u}$	Network state vector	1.3
$\mathbf{v}$	Network output vector	1.3
$\mathbf{v}(\mathbf{p})$	Valid solution vector corresponding to permutation $\mathbf{p}$	1.2
$\mathbf{w}^k$	Eigenvector of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$	3.1
$\mathbf{x}^{kl}$	Eigenvector of $\mathbf{T}^{\mathrm{opr}}$	3.1

Matrices

Matrix	Description	Refer to
$\mathbf{A}$	Matrix of components of $\mathbf{v}$ along the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$	3.1.1
$\mathbf{B}$	Matrix of components of $\mathbf{i}^{\mathrm{opr}}$ along the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$	3.1.1
$\mathbf{I}^{nm}$	Identity Matrix - dimension given as superscript	2.1
$\mathbf{O}$	Matrix of ones - dimension given as superscript	2.1
$\mathbf{P}$	Kronecker product submatrix of $\mathbf{T}^{\mathrm{op}}$, where $\mathbf{T}^{\mathrm{op}} = (\mathbf{P} \otimes \mathbf{Q})$	2.5.1
$\mathbf{Q}$	Kronecker product submatrix of $\mathbf{T}^{\mathrm{op}}$, where $\mathbf{T}^{\mathrm{op}} = (\mathbf{P} \otimes \mathbf{Q})$	2.5.1
$\mathbf{R}$	Matrix which projects out the $\mathbf{o}^n$ component of a vector	2.1
$\mathbf{S}$	Matrix equivalent of vector $\mathbf{s}$	2.2
$\mathbf{T}$	Hopfield or Mean Field Annealing network connection matrix	1.3
$\mathbf{T}^{\mathrm{cn}}$	Constraint matrix term	1.4
$\mathbf{T}^{\mathrm{lin}}$	Linearised network connection matrix	3.3.2
$\mathbf{T}^{\mathrm{nz}}$	Nonzero subspace projection matrix	2.3
$\mathbf{T}^{\mathrm{nzo}}$	Nonzero-one subspace projection matrix	2.3
$\mathbf{T}^{\mathrm{o}}$	Matrix which projects onto vector $\mathbf{o}^{nm}$	2.3
$\mathbf{T}^{\mathrm{op}}$	Optimization matrix term	1.4
$\mathbf{T}^{\mathrm{opr}}$	Valid subspace optimization matrix term	2.5.2
$\mathbf{T}^{\mathrm{zs}}$	Zerosum subspace projection matrix also given by $(\mathbf{R}^n \otimes \mathbf{R}^m)$	2.2
$\mathbf{U}$	Matrix equivalent of network state vector $\mathbf{u}$	1.3
$\mathbf{V}$	Matrix equivalent of network output vector $\mathbf{v}$	1.3
$\mathbf{V}(\mathbf{p})$	Valid solution matrix corresponding to the permutation $\mathbf{p}$	1.2
$\mathbf{W}$	Matrix of normalized eigenvectors of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$	3.1.1
$\mathbf{X}$	Matrix of normalized eigenvectors of $\mathbf{R}^m\mathbf{Q}\mathbf{R}^m$	3.1.1
$\mathbf{Z}$	Matrix of complex components along $\mathbf{x}^{kl}$'s (TSP only)	4.2

Introduction

This thesis is concerned with solving combinatorial optimization problems using neural networks. As an illustrative introduction to these problems, consider the well known Travelling Salesman Problem (TSP). A local variant of this, the Travelling Scholar problem, together with a solution achieved with a neural network, is shown in Fig i.1. The objective is to find the shortest closed tour that visits 30 locations in Cambridge (27 Colleges, the Engineering Department, the University Library and my house), without visiting a location more than once. For this problem the number of possible tours is given by $\frac{n!}{2n}$, which for $n = 30$ gives 4.42×10^{30}. Thus the problem illustrates the key attributes of a combinatorial optimization problem; it has a discrete solution space (the set of all permutations of the integers $1 \cdots 30$), over which a cost function (in this case the corresponding tour length) has to be optimized, and for which there is no known algorithm that is guaranteed to find the optimum solution, apart from a computationally prohibitive exhaustive search.

This difficulty in finding the optimum solution is a characteristic of a whole class of combinatorial optimization problems which are known as $\mathcal{NP}$-hard [19]. For such problems no known algorithm exists that is guaranteed to find the optimum solution in polynomial time. In contrast, $\mathcal{P}$-hard combinatorial optimization problems, such as dynamic programming problems, can be optimally solved in polynomial time. Consequently the solution of the $\mathcal{NP}$-hard problems has been the subject of a vast amount of research. Inevitably, all the generally applicable techniques rely on some sort of heuristic, essentially an intelligent guess, to find good solutions in reasonable search times. The most common techniques range from the simple greedy algorithm to the more complex branch and bound search, A* search [49] and the Lin-Kernighan algorithm [28, 38] (for the TSP and GPP). A comprehensive account of such techniques for TSP type problems can be found in Lawler et al [31].

In recent years, novel algorithms drawing on more diverse sources have been proposed. The technique of Simulated Annealing [29] is based on a statistical mechanics analogy with cooling in matter, while the Genetic Algorithm [14] is based on the use of Darwinian evolution to continually modify and fine tune populations of candidate solutions. Both these techniques can, and have been, successfully applied to a wide variety of $\mathcal{NP}$-hard problems.

Neural networks offer a novel and potentially very powerful heuristic for solving these problems. They are also intrinsically parallel systems, with significant potential for fast hardware implementation. The origins of research in this area can be traced back to

The Travelling Scholar Problem

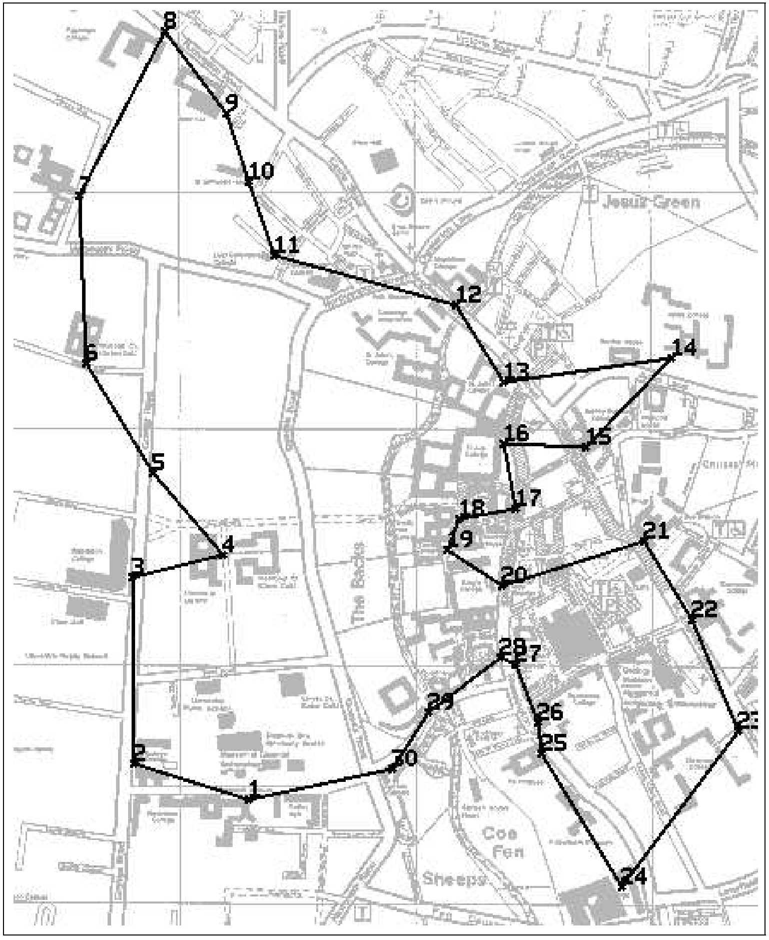


Figure i.1: The shortest tour linking 27 Cambridge Colleges,The Engineering Department, The University Library and my house. The solution shown was achieved using a continuous Hopfield network (see section 1.3.4).

the work of J.J.Hopfield [23, 24, 25], and to related work by several researchers on the statistical mechanics of Spin-Glass systems [6, 18, 27, 7, 42, 10, 45]. Out of this work, two related families of neural network for solving combinatorial optimization problems have emerged: Hopfield networks, and Mean Field Annealing networks.

So far the approach to using these networks has been mainly experimental; a number of researchers has simply taken one of these two families of networks, and applied them to a wide variety of problems [48, 43, 37, 26, 46, 32, 16]. As a consequence, there now exists a large body of rather inconclusive experimental results relating to the performance of these networks for solving combinatorial optimization problems. In some cases a reasonable degree of success has been reported [43, 46], whilst in others very poor results have been reported [48, 26]: the main problem being that the network converges to an output that does not correspond to a solution of the combinatorial optimization problem that is being solved. Such outputs, termed invalid solutions, can occur with almost 90% frequency [48], and even when the network converges to a valid solution, it is often of a poor cost. No theoretical framework has yet been developed which can either account for these inconclusive results in a systematic way, or offer a robust method of correcting the problem of convergence to invalid and poor cost solutions.

The aim of this thesis is to develop such a theoretical framework. Apart from explaining why invalid and poor cost solutions are produced, and how they can be avoided, a further advantage of this framework is that it can be used to:

(i) Create a more unified and simpler the mathematical analysis, than has traditionally been used for these networks.

(ii) Analyse how these networks achieve their solutions, and to characterise which types of problems these networks can be expected to solve accurately.

(iii) Specify a series of modifications which greatly improve the efficiency and final solution quality of these networks.

(iv) Demonstrate that combinatorial optimization problems of practical significance can be solved using these networks.

The overall objective is thus to provide a theoretical complement to the body of experiment results that has already been achieved, and by doing this, develop a framework with which neural networks can be reliably and accurately used to solve combinatorial optimization problems.

Outline of the thesis

Chapter 1 introduces some key combinatorial optimization problems and the main features of the neural networks that can be used to solve them. **Chapter 2** develops a new mathematical framework and ‘subspace’ approach, which are then used to explain and correct the problem of convergence to invalid and poor cost solutions. **Chapter 3** provides an in depth analysis of how these networks solve combinatorial optimization problems, together with a characterization of the features of a problem that allow good solutions to be achieved for it. This is followed by a detailed study in **Chapter 4** of the Euclidean Travelling Salesman problem, which provides a specific confirmation of the analysis in chapter 3. **Chapter 5** then uses the theory developed in the earlier chapters to propose a series of modifications to these networks which increase their efficiency and final solution quality. **Chapter 6** develops a mapping for the Viterbi algorithm onto these networks, which is then used to demonstrate their potential in solving realistic speech recognition problems. Finally **Chapter 7** presents the conclusions and proposals for future work.

Chapter 1

Combinatorial Optimization and Neural Networks

Only a small subset of neural networks are relevant to combinatorial optimization problems. For convenience these will be collectively referred to as Energy-Minimization (EM) networks, since they can all be considered as dynamic systems which minimize an energy function or Hamiltonian. The aim of this chapter is to present the two main classes of EM networks, the Hopfield network [25] and Mean Field Annealing networks [45, 43], and to describe how combinatorial optimization problems can be mapped onto them. A unified theoretical approach will be developed which is based on a combination of the works of J.J.Hopfield [24, 25], D. Van den Bout [45, 46], C. Peterson [42, 43] and P. Simic [44], and this will form the basis of the rest of the theoretical analysis in this dissertation.

1.1 Combinatorial Optimization Problems

It is necessary at this point to describe the features of the combinatorial optimization problems, other than the TSP, which are referred to in the rest of this thesis. Like many other combinatorial optimization problems, they all either arise directly from, or can be viewed as being isomorphic with, well known problems in Graph Theory.

The Hamilton Path problem

This problem is very similar to the TSP, the only difference being that the tour is not closed. In other words, the solution of this problem is the minimum length tour which visits each city once but which does not return to the start city. It is identical to the problem of finding the minimum spanning tree of a graph, where no vertex of the tree has a degree greater than two [31, p29]. A n-city (or n-node) Hamilton Path problem can also be mapped onto an equivalent $(n + 1)$-city TSP. This is achieved by introducing an extra city into the Hamilton Path problem, which is equidistant from all the others. The tour formed by removing this city from the TSP solution of the $(n+1)$-city problem is equivalent to the Hamilton Path solution of the original n-city problem [31, p22].

The Graph Partitioning problem (GPP)

As its name suggests, this problem arises directly from Graph Theory. A graph consists of a set of n nodes, $\{q_1, q_2, \ldots, q_n\}$, which are linked by a set of edges of weight $\{e_{11}, e_{12}, \ldots e_{nn}\}$. The problem consists of finding the minimum cost partitioning of the set of nodes into a number of disjoint sets. The cost corresponds to the sum of the weights of the edges that link nodes in different partitions. Usually the partitioned sets are required to be of equal size, although this is not always the case.

The Graph Matching Problem

This problem consists of finding a way of labelling (permuting) the nodes of one graph so that it optimally matchs another equally sized graph [1]. The measurement of the cost of the match is to some extent arbitrary, however the most commonly used measure is the sum of the products of the corresponding edges in the two graphs (i.e. the relabelled graph and the graph it is being matched with). Let $\mathbf{p}$ be a n-element permutation vector which specifies a relabelling of the elements of the first graph. Thus $q_i \rightarrow q_{p_i}$ and $e_{ij} \rightarrow e_{p_i p_j}$. If the edge weights of the second graph are denoted e'_{ij}, then the cost of the match descibed by $\mathbf{p}$ becomes,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} e_{p_i p_j} e'_{ij} \tag{1.1}$$

The solution of these problems has a practical application in vision processing, where several researchers have proposed solving pattern recognition problems by mapping them onto an equivalent Graph Matching problem [47, 9].

1.2 Representing the Solutions of Combinatorial Optimization Problems

Before combinatorial optimization problems can be mapped onto a neural network, a way has to be found of representing the solutions of these problems with a vector of ones and zeros, which is the typical output of such networks.

All the combinatorial optimization problems considered in this chapter have solutions that can be represented by a fixed length vector of integers. Let the vector which represents such a solution be denoted by $\mathbf{p}$ and be of length n. Further, let its elements be integers in the range $\{1, \ldots, m\}$. Thus,

$$p_i \in \{1, \ldots, m\} \quad \text{where} \quad i \in \{1, \ldots, n\} \tag{1.2}$$

As a specific example, consider the Travelling Salesman problem. For a n-city problem the final solution must represent a possible tour which visits each city once and only once.

[1] It is also possible to formulate more general graph matching problems in which the graphs are of different sizes.

This can be done by letting p_i be the position of city i in the tour. For example, in a 4-city TSP, if the final tour order is (city 4, city 1, city 3, city 2, city 4), then the vector representing this would be $\mathbf{p}^T = [2, 4, 3, 1]$.

For a Graph Partitioning problem with n nodes and m partitions, p_i represents the partition to which node i belongs to. Note that in the case of Travelling Salesman problems there are two extra constraints on $\mathbf{p}$. These are that $n = m$ and that the elements of $\mathbf{p}$ must be unique, since each city can be visited only once.

The next task is to find a way of representing $\mathbf{p}$ by a vector $\mathbf{v}$ which is the output of the network and which is a vector of ones and zeros i.e. $\mathbf{v}$ lies at a hypercube corner in the network output space. This can be achieved as follows:

Let $\boldsymbol{\delta}^p(n)$ be the n dimensional co-ordinate vector given by,

$$\begin{aligned} [\boldsymbol{\delta}^p(n)]_i &= \left\{ \begin{array}{lll} 1 & \text{if} & i = p \\ 0 & \text{if} & i \neq p \end{array} \right. \quad p \in \{1, \ldots, n\} \\ &= \delta_{pi} \end{aligned} \tag{1.3}$$

In effect $\boldsymbol{\delta}^p(n)$ is the p^{th} column of the $n \times n$ identity matrix.

N.B. The (n) part of the vector $\boldsymbol{\delta}^p(n)$ specifies the number of elements of this vector and the range of p. If this is obvious from the context of use, then it will be omitted, leaving just $\boldsymbol{\delta}^p$.

Let $\mathbf{v}(\mathbf{p})$ be the nm dimensional vector representing the form of the final network output vector, $\mathbf{v}$, which corresponds to the problem solution denoted by $\mathbf{p}$. This is defined as follows:

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \boldsymbol{\delta}^{p_1}(m) \\ \boldsymbol{\delta}^{p_2}(m) \\ \vdots \\ \boldsymbol{\delta}^{p_n}(m) \end{bmatrix} \tag{1.4}$$

An alternative way of representing $\mathbf{p}$ is by a $n \times m$ matrix. Let such a matrix be denoted $\mathbf{V}(\mathbf{p})$, and be defined by:

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} \boldsymbol{\delta}^{p_1}(m)^T \\ \boldsymbol{\delta}^{p_2}(m)^T \\ \vdots \\ \boldsymbol{\delta}^{p_n}(m)^T \end{bmatrix} \tag{1.5}$$

or in component form

$$[\mathbf{V}(\mathbf{p})]_{ij} = [\boldsymbol{\delta}^{p_i}(m)]_j \tag{1.6}$$

For the TSP tour given above, the vectors $\mathbf{p}$ and $\mathbf{v}(\mathbf{p})$, and the matrix $\mathbf{V}(\mathbf{p})$ would expand as follows,

$$
\begin{aligned}
\mathbf{p}^T &= [2,4,3,1] \\[1em]
\mathbf{v}(\mathbf{p})^T &= \begin{array}{llll} [0\,1\,0\,0 & 0\,0\,0\,1 & 0\,0\,1\,0 & 1\,0\,0\,0] \\ [\boldsymbol{\delta}^{p_1 T} & \boldsymbol{\delta}^{p_2 T} & \boldsymbol{\delta}^{p_3 T} & \boldsymbol{\delta}^{p_4 T}] \end{array} \\[1em]
\mathbf{V}(\mathbf{p}) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

1.3 Mean Field Annealing (MFA) and Hopfield networks

Mean Field Annealing and Hopfield networks represent the two main classes of neural networks that can be used to solve combinatorial optimization problems [2]. As their name suggests, MFA networks are derived from the application of Mean Field theory to a combination of Spin-Glass models in Statistical Mechanics and Simulated Annealing [29]. Although the Hopfield network was original proposed as a neural network with interesting neuro-physiological and analog hardware parallels [23, 24, 25], it can also be derived from the same Statistical Mechanics theory. In order to present a unified theoretical approach, a common set of notation will be used to describe both networks, even if this is at variance with that used by the original authors.

The network descriptions given below are based on a combination of the works of D. Van den Bout [45, 46], C. Peterson [42, 43], P. Simic [44] and J.J.Hopfield [25].

1.3.1 Derivation of Mean Field Annealing networks

The start point for the derivation of this network is the Ising Spin-Glass model, and its corresponding Hamiltonian. The Ising model is an array of N coupled ferro-magnetic spins, where the i^{th} spin is denoted s_i and can take on a value of 1 (up) or 0 (down). The i^{th} and j^{th} spins are coupled by a spin interaction constant $\frac{1}{2}T_{ij}$. In addition, an external field i_i^b is applied to the i^{th} spin.

Let $\mathbf{s}$, $\mathbf{i}^b$, $\mathbf{T}$, and T^p respectively denote the vector of spins, the vector of external fields, the matrix of spin interaction constants, and the temperature.

The Hamiltonian of this model can be expressed as follows:

$$H(\mathbf{s}) = -\tfrac{1}{2}\mathbf{s}^T\mathbf{T}\mathbf{s} - (\mathbf{i}^b)^T\mathbf{s} \tag{1.7}$$

[2] Other highly effective neural network type techniques have been proposed for solving the Euclidean TSP. For example, R.Durbin & D,Wilshaw's Elastic Network [15] , and the networks of B.Angéniol et al [8] and B.Fritzke et al [17], both of which are based on Kohonen's Self Organising Feature Maps [30]. However, these networks are not considered in this thesis because they are highly specific to the Euclidean TSP.

and hence, from the Boltzmann distribution, the probability that the model is in configuration $\mathbf{s}$ is given by:

$$p(\mathbf{s}) = \frac{\exp\left(-\frac{H(\mathbf{s})}{T^p}\right)}{Z} \tag{1.8}$$

where Z is the partition function:

$$Z = \sum_{\{\mathbf{s}\}} \exp\left(-\frac{H(\mathbf{s})}{T^p}\right)$$

and where the summation runs over all 2^N possible spin configurations.

The quantity that is of interest is the expected value of the Hamiltonian:

$$\langle H(\mathbf{s})\rangle = \sum_{\{\mathbf{s}\}} H(\mathbf{s})p(\mathbf{s})$$

Clearly from (1.8) as the temperature decreases, those spin configurations with a low value of $H(\mathbf{s})$ will dominate, and hence as $T^p \rightarrow 0$,

$$\langle H(\mathbf{s})\rangle \rightarrow H(\mathbf{s}_{\text{min}})$$

where $\mathbf{s}_{\text{min}}$ is the spin configuration which produces the minimum value of $H(\mathbf{s})$.

Alternatively, using the Mean Field Approximation, $\langle H(\mathbf{s})\rangle$ can be written

$$\begin{aligned}\langle H(\mathbf{s})\rangle &= -\sum_i\sum_j \tfrac{1}{2}T_{ij}\langle s_i s_j\rangle - \sum_i i_i^b\langle s_i\rangle \\ &\approx -\sum_i\sum_j \tfrac{1}{2}T_{ij}\langle s_i\rangle\langle s_j\rangle - \sum_i i_i^b\langle s_i\rangle \end{aligned} \tag{1.9}$$

The problem now is to find an expression for $\langle s_i\rangle$. Note that,

$$\begin{aligned} T^p\nabla_{\mathbf{i}^b}\log(Z) &= T^p\frac{\nabla_{\mathbf{i}^b}Z}{Z} \\ &= T^p\frac{\sum_{\{\mathbf{s}\}}\nabla_{\mathbf{i}^b}\left(-\frac{H(\mathbf{s})}{T^p}\right)\exp\left(-\frac{H(\mathbf{s})}{T^p}\right)}{Z} \\ &= \sum_{\{\mathbf{s}\}}\mathbf{s}\frac{\exp\left(-\frac{H(\mathbf{s})}{T^p}\right)}{Z} \\ &= \sum_{\{\mathbf{s}\}}\mathbf{s}p(\mathbf{s}) \\ \Rightarrow T^p\nabla_{\mathbf{i}^b}\log(Z) &= \langle\mathbf{s}\rangle \end{aligned} \tag{1.10}$$

where $\nabla_{\mathbf{i}^b} \equiv \left[\frac{\partial}{\partial i_1^b}, \frac{\partial}{\partial i_2^b}, \cdots, \frac{\partial}{\partial i_N^b}\right]$

Derivation of $\nabla_{\mathbf{i}^b}\mathbf{log}(\mathbf{Z})$

The first step is to derive a more tractable expression for the partition function,

$$Z = \sum_{\{\mathbf{s}\}} \exp\left(-\frac{H(\mathbf{s})}{T^p}\right)$$

This is done by using the multi-dimensional delta function:

$$\int_{R^N} \mathrm{d}\mathbf{v}\, \delta(\mathbf{s}-\mathbf{v}) f(\mathbf{v}) = f(\mathbf{s})$$

Note that the multi-dimensional delta function can expressed as a complex exponential,

$$\delta(\mathbf{s}-\mathbf{v}) = C \int_{I^N} \mathrm{d}\mathbf{u} \;\; \exp(\mathbf{u}^T(\mathbf{s}-\mathbf{v}))$$

where C is a constant and I^N is a N-dimensional imaginary space.

Putting these together allows Z to be written as the summation of an integral, at the expense of introducing $2N$ new variables, $\mathbf{u}$ and $\mathbf{v}$.

$$\begin{aligned} Z &= C \sum_{\{\mathbf{s}\}} \int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u} \;\; \exp(\mathbf{u}^T(\mathbf{s}-\mathbf{v})) \exp\left(-\frac{H(\mathbf{v})}{T^p}\right) \\ \Rightarrow Z &= C \int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u} \;\; \exp\left(-\frac{H(\mathbf{v})}{T^p} - \mathbf{u}^T\mathbf{v}\right) \sum_{\{\mathbf{s}\}} \exp\left(\sum_i u_i s_i\right) \qquad (1.11) \end{aligned}$$

The next step is to deal with the summation. If this is over all 2^N spin configurations, then the following result can be used:

$$\begin{aligned} \sum_{\{\mathbf{s}\}} \exp\left(\sum_i u_i s_i\right) &= \prod_i (\exp(u_i)+1) \\ \Rightarrow \sum_{\{\mathbf{s}\}} \exp\left(\sum_i u_i s_i\right) &= \exp\left(\sum_i \log(\exp(u_i)+1)\right) \qquad (1.12) \end{aligned}$$

Let
$$g(\mathbf{u},\mathbf{v},T^p) = \exp\left(-\frac{H(\mathbf{v})}{T^p} - \mathbf{u}^T\mathbf{v} + \sum_i \log(\exp(u_i)+1)\right) \qquad (1.13)$$

Hence
$$\begin{aligned} Z &= C\int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u}\; g\left(\mathbf{u},\mathbf{v},T^p\right) && (1.14)\\ \Rightarrow \nabla_{\mathbf{i}^b}\log(Z) &= \frac{\nabla_{\mathbf{i}^b} Z}{Z} \\ &= \frac{C\int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u}\; \nabla_{\mathbf{i}^b}\left(-\frac{H(\mathbf{v})}{T^p}\right) g\left(\mathbf{u},\mathbf{v},T^p\right)}{Z} \\ \Rightarrow \nabla_{\mathbf{i}^b}\log(Z) &= \frac{1}{T^p}\frac{\int_{R^N} \mathrm{d}\mathbf{v}(\mathbf{v}) \int_{I^N} \mathrm{d}\mathbf{u}\; g\left(\mathbf{u},\mathbf{v},T^p\right)}{\int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u}\; g\left(\mathbf{u},\mathbf{v},T^p\right)} && (1.15)\end{aligned}$$

The saddle point approximation can now be made. Assuming that $g\left(\mathbf{u},\mathbf{v},T^p\right)$ is a function which is dominated by a single delta function like maximum at ($\tilde{\mathbf{u}}$,$\tilde{\mathbf{v}}$), it can be seen that the integral (1.15) will tend to pick out the value of ($\mathbf{v}$) at this maximum, i.e,

$$\langle \mathbf{s} \rangle = T^p \nabla_{\mathbf{i}^b}\log(Z) = \frac{\int_{R^N} \mathrm{d}\mathbf{v}(\mathbf{v}) \int_{I^N} \mathrm{d}\mathbf{u}\; g\left(\mathbf{u},\mathbf{v},T^p\right)}{\int_{R^N} \mathrm{d}\mathbf{v} \int_{I^N} \mathrm{d}\mathbf{u}\; g\left(\mathbf{u},\mathbf{v},T^p\right)} \approx \tilde{\mathbf{v}} \qquad (1.16)$$

At $\mathbf{u} = \tilde{\mathbf{u}}$,$\mathbf{v} = \tilde{\mathbf{v}}$ the function $g\left(\mathbf{u},\mathbf{v},T^p\right)$ will be at a saddle point, and hence the following saddle point equations will be satisfied,

SP1
$$\frac{\partial g}{\partial u_i} = 0 \qquad (1.17)$$

SP2
$$\frac{\partial g}{\partial v_i} = 0 \qquad (1.18)$$

Evaluating the partial derivatives gives,

$$\begin{aligned} \frac{\partial g}{\partial u_i} &= \left(-v_i + \frac{\exp(u_i)}{\exp(u_i)+1}\right) g\left(\mathbf{u},\mathbf{v},T^p\right) \\ \frac{\partial g}{\partial v_i} &= \left(\frac{\sum_j T_{ij} v_j + i_i^b}{T^p} - u_i\right) g\left(\mathbf{u},\mathbf{v},T^p\right)\end{aligned}$$

Hence at the saddle point,

SP1 ⇒
$$\tilde{v_i} = \tfrac{1}{2} + \tfrac{1}{2}\tanh\left(\frac{\tilde{u_i}}{2}\right) \qquad (1.19)$$

SP2 ⇒
$$\tilde{u_i} = \frac{1}{T^p}\left(\sum_j T_{ij}\tilde{v_j} + i_i^b\right) \qquad (1.20)$$

Thus from (1.9)(1.10)(1.16) the expected value of the Hamiltonian at temperature T^p can be written:

$$\langle H(\mathbf{s}) \rangle = -\tfrac{1}{2}\tilde{\mathbf{v}}^T \mathbf{T} \tilde{\mathbf{v}} - (\mathbf{i}^b)^T \tilde{\mathbf{v}} \qquad (1.21)$$

where $\tilde{\mathbf{v}}$ is given from the solution of the saddle point equations (1.20) and (1.19).

1.3.2 Operation of MFA networks

The key to the operation of MFA networks is the computation of a solution to the saddle point equations SP1 and SP2. This is then used via (1.21) to estimate $\langle H(\mathbf{s})\rangle$ at a particular temperature T^p. If $T^p \to 0$ then $\langle H(\mathbf{s})\rangle \to H(\mathbf{s}_{\text{min}})$, and hence at low temperatures $\tilde{\mathbf{v}}$ gives a good estimate of the spin configuration $\mathbf{s}_{\text{min}}$ that minimizes $H(\mathbf{s})$. Unfortunately, although $g(\mathbf{u}, \mathbf{v}, T^p)$ may be dominated by a global maximum at $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$, at low temperatures it will have many other sub-maxima, and it is quite possible that a solution of the saddle point equations will simply give values for $\mathbf{u}$ and $\mathbf{v}$ that correspond to one of the sub-maxima. This is much less likely at high temperatures, since the smooth $\sum_i \log(\exp(u_i) + 1)$ term of $g(\mathbf{u}, \mathbf{v}, T^p)$ will dominate over the quadratic $-\frac{H(\mathbf{v})}{T^p}$ term. However, at high temperatures $\langle H(\mathbf{s})\rangle$ is an average over many good configurations, and hence $\tilde{\mathbf{v}} = \langle \mathbf{s}\rangle$ does not correspond to a single spin configuration, but the average over many. Therefore, in a method analogous to Simulated Annealing, the saddle point equations are first solved for high temperature estimates of $\langle H(\mathbf{s})\rangle$. The temperature is then lowered and the new solution is computed. As the temperature is gradually lowered $\tilde{\mathbf{v}}$ should also gradually evolve from an average of many configurations to the final desired value of $\mathbf{s}_{\text{min}}$, but in a way that avoids the sub-maxima.

The exact details of the implementation depend upon how the saddle point equations are solved for a particular temperature T^p. The simplest is the asynchronous discrete implementation.

The Asynchronous Discrete Implementation of MFA networks

This implementation is based on the following method of computing values for $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ that satisfy the saddle point equations SP1 and SP2 for a particular temperature T^p.

1. Initialise $\tilde{\mathbf{u}}$ to a random vector, then use SP1 to compute the initial values for $\tilde{\mathbf{v}}$

2. Pick an element of $\tilde{\mathbf{u}}$ at random and update it using SP2. Then use SP1 to calculate the new value for the corresponding element of $\tilde{\mathbf{v}}$

3. Repeat step 2 until there are no further changes in $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$.

Once a solution for a particular temperature has been found, the annealing procedure described above is used until the final low temperature solution is achieved. For details see [45, 46, 42, 43].

1.3.3 Improving MFA networks using Normalization

In the derivation of MFA networks used above, the summation of the spin configurations was over all 2^N possible configurations. In order to get a more accurate expression for the partition function, it is necessary to restrict the summation to cover only those configurations that are valid solutions of the combinatorial optimization problem being solved.

Let **S** be the matrix equivalent of **s** where $S_{xi} = s_{m(x-1)+i}$
Let **U** be the matrix equivalent of **u** where $U_{xi} = u_{m(x-1)+i}$

Unfortunately, it is not possible to restrict the summation to those configurations which are exactly of the form $\mathbf{s} = \mathbf{v}(\mathbf{p})$. However, it is possible to restrict the summation to those configurations where the matrix **S** has either a single element per row with a value of one, or it a single element per column with a value of one (the other elements being all zero) [3].

The key summation in the derivation of the partition function is

$$\sum_{\{\mathrm{s}\}} \exp\left(\mathbf{u}^T\mathbf{s}\right)$$

If the row condition is assumed, and the summation performed over all spin configurations for which only a single element per row of **S** has a value of one, then the following result applies:

$$\begin{aligned} \sum_{\{\mathrm{s}\}} \exp\left(\sum_x \sum_i U_{xi}S_{xi}\right) &= \prod_x \left(\sum_i \exp(U_{xi})\right) \\ \Rightarrow \sum_{\{\mathrm{s}\}} \exp\left(\sum_x \sum_i U_{xi}S_{xi}\right) &= \exp\left(\sum_x \log\left(\sum_i \exp(U_{xi})\right)\right) \end{aligned} \tag{1.22}$$

If this is substituted in the expression for $g(\mathbf{u}, \mathbf{v}, T^p)$ (1.13), then the saddle point equation SP1 becomes:

$$\tilde{V}_{xi} = \frac{\exp(\tilde{U}_{xi})}{\sum_i \exp(\tilde{U}_{xi})} \tag{1.23}$$

Apart from this modification, the operation of the MFA network remains identical to the descriptions given in section 1.3.2. Both D. van den Bout [45, 46] and C. Peterson [43] use this normalization modification in their implementations of MFA networks.

[3] A similar restriction on the summation is used in Statistical Mechanics for the derivation of Potts Glasses [27, 50].

The critical temperature

During the annealing process, it has been found [46, 43] that above a certain critical temperature T^{crit} the solution to the saddle point equations remains constant, with all the $\tilde{v}_i$'s having approximately the same value. At the critical temperature, the network exhibits a behaviour analogous to a phase transition, and the $\tilde{v}_i$'s start diverging in value. Since the network does very little if $T^p > T^{\text{crit}}$, the efficiency of the network can be greatly improved by starting the annealing process with $T^p = T^{\text{crit}}$. Both D. van den Bout [46] and C. Peterson have developed analytic expressions that give good estimates of T^{crit} for solving TSPs and GPPs. This phenomenon is discussed in greater detail in section 5.1.1.

1.3.4 The Hopfield Network

The Hopfield network, as introduced in [23, 25], is constructed by connecting a large number of simple processing elements to each other. In general, the i^{th} processing element, or neuron, is described by two variables: its current state u_i, and its output v_i. The output is related to the state by a simple non-decreasing monotonic output function:

$$v_i = g(u_i)$$

This function operates as a threshold function to limit the output of each neuron to the interval 0 to 1 (or sometimes -1 to $+1$), in order to ensure that the output of the network always lies on or within the unit hypercube.

The output of the i^{th} neuron is fed to the input of the j^{th} neuron by a connection of strength T_{ij}. Each neuron also has an offset bias of i_i^b applied at its input. The state of the i^{th} neuron, u_i, is updated by a function of the total input to the neuron. The exact form of $g(u_i)$ depends upon whether the continuous or discrete Hopfield network is being used. In addition the neurons may be updated individually at random (asynchronously) or all together (synchronously).

The Discrete Model

This is the simplest form of the Hopfield model and is identical to the threshold logic systems investigated by McCulloch and Pitts [40]. In this model $g(u_i)$ is a step function of the form,

$$g(u_i) = \begin{cases} 0 & u_i < 0.5 \\ 1 & u_i \geq 0.5 \end{cases}$$

and consequently v_i is a discrete variable with a value of 0 or 1. In addition u_i is updated in discrete time steps by replacement with a value given by a function of its total input. Thus the value of u_i at time step $n+1$ is calculated from $\mathbf{v}$ at time step n as follows:

$$u_i^{(n+1)} = \sum_j T_{ij} v_j^{(n)} + i_i^b$$

(a) Non-linear threshold functions constraining $\mathbf{v}$ to the unit hypercube

(b) The change in $\mathbf{u}$ is specified by the differential equation (1.26). However in a computer simulation a discrete approximation has to be used, where time is split up into a series of discrete time steps Δt and the change in $\mathbf{u}$ is given by,

$$\Delta\mathbf{u} = \Delta t(-\eta\mathbf{u} + \mathbf{Tv} + \mathbf{i}^b)$$

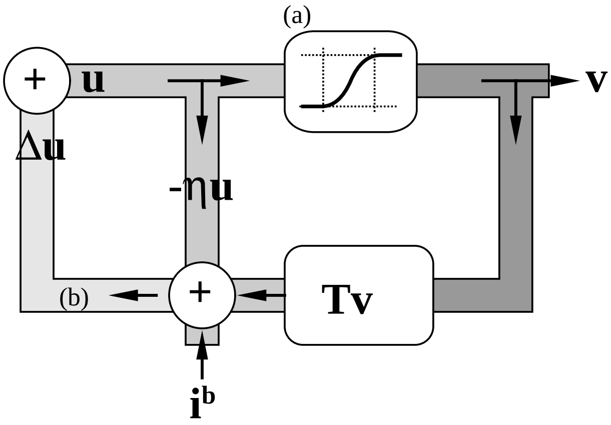


Figure 1.1: Schematic diagram of continuous Hopfield network

If the neurons are updated asynchronously and the connection matrix is symmetric, then this network has a Liapunov function of the form [23]:

$$E = -\tfrac{1}{2}\mathbf{v}^T\mathbf{Tv} - (\mathbf{i}^b)^T\mathbf{v} \tag{1.24}$$

It can be shown that if the elements of $\mathbf{u}$ are updated randomly according to the above rule, then E in (1.24) is always decreased [23].

The Continuous Model

A schematic diagram of the continuous Hopfield Network [25], and the operation of its dynamic equation are shown in Fig 1.1. In the continuous model, v_i is a continuous variable in the interval 0 to 1 and $g(u_i)$ is a continuous function, usually a hyperbolic tangent of the form,

$$g(u_i) = \tfrac{1}{2} + \tfrac{1}{2}\tanh(\beta u_i) \tag{1.25}$$

In this case, so that stability can be guaranteed, u_i is updated continuously by evaluating $\frac{d}{dt}(u_i)$ as a function of the total input to the i^{th} neuron.

$$\dot{\mathbf{u}} = -\eta\mathbf{u} + \mathbf{Tv} + \mathbf{i}^b \quad \text{where } v = g(u) \tag{1.26}$$

Hopfield proves stability for this network by showing that with this dynamic equation the network output $\mathbf{v}$ evolves so as to minimize the Liapunov function [24]:

$$E = -\frac{1}{2}\mathbf{v}^T\mathbf{Tv} - (\mathbf{i}^b)^T\mathbf{v} + \eta\sum_i \int_0^{v_i} g^{-1}(\alpha)d\alpha \tag{1.27}$$

With the continuous network it is possible to use asynchronous or synchronous update. Computer simulations using synchronous update are much more efficient, however, from a theoretical point of view it should make very little difference to the operation of the network as to which mode of update is used.

1.3.5 Equivalence between Hopfield Networks and MFA networks

The saddle point equation SP2 (1.19) for the asynchronous discrete implementation of MFA networks (with or without normalization) has the following form,

$$\tilde{u}_i^{(n+1)} = \frac{1}{T^p}\left(\sum_j T_{ij}\tilde{v}_j + i_i^b\right)$$

We can write this as a discrete approximation to a continuous differential equation,

$$\tilde{u}_i(t+\Delta t) = \tilde{u}_i(t) + \Delta\tilde{u}_i \tag{1.28}$$

where

$$\Delta\tilde{u}_i = \Delta t\left(-\tilde{u}_i + \frac{1}{T^p}\left(\sum_j T_{ij}\tilde{v}_j + i_i^b\right)\right) \tag{1.29}$$

and

$$\Delta t = 1$$

In other words, we can view SP2 as an algorithm for performing an approximate integration [4] of,

$$\frac{d}{dt}(\tilde{u}_i) = -\tilde{u}_i + \frac{1}{T^p}\left(\sum_j T_{ij}\tilde{v}_j + i_i^b\right) \tag{1.30}$$

Consider now the following equations:

$$\begin{aligned} v_i &= g\left(u_i\right) && (1.31)\\ \dot{\mathbf{u}} &= -\eta\mathbf{u} + \frac{1}{T^p}\left(\mathbf{T}\mathbf{v} + \mathbf{i}^b\right) && (1.32)\end{aligned}$$

By making the substitutions shown in Table 1.1, equations (1.31) and (1.32) can be converted to either the continuous Hopfield network dynamic equations or the MFA dynamic equation represented by (1.30). Thus (1.31) and (1.32) can be used as general equations for the dynamics of EM networks.

Clearly, if $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ satisfy SP2, then according to (1.30),

$$\frac{d}{dt}(\tilde{u}_i) = 0$$

Thus the stable states of a continuous Hopfield network with $\eta = 1$ correspond to the expected spin configurations of a Mean Field Annealing network at a temperature $T^p = \frac{1}{2\beta}$.

[4] Along the lines of the Euler method.

Type of Network	T^p	η	u_p	$g\,(u_q)$
Continuous Hopfield	$\frac{1}{2\beta}$	η	$\frac{u_q}{T^p}$	$\frac{1}{1+\exp(-u_q)}$
MFA without normalization	T^p	1	$\tilde{u}_q$	$\frac{1}{1+\exp(-u_q)}$
MFA with normalization	T^p	1	$\tilde{u}_q$	$\frac{\exp(U_{xi})}{\sum_i \exp(U_{xi})}$ $U_{xi} \equiv u_q$ $q{=}m(x{-}1){+}i$

Table 1.1: Substitutions for T^p, η and $g()$ required to convert equations (1.31) and (1.32) to the dynamic equations for the continuous Hopfield and discrete asynchronous MFA networks

Since the continuous Hopfield network functions by performing a descent minimization of the Liapunov function (1.27),

$$E = E^{op} + E^{cn} + \eta \sum \int_0^v g^{-1}(\alpha) d\alpha$$

we can also view the underlying operation of the asynchronous discrete MFA update algorithm as performing a descent minimization of the above Liapunov function.

1.4 Mapping Combinatorial Optimization Problems

1.4.1 Enforcing Valid Solutions and Optimizing the Solution Cost

The EM networks considered in this thesis all operate by minimizing an energy function, or Hamiltonian, which is a quadratic function of the network output vector $\mathbf{v}$. For consistency the following quadratic function will be used to denote both the energy of the Hopfield network, and the expected value of Hamiltonian for MFA networks.

$$E = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}\mathbf{v} - (\mathbf{i}^b)^T\mathbf{v}$$

Unfortunately, the set of valid forms of $\mathbf{v}$ (i.e where $\mathbf{v} = \mathbf{v}(\mathbf{p})$) is much smaller than the set of possible hypercube corners: for example, in the TSP, the former has $n!$ members, while the latter has 2^n members. Hence, the key to successfully mapping a particular problem onto an EM network is to find a way of setting $\mathbf{T}$ and $\mathbf{i}^b$ such that in minimizing E, the network both converges to a valid solution $\mathbf{v}(\mathbf{p})$, and finds a solution that is of optimum cost.

Let E be split up into a confinement (cn) and optimization (op) term, such that in minimizing E^{cn} the network ensures $\mathbf{v}$ is a valid form, and in minimizing E^{op} the network moves $\mathbf{v}$ towards an optimum cost solution. This can be achieved by splitting $\mathbf{T}$ and $\mathbf{i}^b$ so that,

$$\begin{aligned} \mathbf{T} &= \mathbf{T}^{cn} + \mathbf{T}^{op} \\ \mathbf{i}^b &= \mathbf{i}^{cn} + \mathbf{i}^{op} \end{aligned} \tag{1.33}$$

Thus E splits into,

$$\begin{aligned} E &= E^{cn} + E^{op} \\ \text{where } E^{cn} &= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{cn}\mathbf{v} - (\mathbf{i}^{cn})^T\mathbf{v} \qquad (1.34) \\ \text{and } E^{op} &= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{op}\mathbf{v} - (\mathbf{i}^{op})^T\mathbf{v} \qquad (1.35) \end{aligned}$$

1.4.2 Expressions for E^{cn}, $\mathbf{T}^{cn}$ and $\mathbf{i}^{cn}$

This section lists a variety of expressions for E^{cn}, T^{cn} and $\mathbf{i}^{cn}$ for the TSP and GPP. These have been taken from the works of Hopfield and Tank [25], S.Abe [1], C.Peterson [43] and D. van den Bout [46], the aim being to show both the key ideas of the mappings, and also that there are several equivalent expressions for them.

In all the expressions, the elements of the nm-element output vector $\mathbf{v}$ are indexed by its $n \times m$ matrix equivalent $\mathbf{V}$, where $\mathbf{V}$ bears the same relation to $\mathbf{v}$ as $\mathbf{V}(\mathbf{p})$ does to $\mathbf{v}(\mathbf{p})$. i.e.

$$\mathbf{v} = \text{vec}(\mathbf{V}^T) \quad \text{and} \quad V_{xi} \equiv v_{m(x-1)+i} \quad \text{where } x, i \in \{1, \ldots, m\}$$

Following Hopfield and Tank's notation [25] , the $nm \times nm$ matrix $\mathbf{T}^{cn}$ and nm-element vector $\mathbf{i}^{cn}$ will be indexed using double indices, defined as follows,

$$\begin{aligned} T_{xi,yj} &\equiv T_{pq} \\ i^b_{xi} &\equiv i^b_p \end{aligned} \qquad \text{where} \qquad \begin{aligned} x, y &\in \{1, \ldots, n\} \\ i, j &\in \{1, \ldots, m\} \\ p &= m(x-1) + i \\ q &= m(y-1) + j \end{aligned}$$

N.B. For the TSP $n = m =$ the number of cities, and for the GPP $n =$ the number of nodes, $m =$ the number of partitions.

The expressions are based on forming a quadratic expression of $\mathbf{v}$ which, over the set of 2^N hypercube corners, is minimized for those corners of the form $\mathbf{v}(\mathbf{p})$. These quadratic expressions can be further subdivided into row and column constraint terms.

The Row Constraint Terms

These are minimized when one element of each row of $\mathbf{V}$ has a value of 1 and the rest have a value of 0. There are two common expressions.

$$\begin{aligned} E^{r1} &= \tfrac{1}{2}\sum_x \left(\sum_i V_{xi} - 1\right)^2 \\ E^{r2} &= \tfrac{1}{2}\sum_x \sum_i \sum_{j\neq i} V_{xi}V_{xj} \end{aligned}$$

Note that E^{r1} is minimized when there is exactly one element per row of $\mathbf{V}$ with a value of 1, while E^{r2} is minimized when there is either one element or no elements per row of $\mathbf{V}$ with a value of 1.

The Column Constraint Terms

These are minimized when 1 element of each column of $\mathbf{V}$ has a value of 1 and the rest have a value of 0. There are also two common expressions.

$$\begin{aligned} E^{c1} &= \tfrac{1}{2}\sum_i \left(\sum_x V_{xi} - 1\right)^2 \\ E^{c2} &= \tfrac{1}{2}\sum_i \sum_x \sum_{y\neq x} V_{xi}V_{yi} \end{aligned}$$

As with the row case, E^{c1} is minimized when there is exactly one element per column of $\mathbf{V}$ with a value of 1, while E^{c2} is minimized when there is either one element or no elements per column of $\mathbf{V}$ with a value of 1.

In the case of the Travelling Salesman problem, both the row and column constraints need to be satisfied if $\mathbf{V} = \mathbf{V}(\mathbf{p})$. Table 1.2 shows some expressions for E^{cn} that have been used by various authors, and Table 1.3 shows the corresponding values for $\mathbf{T}^{cn}$ and $\mathbf{i}^{cn}$.

Author	E^{cn}
S.Abe [1]	$A(E^{r1}) + B(E^{c1})$
C.Peterson [43]	$\beta(E^{r2}) + \alpha(E^{c1})$
J.Hopfield & D.Tank [25]	$A(E^{r2}) + B(E^{c2}) + C\tfrac{1}{2}\left(\sum_x \sum_i V_{xi} - N\right)^2$

Table 1.2: Various E^{cn} expressions for the TSP

Note that in the case of Hopfield and Tank's expression, there is an extra C term. This is required because the A and B terms are minimized not only if there is just a single neuron on per row/column, but also if there are no neurons on. A trivial case is $\mathbf{v}=\mathbf{0}$, in which case both the A and B terms are 0. To compensate for this effect, the C term is used to ensure that the network settles down with exactly N neurons on.

Author	$T^{cn}_{xi,yj}$	i^{cn}_{xi}
S.Abe [1]	$-A\delta_{xy} - B\delta_{ij}$	A+B
C.Peterson [43]	$-\beta\delta_{xy}(1-\delta_{ij}) - \alpha\delta_{ij}$	α
J.Hopfield & D.Tank [25]	$-A\delta_{xy}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{xy}) - C$	Cn

Table 1.3: Values for $\mathbf{T}^{cn}$ and $\mathbf{i}^{cn}$ which give the E^{cn} expressions in Table 1.2

In the case of the GPP, the column constraint has to be slightly modified because (as is usually the case) it is required that the size of each partition is the same. If there are n nodes and m partitions, this means exactly n/m elements of each column of $\mathbf{V}$ have a value of 1, whilst the rest are zero. This can be achieved by changing E^{c1} to,

$$E^{c1} = \tfrac{1}{2}\sum_i \left(\sum_x V_{xi} - \frac{n}{m}\right)^2$$

The E^{c2} term cannot be modified to accomplish this, so it is not possible to use Hopfield and Tank's E^{cn}. S. Abe's expression is not concerned with the GPP, which leaves C. Peterson's expression. This gives the following values for $\mathbf{T}^{cn}$ and $\mathbf{i}^{cn}$:

$$T^{cn}_{xi,yj} = -\beta\delta_{xy}(1-\delta_{ij}) - \alpha\delta_{ij} \qquad (1.36)$$

$$i^{cn}_{xi} = \frac{n}{m}\alpha \qquad (1.37)$$

1.4.3 Expressions for E^{op}, $\mathbf{T}^{op}$ and $\mathbf{i}^{op}$

The purpose of the E^{op} term is to measure the cost of a valid solution $\mathbf{v}(\mathbf{p})$, so that in minimizing $E = E^{cn} + E^{op}$ the network will find the optimum cost solution. For the TSP, this means E^{op} should measure the total length of the tour specified by $\mathbf{p}$, and for the GPP, E^{op} should measure the sum of the weights of all the edges that are cut by the partitions specified by $\mathbf{p}$.

Let d_{xy} be the distance between city x and city y for the TSP.
Let e_{xy} be the weight between node x and node y for the GPP.

Table 1.4 gives the expressions for E^{op}, $\mathbf{T}^{op}$ and $\mathbf{i}^{op}$ that result in E^{op} measuring the appropriate cost.

Problem	E^{op}	$T^{op}_{xi,yj}$	i^{op}_{xi}
TSP[25]	$\frac{1}{2}\sum_{x}\sum_{y\neq x}\sum_{i} d_{xy}V_{xi}(V_{y,i\oplus 1}+V_{y,i\ominus 1})$	$-d_{xy}(\delta_{j,i\oplus 1}+\delta_{j,i\ominus 1})$	0
	$a\oplus b = a+b,\ a\ominus b = a-b$ except for $n\oplus 1 = 1$ and $1\ominus 1 = n$		
GPP[43]	$\frac{1}{2}\sum_{x}\sum_{y}\sum_{i}\sum_{i\neq j} e_{xy}V_{xi}V_{yj}$	$-e_{xy}(1-\delta_{ij})$	0

Table 1.4: Expressions for E^{op}, $\mathbf{T}^{op}$ and $\mathbf{i}^{op}$

$$
\begin{aligned}
\frac{1}{\eta} &\equiv RC \\
T_{ij} &\equiv \frac{1}{R_{ij}} \\
v_i &= \text{Output Voltage of i}^{\text{th}}\text{ OpAmp} \\
u_i &= \text{Input Voltage of i}^{\text{th}}\text{ OpAmp}
\end{aligned}
$$

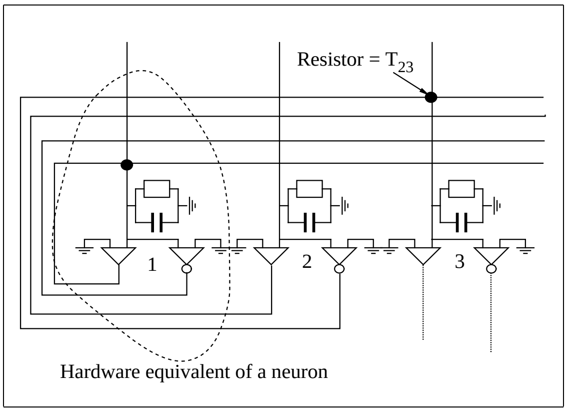


Figure 1.2: Analog Hardware Implementation of Hopfield Network

1.5 Hardware Implementation

One of the key reasons for the popularity of the continuous Hopfield network is the possibility for implementing it on analog hardware. When they first proposed the continuous network in [25], Hopfield and Tank also demonstrated that it had a very simple analog hardware equivalent shown in Fig 1.2. Recently, far more sophisticated hardware implementations have been proposed, using a variety of different approaches, but all based on the fact that the dynamics of the Hopfield network require a matrix-vector multiplication followed by a sigmoidal non-linearity [39, 16, 41].

Unfortunately, the same easy mapping onto analog hardware is not available for the MFA networks with normalization, because of the presence of the complicated denominator in SP1. Although this is not a problem if a fast parallel digital architecture is being used, it does rule out the potential speed gains from fully analog hardware.

1.6 Results achieved with EM networks

The first published results on solving combinatorial optimization problems with EM networks are given in Hopfield and Tank's 1985 paper [25]. In this paper they performed computer simulations of the continuous Hopfield network being used to solve 10 and 30 city Euclidean TSPs. The connection matrix and input bias were set according to the expressions given in Tables 1.3 and 1.4. They report moderate success with the 10 city problem (the optimum solution being achieved 30% of the time). For the 30 city problem the results were quite bad, with the best solution being 20% off the optimum, and for the majority of the time the network converging to solutions which were invalid (i.e not of the form $\mathbf{v}(\mathbf{p})$). The problem of convergence to invalid solutions has also been reported by B. Kamgar-Parsi [26] and V. Wilson & G. S. Pawley [48]. The general perception is that although the Hopfield network has the potential to solve the TSP, in practice it rarely does so. One of the main reasons cited for this is the difficulty of finding the correct values for the arbitrary constants A,B,C and D which control the relative weighting of the constraint and optimization terms, E^{cn} and E^{op} [22]. It has also been proposed [43] that the constraint and optimization terms frustrate each other, either preventing valid solutions, or preventing solutions of good cost.

Much greater success has been reported for the MFA networks which employ the normalization modification. D.van den Bout has reported very encouraging results for the GPP [46], with the MFA network out-performing both Simulated Annealing and the powerful Kernighan-Lin heuristic [28]. Similar success for both the TSP and GPP has been reported by C. Peterson [43], who has achieved very good results for TSPs of up to 200 cities.

Without the normalization modification, the results for the MFA networks are much the same as the continuous Hopfield network. Since the effect of normalization is to rigidly enforce the row constraint, this seems to point to either a problem with the constraint term, or a frustration between the constraint and optimization terms, as the root cause of the poor performance of both the Hopfield network and MFA networks without normalization. Even for the MFA network with normalization, the formulation of the constraint term can be problematic, since to avoid convergence to invalid solutions, the parameters of the network have to be set very carefully [43]. Thus, if EM networks are to be used reliably for solving combinatorial optimization problems, a robust way has to be found of overcoming the problem of convergence to invalid solutions, and it is this issue that is addressed in the next chapter.

Chapter 2

Enforcing validity: The Subspace Approach

It was reported in the last chapter that many researchers have had great difficulty in ensuring that EM networks achieve valid solutions to combinatorial optimization problems [1]. In this chapter it is shown that the reason for this is that the constraint and optimization terms in (1.34) and (1.35) interfer with each other in a destructive way. With the aid of a new mathematical framework, a novel way of avoiding this problem is then described, which is based on decoupling the effects of these terms into separate vector subspaces: hence the 'subspace approach'. Using this approach, a new expression for the constraint terms is developed which guarantees convergence to valid solutions. Finally, the implications that this has for the minimization of the optimization term are analyzed.

2.1 A New Mathematical Framework

Before describing the subspace approach it is first necessary to introduce a mathematical framework. This is based on the use of Kronecker products [21] (also known as tensor products) and a set of matrix and vector identities. With these, it is possible to simplify the notation commonly used to describe EM networks, greatly aiding in the subsequent analysis of them.

Let $[\mathbf{P}]_{ij}$ denote the element in the i^{th} row and j^{th} column of matrix $\mathbf{P}$. Similarly let $[\mathbf{p}]_i$ denote the i^{th} element of vector $\mathbf{p}$. Where there is no danger of ambiguity the alternative P_{ij} and p_i notation will be used.

[1] This statement mainly applies to the Hopfield network and MFA network without the normalization modification, since the issue of convergence to invalid solutions is much less of a problem for MFA networks with normalization.

Let $\mathbf{U}_{.q}$ be a m-element vector corresponding to the q^{th} column of the $m \times n$ matrix $\mathbf{U}$. i.e,

$$\mathbf{U}_{.q} = \begin{bmatrix} U_{1q} \\ U_{2q} \\ \cdots \\ U_{mq} \end{bmatrix} \tag{2.1}$$

From this definition it follows that,

$$[\mathbf{A}^T\mathbf{V}\mathbf{B}]_{kl} = (\mathbf{A}_{.k})^T\mathbf{V}\mathbf{B}_{.l} \tag{2.2}$$

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

$$\begin{aligned} \mathbf{v} = \text{vec}(\mathbf{U}) &= \begin{bmatrix} \mathbf{U}_{.1} \\ \mathbf{U}_{.2} \\ \cdots \\ \mathbf{U}_{.n} \end{bmatrix} \\ &= [U_{11}, U_{21} \ldots U_{m1}, U_{12}, U_{22} \ldots U_{m2}, \cdots\cdots, U_{1n}, U_{2n} \ldots U_{mn}]^T \end{aligned} \tag{2.3}$$

Note also that if $\mathbf{v}$ is a valid solution, then

$$\mathbf{v} = \mathbf{v}(\mathbf{p}) = \text{vec}(\mathbf{V}(\mathbf{p})^T) \tag{2.4}$$

Let $(\mathbf{P} \otimes \mathbf{Q})$ denote the Kronecker product of two matrices.
Thus if $\mathbf{P}$ is an $n \times n$ matrix, and $\mathbf{Q}$ is an $m \times m$ matrix, then $(\mathbf{P} \otimes \mathbf{Q})$ is a $nm \times nm$ matrix given by,

$$(\mathbf{P} \otimes \mathbf{Q}) = \begin{bmatrix} P_{11}\mathbf{Q} & P_{12}\mathbf{Q} & \cdots & P_{1n}\mathbf{Q} \\ P_{21}\mathbf{Q} & P_{22}\mathbf{Q} & \cdots & P_{2n}\mathbf{Q} \\ \cdots & \cdots & \cdots & \cdots \\ P_{n1}\mathbf{Q} & P_{n2}\mathbf{Q} & \cdots & P_{nn}\mathbf{Q} \end{bmatrix} \tag{2.5}$$

Equation (2.5) can be written in component form as,

$$[(\mathbf{P} \otimes \mathbf{Q})]_{pq} = P_{xy}Q_{ij} \qquad \text{where} \qquad \begin{aligned} p &= m(x-1)+i \\ q &= m(y-1)+j \\ x, y &\in \{1, \ldots, n\} \\ i, j &\in \{1, \ldots, m\} \end{aligned} \tag{2.6}$$

In Hopfield and Tank's double index notation [25], $[(\mathbf{P} \otimes \mathbf{Q})]_{pq}$ would be written as $[(\mathbf{P} \otimes \mathbf{Q})]_{xi,yj}$.

Similarly if $\mathbf{w}$ is a n-element vector and $\mathbf{h}$ a m-element vector, then $(\mathbf{w} \otimes \mathbf{h})$ is a nm-element vector given by,

$$(\mathbf{w} \otimes \mathbf{h}) = \begin{bmatrix} w_1\mathbf{h} \\ w_2\mathbf{h} \\ \cdots \\ w_n\mathbf{h} \end{bmatrix} \tag{2.7}$$

The following properties of the vec() operator and Kronecker products are utilised later (for the proofs refer to [21]).

$$\begin{aligned}
(\lambda\mathbf{w}\otimes\gamma\mathbf{h}) &= \lambda\gamma(\mathbf{w}\otimes\mathbf{h}) && (2.8)\\
(\mathbf{w}\otimes\mathbf{h})^T(\mathbf{x}\otimes\mathbf{g}) &= (\mathbf{w}^T\mathbf{x})(\mathbf{h}^T\mathbf{g}) && (2.9)\\
(\mathbf{P}\otimes\mathbf{Q})(\mathbf{w}\otimes\mathbf{h}) &= (\mathbf{P}\mathbf{w}\otimes\mathbf{Q}\mathbf{h}) && (2.10)\\
(\mathbf{P}^T\otimes\mathbf{Q}^T) &= (\mathbf{P}\otimes\mathbf{Q})^T && (2.11)\\
(\mathbf{P}\otimes\mathbf{Q})(\mathbf{E}\otimes\mathbf{F}) &= (\mathbf{PE}\otimes\mathbf{QF}) && (2.12)\\
\mathrm{trace}(\mathbf{V}^T\mathbf{E}) &= \mathrm{vec}(\mathbf{V})^T\mathrm{vec}(\mathbf{E}) && (2.13)\\
\mathrm{vec}(\mathbf{QVP}^T) &= (\mathbf{P}\otimes\mathbf{Q})\mathrm{vec}(\mathbf{V}) && (2.14)\\
\mathrm{trace}(\mathbf{V}^T\mathbf{QVP}^T) &= \mathrm{vec}(\mathbf{V})^T(\mathbf{P}\otimes\mathbf{Q})\mathrm{vec}(\mathbf{V}) && (2.15)
\end{aligned}$$

Let $\mathbf{I}^n$ and $\mathbf{I}^{nm}$ be respectively the $n \times n$ and $nm \times nm$ identity matrices, i.e,

$$\begin{aligned}
I^n_{ij} &= \delta_{ij} \quad i,j \in \{1,\ldots,n\}\\
\text{and } \mathbf{I}^{nm} &= (\mathbf{I}^n\otimes\mathbf{I}^m) && (2.16)
\end{aligned}$$

Let $\mathbf{o}^n$ and $\mathbf{o}^{nm}$ be respectively the n element and nm element vectors of ones, i.e,

$$\begin{aligned}
o^n_p &= 1 \quad p \in \{1,\ldots,n\}\\
\text{and } \mathbf{o}^{nm} &= (\mathbf{o}^n\otimes\mathbf{o}^m) && (2.17)
\end{aligned}$$

Let $\mathbf{O}^n$ and $\mathbf{O}^{nm}$ be respectively the $n \times n$ and $nm \times nm$ matrices of ones, i.e,

$$\begin{aligned}
O^n_{ij} &= 1 \quad i,j \in \{1,\ldots,n\}\\
\text{and } \mathbf{O}^{nm} &= (\mathbf{O}^n\otimes\mathbf{O}^m) && (2.18)
\end{aligned}$$

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

$$\mathbf{R}^n = \mathbf{I}^n - \tfrac{1}{n}\mathbf{o}^n\mathbf{o}^{nT} = \mathbf{I}^n - \tfrac{1}{n}\mathbf{O}^n \tag{2.19}$$

or in component form,

$$R^n_{ij} = \delta_{ij} - \tfrac{1}{n} \tag{2.20}$$

If $\mathbf{x}$ is an arbitrary n dimensional vector, then

$$\mathbf{R}^n\mathbf{x} = \begin{cases} \mathbf{x} & \text{if } \mathbf{o}^{nT}\mathbf{x} = 0\\ \mathbf{0} & \text{if } \mathbf{x} = k\mathbf{o}^n\\ \mathbf{x} - \tfrac{1}{n}\mathbf{O}^n\mathbf{x} & \text{otherwise} \end{cases} \tag{2.21}$$

Also,

$$\begin{aligned}
\mathbf{R}^n\mathbf{R}^n &= \mathbf{I}^n\mathbf{I}^n - \tfrac{1}{n}\mathbf{I}^n\mathbf{O}^n\\
&\quad -\tfrac{1}{n}\mathbf{O}^n\mathbf{I}^n + \tfrac{1}{n^2}\mathbf{O}^n\mathbf{O}^n\\
&= \mathbf{I}^n - \tfrac{1}{n}\mathbf{O}^n\\
\Rightarrow \mathbf{R}^n\mathbf{R}^n &= \mathbf{R}^n && (2.22)
\end{aligned}$$

Thus $\mathbf{R}^n$ is a projection matrix which removes the $\mathbf{o}^n$ component from $\mathbf{x}$, so that if $\mathbf{x}$ is orthogonal to $\mathbf{o}^n$ it is left unchanged. Note that if $\mathbf{y} = \mathbf{R}^n\mathbf{x}$, then for any $\mathbf{x}$ the sum of the elements of $\mathbf{y}$ is always zero.

$$\begin{aligned}
\sum_{k=1}^{n} y_k &= \sum_{k=1}^{n} [\mathbf{x} - \tfrac{1}{n}\mathbf{O}^n\mathbf{x}]_k \\
&= \mathbf{o}^{nT}[\mathbf{x} - \tfrac{1}{n}\mathbf{O}^n\mathbf{x}] \\
&= \mathbf{o}^{nT}\mathbf{x} - \tfrac{1}{n}(\mathbf{o}^{nT}\mathbf{o}^n\mathbf{o}^{nT})\mathbf{x} \\
&= \mathbf{o}^{nT}\mathbf{x} - \mathbf{o}^{nT}\mathbf{x} \\
\Rightarrow \sum_{k=1}^{n} y_k &= \sum_{k} [\mathbf{R}^n\mathbf{x}]_k = 0 \qquad (2.23)
\end{aligned}$$

In terms of eigenvalues this corresponds to $\mathbf{R}^n$ having a degenerate eigenvalue of 1 for all vectors orthogonal to $\mathbf{o}^n$, and an eigenvalue of 0 for $\mathbf{o}^n$.

2.2 The Valid Subspace

In this section it is shown that there exists a particular subspace, termed the **valid subspace** [2] , such that if a hypercube corner lies within it, then it must be a valid solution hypercube corner of the form $\mathbf{v}(\mathbf{p})$. The exact form of this subspace depends upon the type of combinatorial problem being solved. As examples, explict expressions of the valid subspace will be derived for the TSP and GPP.

2.2.1 A simple 3-dimensional example

(a) the plane or 2D valid subspace where, $\sum_{k=1}^{3} v_k = 1$, which contains the cube corners

$$\{(1,0,0),\ (0,1,0),\ (0,0,1)\}$$

(b) the plane or 2D valid subspace where, $\sum_{k=1}^{3} v_k = 2$, which contains the cube corners

$$\{(1,1,0),\ (1,0,1),\ (0,1,1)\}$$

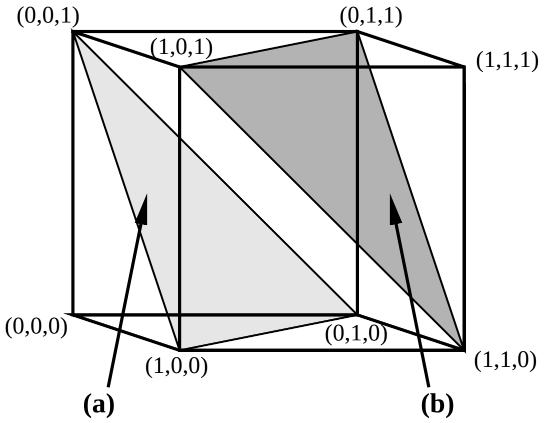


Figure 2.1: A Simplified 3D illustration of the valid subspace

[2] Strictly speaking this should be the valid affine subspace or valid linear manifold, since as later analysis reveals, it does not pass through the origin.

Fig 2.1 shows a simplified illustration of the key principle of this analysis: that a subspace can be constructed which contains only those hypercube corners which correspond to a valid solution. Consider two possible forms of valid solution,

the set, $$\{(1,0,0),\ (0,1,0),\ (0,0,1)\} \qquad (2.24)$$

or the set, $$\{(1,1,0),\ (1,0,1),\ (0,1,1)\} \qquad (2.25)$$

The lighter shaded plane is in effect a 2-dimensional valid subspace which only contains corners from the first set, while the darker shaded plane only contains corners from the second set.

Let,

$$S = \sum_{k=1}^{3} v_k$$

Another way of viewing the lighter shaded plane is that it is the plane where $S = 1$, while the darker shaded plane is where $S = 2$. Clearly, if $S = 1$ and $\mathbf{v}$ is a cube corner then $\mathbf{v}$ must belong to the first set of valid solutions, likewise if $S = 2$, then $\mathbf{v}$ must belong to the second set.

Recalling the definitions of $\mathbf{R}^n$ and $\mathbf{o}^n$ (see equations (2.19) and (2.17)), let $\mathbf{v}$ satisfy the equation,

$$\mathbf{v} = \tfrac{1}{3}\mathbf{o}^3 + \mathbf{R}^3\mathbf{v} \qquad (2.26)$$

Since from (2.23),

$$\begin{aligned} \sum_{k=1}^{3}[\mathbf{R}^3\mathbf{v}]_k &= 0 \\ \text{and} \quad \sum_{k=1}^{3}[\tfrac{1}{3}\mathbf{o}^3]_k &= 1 \end{aligned}$$

it can be seen that if $\mathbf{v}$ satisfies (2.26) then it also satisfies $S = 1$. Therefore (2.26) is the equation of the valid subspace for the first set of valid solution cube corners. Essentially the $\mathbf{R}^3\mathbf{v}$ term of equation (2.26) is just a projection of $\mathbf{v}$ onto the subspace orthogonal to $\mathbf{o}^3$, i.e the plane where $S = 0$, and it is the $\frac{1}{3}\mathbf{o}^3$ term of (2.26) that controls the value of S. Thus changing this term to $\frac{2}{3}\mathbf{o}^3$ ensures that $S = 2$, and hence

$$\mathbf{v} = \tfrac{2}{3}\mathbf{o}^3 + \mathbf{R}^3\mathbf{v} \qquad (2.27)$$

is the equation of the valid subspace for the second set of valid solution cube corners.

2.2.2 Generalization to hypercube corners of the form v(p)

The following analysis is based upon the properties of the $n \times m$ matrix $\mathbf{V}$ which is related to the nm element network output vector $\mathbf{v}$ by the expression:

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

The properties of interest are the row and column sums of **V**, which are defined as follows:

Let **c** be the m element column sum vector of **V**.

$$\begin{aligned} c_i &= \sum_{x=1}^{n} V_{xi} \\ \Rightarrow \mathbf{c}^T &= \mathbf{o}^{nT}\mathbf{V} \end{aligned} \tag{2.28}$$

Let **r** be the n element row sum vector of **V**.

$$\begin{aligned} r_x &= \sum_{i=1}^{m} V_{xi} \\ \Rightarrow \mathbf{r} &= \mathbf{V}\mathbf{o}^m \end{aligned} \tag{2.29}$$

If **v** is a valid solution **v**(**p**), then the row and column sums of **V** are given by those of **V**(**p**), which has the following properties for TSPs and GPPs:

TSPs

In this case $n = m$ and **V**(**p**) is a permutation matrix where a single element per row or column of **V**(**p**) has a value of 1, while the rest have values of 0. This corresponds to **V** having row and column sums of 1. Hence for this class of problems, given that $\mathbf{v} = \text{vec}(\mathbf{V}^T)$ is a hypercube corner, the condition

$$\mathbf{c} = \mathbf{r} = \mathbf{o}^n \tag{2.30}$$

is sufficient to guarantee that **v** is of the form **v**(**p**).

GPPs

If n is the number of nodes, and m the number of partitions, then $p_x = i$ if node x is in partition i. Since $[\mathbf{V}(\mathbf{p})]_{xi} = 1$ iff $p_x = i$ and node x can only belong to one partition, it can be seen that only one element per row of **V**(**p**) has a value of 1 (the rest have values of 0). Thus

$$\begin{aligned} r_x &= 1 \\ \Rightarrow \mathbf{r} &= \mathbf{o}^n \end{aligned} \tag{2.31}$$

The column sums, **c**, now represent the number of nodes in each partition. For certain GPPs it is desirable to have partitions of equal size. In this case c_i will be given by

$$\begin{aligned} c_i &= \tfrac{n}{m} \\ \Rightarrow \mathbf{c} &= \tfrac{n}{m}\mathbf{o}^m \end{aligned} \tag{2.32}$$

It is however possible to set c_i to any value as long as the sum of the column sums, $\sum_{i=1}^{m} c_i$, is equal to n.

The Zerosum Subspace

In order to define the valid subspace it is first necessary to define the **zerosum subspace** (zs). This has the property that if $\mathbf{v}$ lies in the zerosum subspace, then $\mathbf{V}$, where $\mathbf{v} = \mathrm{vec}(\mathbf{V}^T)$, has row and column sums equal to zero.

Let $\mathbf{v}^{\mathrm{zs}}$ be the vector given by,

$$\mathbf{v}^{\mathrm{zs}} = \mathbf{T}^{\mathrm{zs}}\mathbf{v} \tag{2.33}$$

where $\mathbf{T}^{\mathrm{zs}}$ is given by,

$$\mathbf{T}^{\mathrm{zs}} = (\mathbf{R}^n \otimes \mathbf{R}^m) \tag{2.34}$$

It will now be shown that $\mathbf{T}^{\mathrm{zs}}$ is a projection matrix which projects $\mathbf{v}$ onto the zerosum subspace, giving $\mathbf{v}^{\mathrm{zs}}$ i.e. the component of $\mathbf{v}$ in the zerosum subspace; this will be done by evaluating the row and column sums of $\mathbf{V}^{\mathrm{zs}}$, where

$$\mathbf{v}^{\mathrm{zs}} = \mathrm{vec}(\mathbf{V}^{\mathrm{zs}T}) \tag{2.35}$$

Using the identity (2.14) and the fact that $\mathbf{R}^n = \mathbf{R}^{nT}$, allows (2.35) to be transformed as follows:

$$\begin{aligned} \mathbf{T}^{\mathrm{zs}}\mathbf{v} &= \mathrm{vec}(\mathbf{V}^{\mathrm{zs}T}) \\ \Rightarrow (\mathbf{R}^n \otimes \mathbf{R}^m)\mathrm{vec}(\mathbf{V}^T) &= \mathrm{vec}(\mathbf{V}^{\mathrm{zs}T}) \\ \Rightarrow \mathrm{vec}(\mathbf{R}^m\mathbf{V}^T\mathbf{R}^n) &= \mathrm{vec}(\mathbf{V}^{\mathrm{zs}T}) \\ \Rightarrow \mathbf{R}^n\mathbf{V}\mathbf{R}^m &= \mathbf{V}^{\mathrm{zs}} \end{aligned} \tag{2.36}$$

Let $\mathbf{c}^{\mathrm{zs}}$ and $\mathbf{r}^{\mathrm{zs}}$ be the vectors of column and row sums of $\mathbf{V}^{\mathrm{zs}}$. From (2.28), (2.29) and $\mathbf{R}^n\mathbf{o}^n = \mathbf{0}$ (2.21) it can be seen that the row and column sums of $\mathbf{V}^{\mathrm{zs}}$ are zero:

$$\begin{array}{llll} \mathbf{c}^{\mathrm{zs}T} & = \mathbf{o}^n\mathbf{V}^{\mathrm{zs}} & = (\mathbf{o}^{nT}\mathbf{R}^n)\mathbf{V}\mathbf{R}^m & = \mathbf{0} \\ \mathbf{r}^{\mathrm{zs}} & = \mathbf{V}^{\mathrm{zs}}\mathbf{o}^n & = \mathbf{R}^n\mathbf{V}\mathbf{R}^m\mathbf{o}^m & = \mathbf{0} \end{array}$$

General expression for the Valid Subspace

Consider the equation

$$\begin{aligned} \mathbf{V} &= \mathbf{R}^n\mathbf{V}\mathbf{R}^m + \mathbf{o}^n\mathbf{a}^T \\ \Rightarrow \mathbf{V} &= \mathbf{V}^{\mathrm{zs}} + \mathbf{o}^n\mathbf{a}^T \end{aligned} \tag{2.37}$$

If $\mathbf{V}$ satisfies this equation, then its row and column sums are given by those of the matrix $\mathbf{o}^n\mathbf{a}^T$, since $\mathbf{V}^{\mathrm{zs}}$ has row and column sums of zero.

Hence if (2.37) is satisfied,

$$\begin{array}{rll} \mathbf{c}^T & = \mathbf{o}^{nT}(\mathbf{o}^n\mathbf{a}^T) & = n\mathbf{a}^T \\ \mathbf{r} & = (\mathbf{o}^n\mathbf{a}^T)\mathbf{o}^m & = \mathbf{o}^n(\mathbf{a}^T\mathbf{o}^m) \end{array} \tag{2.38}$$

Problem	Requirements	a	s	Valid Subspace Equation
TSP	$\mathbf{c} = \mathbf{r} = \mathbf{o}^n$	$\frac{1}{n}\mathbf{o}^n$	$\frac{1}{n}(\mathbf{o}^n \otimes \mathbf{o}^n)$	$\mathbf{v} = (\mathbf{R}^n \otimes \mathbf{R}^n)\mathbf{v} + \frac{1}{n}(\mathbf{o}^n \otimes \mathbf{o}^n)$
GPP	$\mathbf{r} = \mathbf{o}^n$ $\mathbf{c} = \frac{n}{m}\mathbf{o}^m$	$\frac{1}{m}\mathbf{o}^m$	$\frac{1}{m}(\mathbf{o}^n \otimes \mathbf{o}^m)$	$\mathbf{v} = (\mathbf{R}^n \otimes \mathbf{R}^m)\mathbf{v} + \frac{1}{m}(\mathbf{o}^n \otimes \mathbf{o}^m)$

Table 2.1: Expressions for the Valid Subspace

The matrix equation (2.37) can now be turned into a vector equation by first transposing it, then applying the vec() operator followed by the identity (2.14),

$$\begin{aligned} \mathbf{V}^T &= \mathbf{R}^m\mathbf{V}^T\mathbf{R}^n + \mathbf{a}\mathbf{o}^{nT} \\ \Rightarrow \mathrm{vec}(\mathbf{V}^T) &= \mathrm{vec}(\mathbf{R}^m\mathbf{V}^T\mathbf{R}^n) + \mathrm{vec}(\mathbf{a}.1.\mathbf{o}^{nT}) \\ \Rightarrow \mathrm{vec}(\mathbf{V}^T) &= (\mathbf{R}^n \otimes \mathbf{R}^m)\mathrm{vec}(\mathbf{V}^T) + (\mathbf{o}^n \otimes \mathbf{a})\mathrm{vec}(1) \end{aligned}$$

By substituting $\mathbf{v}$ for vec($\mathbf{V}^T$), the general equation for the valid subspace is obtained

$$\boxed{\mathbf{v} = \mathbf{T}^{\mathrm{zs}}\mathbf{v} + \mathbf{s}} \tag{2.39}$$

where $\mathbf{s} = (\mathbf{o}^n \otimes \mathbf{a})$
$\mathbf{T}^{\mathrm{zs}} = (\mathbf{R}^n \otimes \mathbf{R}^m)$
$\mathbf{a}$ is an arbitrary m-element vector

The space spanned by all vectors, $\mathbf{v}$, that satisfy (2.39) is the definition of the valid subspace.

The key property of this subspace is that the row and column sums of $\mathbf{V}$ are purely a function of $\mathbf{s}$. Therefore, depending on the particular combinatorial optimization problem, the value of $\mathbf{s}$ can be set to ensure that the subspace contains only the valid solution hypercube corners. For the TSP and GPP, Table 2.1 gives the appropriate expressions for $\mathbf{s}$, $\mathbf{a}$ and the resulting valid subspace equations.

Note that $\mathbf{s}$ is orthogonal to the zerosum subspace, since

$$\mathbf{T}^{\mathrm{zs}}\mathbf{s} = (\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{o}^n \otimes \mathbf{a}) = (\mathbf{R}^n\mathbf{o}^n \otimes \mathbf{R}^m\mathbf{a}) = \mathbf{0} \tag{2.40}$$

2.3 Frustration between constraint and optimization

It is now possible to analyse the constraint and optimization terms given in Tables 1.2 and 1.3, to see whether they correctly confine $\mathbf{v}$ to the valid subspace, or if they frustrate each other. In order to aid the analysis, it is first necessary to rewrite the expressions given in Tables 1.2 and 1.3. This is done both to simplify the expressions, and to make explict their relationship to the valid subspace.

2.3.1 New Expressions for $\mathbf{T}^{\mathrm{cn}}, \mathbf{i}^{\mathrm{cn}}$ and E^{cn}

First note that in double index notation (see equation (2.6)),

$$\begin{aligned}
[\mathbf{I}^n \otimes \mathbf{I}^m]_{xi,yj} &= \delta_{xy}\delta_{ij} \\
[\mathbf{I}^n \otimes \mathbf{O}^m]_{xi,yj} &= \delta_{xy} \\
[\mathbf{O}^n \otimes \mathbf{I}^m]_{xi,yj} &= \delta_{ij} \\
[\mathbf{O}^n \otimes \mathbf{O}^m]_{xi,yj} &= 1
\end{aligned}$$

Using these, the expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$ in Table (1.3) can be rewritten in Kronecker product form; the results are show in Table (2.2).

Author	$\mathbf{T}^{\mathrm{cn}}$	$\mathbf{i}^{\mathrm{cn}}$
S.Abe	$-A(\mathbf{I}^n \otimes \mathbf{O}^n) - B(\mathbf{O}^n \otimes \mathbf{I}^n)$	$(A+B)(\mathbf{o}^n \otimes \mathbf{o}^n)$
C.Peterson	$-\beta(\mathbf{I}^n \otimes \mathbf{O}^n) - \alpha(\mathbf{O}^n \otimes \mathbf{I}^n) + (\alpha+\beta)(\mathbf{I}^n \otimes \mathbf{I}^n)$	$\alpha(\mathbf{o}^n \otimes \mathbf{o}^n)$
J.Hopfield	$-A(\mathbf{I}^n \otimes \mathbf{O}^n) - B(\mathbf{O}^n \otimes \mathbf{I}^n)$ $+(A+B)(\mathbf{I}^n \otimes \mathbf{I}^n) - C(\mathbf{O}^n \otimes \mathbf{O}^n)$	$C(\mathbf{o}^n \otimes \mathbf{o}^n)$

Table 2.2: Kronecker Product Expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$ for the TSP

Let the subspace orthogonal to the zerosum subspace be termed the **nonzero subspace** (nz). Any nm-dimensional vector can be expressed as the sum of its components in each of these subspaces, i.e,

$$\mathbf{v} = \mathbf{v}^{\mathrm{nz}} + \mathbf{v}^{\mathrm{zs}} \quad \text{with} \quad \mathbf{v}^{\mathrm{nz}T}\mathbf{v}^{\mathrm{zs}} = 0$$

Let the nonzero subspace be further decomposed into the space spanned by all vectors orthogonal to $\mathbf{o}^{nm}$, termed the nonzero-one (nzo) subspace, and a component along $\mathbf{o}^{nm}$ (o). Thus any vector in the nonzero subspace can be decomposed as follows,

$$\mathbf{v}^{\mathrm{nz}} = \mathbf{v}^{\mathrm{nzo}} + \mathbf{v}^{\mathrm{o}} \quad \text{with} \quad \mathbf{v}^{\mathrm{nzo}T}\mathbf{v}^{\mathrm{o}} = 0$$

Thus for $\mathbf{v}$ satisfying the valid subspace equations given in Table 2.1,

$$\begin{aligned} \mathbf{v}^{\mathrm{zs}} &= \begin{cases} (\mathbf{R}^n \otimes \mathbf{R}^n)\mathbf{v} & \text{for the TSP} \\ (\mathbf{R}^n \otimes \mathbf{R}^m)\mathbf{v} & \text{for the GPP} \end{cases} \\ \mathbf{v}^{\mathrm{nzo}} &= 0 \\ \mathbf{v}^{\mathrm{o}} &= \begin{cases} \frac{1}{n}\mathbf{o}^{nn} & \text{for the TSP} \\ \frac{1}{m}\mathbf{o}^{nm} & \text{for the GPP} \end{cases} \end{aligned} \tag{2.41}$$

Let $\mathbf{T}^{\mathrm{nzo}}$ be the matrix which projects $\mathbf{v}$ onto the nonzero-one subspace.
Let $\mathbf{T}^{\mathrm{o}}$ be the matrix which projects $\mathbf{v}$ onto $\mathbf{o}^{nm}$.

Since,

$$\mathbf{v} = \mathbf{v}^{\mathrm{zs}} + \mathbf{v}^{\mathrm{nzo}} + \mathbf{v}^{\mathrm{o}}$$

if follows that,

$$\mathbf{I}^{nm} = \mathbf{T}^{\mathrm{zs}} + \mathbf{T}^{\mathrm{nzo}} + \mathbf{T}^{\mathrm{o}} \tag{2.42}$$

From the definitions of $\mathbf{T}^{\mathrm{zs}}$ and $\mathbf{T}^{\mathrm{o}}$ it can be seen that,

$$\begin{aligned} \mathbf{T}^{\mathrm{zs}} &= (\mathbf{R}^n \otimes \mathbf{R}^m) \\ &= \left((\mathbf{I}^n - \frac{1}{n}\mathbf{O}^n) \otimes (\mathbf{I}^m - \frac{1}{m}\mathbf{O}^m)\right) \\ \Rightarrow \mathbf{T}^{\mathrm{zs}} &= \mathbf{I}^{nm} - \frac{1}{m}(\mathbf{I}^n \otimes \mathbf{O}^m) - \frac{1}{n}(\mathbf{O}^n \otimes \mathbf{I}^m) + \frac{1}{nm}\mathbf{O}^{nm} \\ \mathbf{T}^{\mathrm{o}} &= \frac{1}{nm}\mathbf{O}^{nm} \end{aligned} \tag{2.43}$$

Hence from (2.42) and (2.43),

$$\mathbf{T}^{\mathrm{nzo}} = \frac{1}{m}(\mathbf{I}^n \otimes \mathbf{O}^m) + \frac{1}{n}(\mathbf{O}^n \otimes \mathbf{I}^m) - 2\mathbf{T}^{\mathrm{o}} \tag{2.44}$$

Using (2.42), (2.44) and (2.43), it is possible to rewrite Table 2.2 in terms of $\mathbf{T}^{\mathrm{zs}}$, $\mathbf{T}^{\mathrm{nzo}}$ and $\mathbf{T}^{\mathrm{o}}$. For simplicity, it is assumed (as is almost always the case) that $A = B$ and $\alpha = \beta$ (i.e the row and column constraints have equal weight). Also, since Table 2.2 applies to the TSP, $n = m$. The results are shown in Table 2.3

The E^{cn} expressions in Table 1.2 can be rewritten in terms of the magnitude of the zerosum, nonzero-one and $\mathbf{o}^{nn}$ components of $\mathbf{v}$, as is shown in Table 2.4. (For details of the derivation of Table 2.4 from Table 1.2, see Appendix A.1).

With the expressions in Table 2.4, it is easy to see the conditions on the zerosum, nonzero-one and $\mathbf{o}^{nn}$ components of $\mathbf{v}$, necessary to minimize E^{cn}. These are shown in Table 2.5.

Author	$\mathbf{T}^{\mathrm{cn}}$	$\mathbf{i}^{\mathrm{cn}}$
S.Abe	$-An\mathbf{T}^{\mathrm{nzo}} - 2An\mathbf{T}^{\mathrm{o}}$	$2A\mathbf{o}^{nn}$
C.Peterson	$-\alpha(n-2)\mathbf{T}^{\mathrm{nzo}} - 2\alpha(n-1)\mathbf{T}^{\mathrm{o}} + 2\alpha\mathbf{T}^{\mathrm{zs}}$	$\alpha\mathbf{o}^{nn}$
J.Hopfield	$-A(n-2)\mathbf{T}^{\mathrm{nzo}} - (2A(n-1)+Cn^2)\mathbf{T}^{\mathrm{o}} + 2A\mathbf{T}^{\mathrm{zs}}$	$C\mathbf{o}^{nn}$

Table 2.3: Subspace Expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$ for the TSP

Author	E^{cn}
S.Abe	$\frac{1}{2}An\left\|\mathbf{v}^{\mathrm{nzo}}\right\|^2 + An\left\|\mathbf{v}^{\mathrm{o}} - \frac{1}{n}\mathbf{o}^{nn}\right\|^2 - \frac{1}{2}$
C.Peterson	$\frac{\alpha}{2}(n-2)\left\|\mathbf{v}^{\mathrm{nzo}}\right\|^2 - \alpha\left\|\mathbf{v}^{\mathrm{zs}}\right\|^2 + \alpha(n-1)\left\|\mathbf{v}^{\mathrm{o}} - \frac{\alpha}{2\alpha(n-1)}\mathbf{o}^{nn}\right\|^2 - \mathrm{cnst}$
J.Hopfield	$\frac{A(n-2)}{2}\left\|\mathbf{v}^{\mathrm{nzo}}\right\|^2 - A\left\|\mathbf{v}^{\mathrm{zs}}\right\|^2 + \frac{2A(n-1)+Cn^2}{2}\left\|\mathbf{v}^{\mathrm{o}} - \frac{C}{2A(n-1)+Cn^2}\mathbf{o}^{nn}\right\|^2 - \mathrm{cnst}$

Table 2.4: E^{cn} for the TSP expressed in terms of $\mathbf{v}^{\mathrm{zs}}$, $\mathbf{v}^{\mathrm{nzo}}$ and $\mathbf{v}^{\mathrm{o}}$

2.3.2 Why minimizing E^{cn} causes invalid and poor cost solutions for the TSP

It is now possible to see the cause of the invalid and poor cost solutions to the TSP that many researchers have reported [48, 26]. Essentially, it is that the minimization of E^{cn}

Author	E^{cn} is minimized when		
S.Abe		$\mathbf{v}^{\mathrm{nzo}} \to 0$	$\mathbf{v}^{\mathrm{o}} \to \frac{1}{n}\mathbf{o}^{nn}$
C.Peterson	$\left\|\mathbf{v}^{\mathrm{zs}}\right\| \to \infty$	$\mathbf{v}^{\mathrm{nzo}} \to 0$	$\mathbf{v}^{\mathrm{o}} \to \frac{\alpha}{2\alpha(n-1)}\mathbf{o}^{nn}$
J.Hopfield	$\left\|\mathbf{v}^{\mathrm{zs}}\right\| \to \infty$	$\mathbf{v}^{\mathrm{nzo}} \to 0$	$\mathbf{v}^{\mathrm{o}} \to \frac{C}{2A(n-1)+Cn^2}\mathbf{o}^{nn}$

Table 2.5: Minimization of E^{cn} (TSP) expressed as conditions on $\mathbf{v}^{\mathrm{zs}}$, $\mathbf{v}^{\mathrm{nzo}}$ and $\mathbf{v}^{\mathrm{o}}$

does not correctly confine $\mathbf{v}$ to the valid subspace. It also interfers with the minimization of E^{op}. For $\mathbf{v}$ to satisfy the valid subspace equation for the TSP (see Table 2.1) the conditions in (2.41) need to be satisfied. From Table 2.5 it can be seen that of the three versions of E^{cn}, it is only S.Abe's [1] that satsifies all the conditions in (2.41). There are two key problems with C.Peterson's and J.J.Hopfield's formulation:

(i) In order to minimize E^{cn} the $\mathbf{v}^{\mathrm{o}}$ component is set to a value other than $\frac{1}{n}\mathbf{o}^{nn}$. Hence $\mathbf{v}$ is not correctly confined to the valid subspace, and as it approaches a hypercube corner it will not correspond to a valid solution.

(ii) E^{cn} can be reduced simply by increasing the magnitude of the $\mathbf{v}^{\mathrm{zs}}$ component. This behaviour is the cause of the poor cost solutions. For all valid solutions $\mathbf{v}^{\mathrm{nzo}} = 0$ and $\mathbf{v}^{\mathrm{o}} = \frac{1}{n}\mathbf{o}^{nn}$, hence it is the $\mathbf{v}^{\mathrm{zs}}$ component of $\mathbf{v}$ that selects between different valid solutions. Ideally, the value of this component should only be controlled by the minimization of E^{op}. If, as is the case in C.Peterson's and J.J.Hopfield's formulations, the minimization E^{cn} also affects the $\mathbf{v}^{\mathrm{zs}}$ component, then a degradation of the final solution quality will occur.

The root cause of problems (i) and (ii) can be traced to the use by C.Peterson and J.Hopfield of the E^{r2} and E^{c2} row and column constraint terms, instead of the E^{r1} and E^{c1} terms (see section 1.4.2). Unfortunately, although S.Abe formulates E^{cn} correctly, by just using the E^{r1} and E^{c1} terms, in his actual implementation [1] he sets the leading diagonal of $\mathbf{T}$ to zero and uses the wrong setting for $\mathbf{i}^{b}$. Thus what emerges is that although superficially it seems there are several alternative formulations for E^{cn}, $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$, a deeper analysis reveals that there is only one which correctly enforces convergence to the valid subspace. It is the mistaken use of the others that is the main cause of invalid solutions.

2.4 Ensuring Confinement to the Valid Subspace

In order to ensure both confinement to the valid subspace and non-interference with the optimization terms I propose [3] the following expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$.

$$\begin{aligned} \mathbf{T}^{\mathrm{cn}} &= \theta(\mathbf{T}^{\mathrm{zs}} - \mathbf{I}^{nm}) && (2.45) \\ \mathbf{i}^{\mathrm{cn}} &= \theta\mathbf{s} && (2.46) \\ \text{where } \theta &\text{ is an arbitrary positive constant} \end{aligned}$$

Substituting the decomposition $\mathbf{v} = \mathbf{v}^{\mathrm{zs}} + \mathbf{v}^{\mathrm{nz}}$ in (1.34), rewriting $\mathbf{T}^{\mathrm{cn}}$ with (2.45) and $\mathbf{i}^{\mathrm{cn}}$ with (2.46), leads to,

$$E^{\mathrm{cn}} = -\tfrac{1}{2}\theta(\mathbf{v}^{\mathrm{nz}} + \mathbf{v}^{\mathrm{zs}})^{T}(\mathbf{T}^{\mathrm{zs}} - \mathbf{I}^{nm})(\mathbf{v}^{\mathrm{nz}} + \mathbf{v}^{\mathrm{zs}})$$

[3] It would also to be possible to use S.Abe's [1] expressions for the TSP. However, the expressions I describe are more directly related to the form of the valid subspace, and hence can be used for a wider variety of problems.

$$-\theta \mathbf{s}^T(\mathbf{v}^{\rm nz} + \mathbf{v}^{\rm zs})$$

But $\mathbf{T}^{\rm zs}$ is a projection matrix (by definition symmetric),

$$\begin{aligned} \text{hence } \mathbf{T}^{\rm zs}\mathbf{v}^{\rm zs} &= \mathbf{T}^{\rm zs}\mathbf{T}^{\rm zs}\mathbf{v} = \mathbf{v}^{\rm zs} \\ \text{and } \mathbf{v}^{{\rm zs}T}\mathbf{T}^{\rm zs} &= \mathbf{v}^T\mathbf{T}^{\rm zs}\mathbf{T}^{\rm zs} = \mathbf{v}^{{\rm zs}T} \end{aligned}$$

Also,

$$\mathbf{v}^{{\rm nz}T}\mathbf{T}^{\rm zs} = \mathbf{0}^T \text{ and } \mathbf{T}^{\rm zs}\mathbf{v}^{\rm nz} = \mathbf{0}$$

Thus,

$$\begin{aligned} E^{\rm cn} &= -\tfrac{1}{2}\theta[\mathbf{v}^{{\rm zs}T}\mathbf{v}^{\rm zs} - \mathbf{v}^{{\rm zs}T}\mathbf{v}^{\rm zs} - \mathbf{v}^{{\rm nz}T}\mathbf{v}^{\rm nz}] \\ &\quad -\theta\mathbf{s}^T\mathbf{v}^{\rm nz} \\ &= \tfrac{1}{2}\theta[\mathbf{v}^{{\rm nz}T}\mathbf{v}^{\rm nz} - 2\mathbf{s}^T\mathbf{v}^{\rm nz}] \\ \Rightarrow E^{\rm cn} &= \tfrac{1}{2}\theta|\mathbf{v}^{\rm nz} - \mathbf{s}|^2 - \tfrac{1}{2}\theta|\mathbf{s}|^2 \qquad (2.47) \end{aligned}$$

Therefore, since $\mathbf{s}$ is fixed, $E^{\rm cn}$ is minimized when

$$\mathbf{v}^{\rm nz} = \mathbf{s}$$

which corresponds to $\mathbf{v}$ satisfying the valid subspace equation,

$$\mathbf{v} = \mathbf{s} + \mathbf{v}^{\rm zs} = \mathbf{s} + \mathbf{T}^{\rm zs}\mathbf{v}$$

EM networks function by minimizing the sum of $E^{\rm cn}$ and $E^{\rm op}$:

$$\begin{aligned} E &= E^{\rm cn} + E^{\rm op} \qquad (2.48) \\ &= \tfrac{1}{2}\theta|\mathbf{v}^{\rm nz} - \mathbf{s}|^2 - \tfrac{1}{2}\theta|\mathbf{s}|^2 + E^{\rm op} \qquad (2.49) \end{aligned}$$

By setting θ to a large enough value, we can make the $\theta|\mathbf{v}^{\rm nz} - \mathbf{s}|^2$ term dominate E. Thus to minimize E the network will confine $\mathbf{v}^{\rm nz}$ so that,

$$|\mathbf{v}^{\rm nz} - \mathbf{s}| < \epsilon$$

where ϵ can be made arbitrarily small by making θ arbitrarily large. Hence, with these new expressions for $\mathbf{T}^{\rm cn}$ and $\mathbf{i}^{\rm cn}$, we can ensure confinement to the valid subspace within an arbitrary tolerance ϵ.

2.5 Implications for the minimization of $E^{\rm op}$

2.5.1 New Expressions for $\mathbf{T}^{\rm op}$, $\mathbf{i}^{\rm op}$ and $E^{\rm op}$

It is convenient for the later analysis to recast the expressions for $\mathbf{T}^{\rm op}$, $\mathbf{i}^{\rm op}$ and $E^{\rm op}$, given in Table 1.4, using the new mathematical framework.

Let,

$$\begin{aligned} \mathbf{T}^{\mathrm{op}} &= (\mathbf{P} \otimes \mathbf{Q}) \\ \mathbf{i}^{\mathrm{op}} &= \mathbf{0} \end{aligned} \tag{2.50}$$

We can see form the definition of Kronecker products given in (2.6) that,

$$[\mathbf{T}^{\mathrm{op}}]_{xi,yj} = P_{xy} Q_{ij}$$

hence using Table 1.4, the values of P_{xy} and Q_{ij} can be set so that $\mathbf{T}^{\mathrm{op}}$ measures the appropriate cost. Table 2.6, shows the resulting expressions.

Problem	$T^{\mathrm{op}}_{xi,yj} = [(\mathbf{P} \otimes \mathbf{Q})]_{xi,yj}$	P_{xy}	Q_{ij}
TSP[25]	$-d_{xy}(\delta_{j,i\oplus 1} + \delta_{j,i\ominus 1})$	$-d_{xy}$	$\delta_{j,i\oplus 1} + \delta_{j,i\ominus 1}$
GPP[43]	$-e_{xy}(1 - \delta_{ij})$	$-e_{xy}$	$1 - \delta_{ij}$
	$a \oplus b = a + b$ except for $n \oplus 1 = 1$ $a \ominus b = a - b$ except for $1 \ominus 1 = n$ d_{xy} = distance between city x and y e_{xy} = weight of edge between node i and j		

Table 2.6: Kronecker Product Expressions for $\mathbf{T}^{\mathrm{op}}$

2.5.2 Confinement to the Valid Subspace and the minimization of E^{op}

We can now account for the implications that confinement to the valid subspace has on the minization of E^{op}

Using the expansion,

$$\begin{aligned} E^{\mathrm{op}} &= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{op}}\mathbf{v} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v} \\ &= -\tfrac{1}{2}(\mathbf{v}^{\mathrm{zs}} + \mathbf{v}^{\mathrm{nz}})^T\mathbf{T}^{\mathrm{op}}(\mathbf{v}^{\mathrm{zs}} + \mathbf{v}^{\mathrm{nz}}) - (\mathbf{i}^{\mathrm{op}})^T(\mathbf{v}^{\mathrm{zs}} + \mathbf{v}^{\mathrm{nz}}) \\ &= -\tfrac{1}{2}\mathbf{v}^{\mathrm{zs}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{zs}} - \mathbf{v}^{\mathrm{zs}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{nz}} - \tfrac{1}{2}\mathbf{v}^{\mathrm{nz}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{nz}} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v}^{\mathrm{zs}} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v}^{\mathrm{nz}} \end{aligned}$$

E^{op}, can now be decoupled into a term that is functional on $\mathbf{v}^{\mathrm{nz}}$ only, and a term that is functional on both $\mathbf{v}^{\mathrm{zs}}$ and $\mathbf{v}^{\mathrm{nz}}$.

$$E^{\mathrm{op}} = E^{\mathrm{nz}}(\mathbf{v}^{\mathrm{nz}}) + E^{\mathrm{zz}}(\mathbf{v}^{\mathrm{zs}}, \mathbf{v}^{\mathrm{nz}}) \tag{2.51}$$

where

$$E^{\mathrm{zz}}(\mathbf{v}^{\mathrm{zs}}, \mathbf{v}^{\mathrm{nz}}) = -\tfrac{1}{2}\mathbf{v}^{\mathrm{zs}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{zs}} - \mathbf{v}^{\mathrm{nz}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{zs}} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v}^{\mathrm{zs}} \tag{2.52}$$

$$E^{\mathrm{nz}}(\mathbf{v}^{\mathrm{nz}}) = -\tfrac{1}{2}\mathbf{v}^{\mathrm{nz}T}\mathbf{T}^{\mathrm{op}}\mathbf{v}^{\mathrm{nz}} - (\mathbf{i}^{\mathrm{op}})^T\mathbf{v}^{\mathrm{nz}} \tag{2.53}$$

Assuming that the minimization of E^{cn} is enforcing $\mathbf{v}^{\mathrm{nz}} = \mathbf{s}$, and substituting

$$\begin{aligned}
\mathbf{v}^{\mathrm{nz}} &= \mathbf{s} = (\mathbf{o}^n \otimes \mathbf{a}) \\
\mathbf{v}^{\mathrm{zs}} &= \mathbf{T}^{\mathrm{zs}}\mathbf{v} = (\mathbf{R}^n \otimes \mathbf{R}^m)\mathbf{v} \\
\mathbf{T}^{\mathrm{op}} &= (\mathbf{P} \otimes \mathbf{Q})
\end{aligned}$$

into (2.52) leads to,

$$\begin{aligned}
E^{\mathrm{zz}}(\mathbf{v}^{\mathrm{zs}}, \mathbf{s}) &= -\tfrac{1}{2}\mathbf{v}^T(\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{P} \otimes \mathbf{Q})(\mathbf{R}^n \otimes \mathbf{R}^m)\mathbf{v} \\
&\quad - \left((\mathbf{o}^n \otimes \mathbf{a})^T(\mathbf{P} \otimes \mathbf{Q}) + \mathbf{i}^{\mathrm{op}T}\right)\mathbf{T}^{\mathrm{zs}}\mathbf{v}
\end{aligned}$$

In effect, E^{zz} is functional in $\mathbf{v}^{\mathrm{zs}}$ only, since the $\mathbf{v}^{\mathrm{nz}}$ term is fixed by the minimization of E^{nz}. Using (2.11), (2.10) and (2.12) and letting $E^{\mathrm{opr}} = E^{\mathrm{zz}}(\mathbf{v}^{\mathrm{zs}}, \mathbf{s})$, we get,

$$\begin{aligned}
E^{\mathrm{opr}} &= -\tfrac{1}{2}\mathbf{v}^T\left((\mathbf{R}^n\mathbf{P}\mathbf{R}^n) \otimes (\mathbf{R}^m\mathbf{Q}\mathbf{R}^m)\right)\mathbf{v} \\
&\quad - \left((\mathbf{R}^n\mathbf{P}\mathbf{o}^n) \otimes (\mathbf{R}^m\mathbf{Q}\mathbf{a}) + \mathbf{T}^{\mathrm{zs}}\mathbf{i}^{\mathrm{op}}\right)^T\mathbf{v}
\end{aligned}$$

$$\Rightarrow \boxed{E^{\mathrm{opr}} = -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{opr}}\mathbf{v} - \mathbf{i}^{\mathrm{opr}T}\mathbf{v}} \qquad (2.54)$$

where

$$\begin{aligned}
\mathbf{T}^{\mathrm{opr}} &= (\mathbf{P}^r \otimes \mathbf{Q}^r) && (2.55) \\
\mathbf{P}^{\mathrm{r}} &= \mathbf{R}^n\mathbf{P}\mathbf{R}^n && (2.56) \\
\mathbf{Q}^{\mathrm{r}} &= \mathbf{R}^m\mathbf{Q}\mathbf{R}^m && (2.57) \\
\mathbf{i}^{\mathrm{opr}} &= \mathbf{T}^{\mathrm{zs}}(\mathbf{T}^{\mathrm{op}}\mathbf{s} + \mathbf{i}^{\mathrm{op}}) \\
&= ((\mathbf{R}^n\mathbf{P}\mathbf{o}^n) \otimes (\mathbf{R}^m\mathbf{Q}\mathbf{a})) + \mathbf{T}^{\mathrm{zs}}\mathbf{i}^{\mathrm{op}} && (2.58)
\end{aligned}$$

Hence E^{opr} represents the optimization term that the network minimizes, if it is assumed that the output of the network is being confined to the valid subspace.

Chapter 3

Solution Quality and Network Dynamics

The last chapter was primarily concerned with the constraint E^{cn} part of the Energy function/Hamiltonian associated with EM networks. A new mathematical framework and a geometric 'subspace approach' were developed. With these a robust expression was derived for E^{cn} which ensured convergence to valid solutions of the problem being solved. This chapter now focuses on the optimization term E^{op}, and specifically on how the interaction between the dynamics of the network and the form of E^{op} affects the final solution quality.

The start point of the analysis is the optimization energy term E^{opr} (2.54). In the first section, the eigenvectors and eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ are determined, and a set of equations are derived which express $\mathbf{v}^{\mathrm{zs}}$, $|\mathbf{v}^{\mathrm{zs}}|^2$ and E^{opr} in terms of their components along these eigenvectors. These expressions then form the basis for the rest of the analysis, which has two parts. Firstly, the likely forms of the optimum solutions of a particular combinatorial optimization problem are analyzed in terms of these expressions, and secondly, theoretical predictions are made, again in terms of these expressions, of the form of the solutions that will be achieved, given the dynamics of an EM network. By characterizing the conditions under which the solutions from the first part of the analysis agree with the solutions from the second part, a firm theoretical link is then established between solution quality and network dynamics.

3.1 Eigenvector/value decomposition of $\mathbf{T}^{\mathrm{opr}}$

The first stage in the analysis is to derive a set of expressions for $\mathbf{v}^{\mathrm{zs}}$, $|\mathbf{v}^{\mathrm{zs}}|^2$ and E^{opr} in terms of the eigenvectors and eigenvalues of $\mathbf{T}^{\mathrm{opr}}$.

Let χ_{kl} and $\mathbf{x}^{kl}$ be the eigenvalues and corresponding normalized eigenvectors of $\mathbf{T}^{\mathrm{opr}}$.

Thus,

$$\mathbf{T}^{\mathrm{opr}}\mathbf{x}^{kl} = \chi_{kl}\mathbf{x}^{kl} \tag{3.1}$$

We can now express χ_{kl} and $\mathbf{x}^{kl}$ in terms of the eigenvalues and eigenvectors of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ and $\mathbf{R}^m\mathbf{Q}\mathbf{R}^m$.

Let $\mathbf{w}^k$ be an eigenvector of $\mathbf{P}^{\mathrm{r}} = \mathbf{R}^n\mathbf{P}\mathbf{R}^n$ (2.56) with an eigenvalue of λ_k, and $\mathbf{h}^l$ be an eigenvector of $\mathbf{Q}^{\mathrm{r}} = \mathbf{R}^m\mathbf{Q}\mathbf{R}^m$ (2.57) with an eigenvalue of γ_l. From (2.55) and (2.10) we can see that,

$$\begin{aligned} \mathbf{P}^{\mathrm{r}}\mathbf{w}^k &= \lambda_k\mathbf{w}^k && (3.2)\\ \mathbf{Q}^{\mathrm{r}}\mathbf{h}^l &= \gamma_l\mathbf{h}^l && (3.3)\\ \Rightarrow (\mathbf{P}^r \otimes \mathbf{Q}^r)(\mathbf{w}^k \otimes \mathbf{h}^l) &= \lambda_k\gamma_l(\mathbf{w}^k \otimes \mathbf{h}^l) && (3.4)\\ \Rightarrow \mathbf{T}^{\mathrm{opr}}(\mathbf{w}^k \otimes \mathbf{h}^l) &= \lambda_k\gamma_l(\mathbf{w}^k \otimes \mathbf{h}^l) && \\ \Rightarrow \mathbf{x}^{kl} &= (\mathbf{w}^k \otimes \mathbf{h}^l) && (3.5)\\ \Rightarrow \chi_{kl} &= \lambda_k\gamma_l && (3.6) \end{aligned}$$

Hence $(\mathbf{w}^k \otimes \mathbf{h}^l)$ is an eigenvector of $\mathbf{T}^{\mathrm{opr}}$ with an eigenvalue of $\lambda_k\gamma_l$.

Clearly $\mathbf{R}^n\mathbf{P}\mathbf{R}^n\mathbf{o}^n = \mathbf{R}^m\mathbf{Q}\mathbf{R}^m\mathbf{o}^m = \mathbf{0}$ since $\mathbf{R}\mathbf{o}^n = \mathbf{0}$ (2.21). Thus $\mathbf{o}^n$ and $\mathbf{o}^m$ are eigenvectors of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ and $\mathbf{R}^m\mathbf{Q}\mathbf{R}^m$, both with an eigenvalue of 0. To be consistent with this, we can assign $\mathbf{w}^1$ and $\mathbf{h}^1$ as follows

$$\mathbf{w}^1 = \frac{1}{\sqrt{n}}\mathbf{o}^n \quad \text{and} \quad \lambda_1 = 0 \tag{3.7}$$

$$\mathbf{h}^1 = \frac{1}{\sqrt{m}}\mathbf{o}^m \quad \text{and} \quad \gamma_1 = 0 \tag{3.8}$$

Since $\mathbf{w}^1 \cdots \mathbf{w}^n$ and $\mathbf{h}^1 \cdots \mathbf{h}^m$ form orthogonal sets (a consequence of the fact that they are the normalized eigenvectors of symmetric matrices [1]) it follows from (2.21) that,

$$\mathbf{R}^n\mathbf{w}^k = \begin{cases} 0 & k = 1 \\ \mathbf{w}^k & k = 2 \cdots n \end{cases} \qquad \mathbf{R}^n\mathbf{h}^l = \begin{cases} 0 & l = 1 \\ \mathbf{h}^l & l = 2 \cdots m \end{cases} \tag{3.9}$$

3.1.1 The eigenvector component matrices A and B

Let the $n \times m$ matrices $\mathbf{A}$ and $\mathbf{B}$ be such that A_{kl} and B_{kl} are respectively the components of $\mathbf{v}$ and $\mathbf{i}^{\mathrm{opr}}$ along the normalized eigenvector $\mathbf{x}^{kl}$ of $\mathbf{T}^{\mathrm{opr}}$,

$$\begin{aligned} A_{kl} &= \mathbf{x}^{kl^T}\mathbf{v} && (3.10)\\ B_{kl} &= \mathbf{x}^{kl^T}\mathbf{i}^{\mathrm{opr}} && (3.11) \end{aligned}$$

[1] Although this is true of all the combinatorial problems described in section 1.1, there are cases, such the Viterbi mapping of chapter 6, where $\mathbf{P}$ and $\mathbf{Q}$ are not symmetric and the assumption of orthogonality breaks down.

Note that from (3.9),

$$\begin{aligned}
\mathbf{T}^{\text{zs}}\mathbf{x}^{kl} &= (\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{w}^k \otimes \mathbf{h}^l) \\
&= (\mathbf{R}^n\mathbf{w}^k \otimes \mathbf{R}^m\mathbf{h}^l) \\
\Rightarrow \mathbf{T}^{\text{zs}}\mathbf{x}^{kl} &= \begin{cases} \mathbf{0} & k=1 \quad \text{or} \quad l=1 \\ \mathbf{x}^{kl} & k \geq 2 \quad \text{and} \quad l \geq 2 \end{cases} \qquad (3.12)
\end{aligned}$$

Hence the first row and column of **A** and **B** correspond to the nonzero subspace components of **v** and $\mathbf{i}^{\text{opr}}$, whilst rows $2\cdots n$ and columns $2\cdots m$ correspond to the zerosum subspace components. But from (2.58)

$$\begin{aligned}
\mathbf{i}^{\text{opr}} &= \mathbf{T}^{\text{zs}}(\mathbf{T}^{\text{op}}\mathbf{s} + \mathbf{i}^{\text{op}}) \\
\Rightarrow \mathbf{x}^{kl^T}\mathbf{i}^{\text{opr}} &= (\mathbf{T}^{\text{zs}}\mathbf{x}^{kl})^T(\mathbf{T}^{\text{op}}\mathbf{s} + \mathbf{i}^{\text{op}}) \\
\Rightarrow B_{kl} &= \begin{cases} 0 & k=1 \quad \text{or} \quad l=1 \\ \mathbf{x}^{kl^T}\mathbf{i}^{\text{opr}} & k \geq 2 \quad \text{and} \quad l \geq 2 \end{cases} \qquad (3.13)
\end{aligned}$$

Thus the elements in the first row and column of **B** are zero.

Using the Kronecker product identity (2.14), $\mathbf{v} = \text{vec}(\mathbf{V}^T)$, and (3.5) we can rewrite (3.10) as,

$$\begin{aligned}
A_{kl} &= (\mathbf{w}^k \otimes \mathbf{h}^l)^T\text{vec}(\mathbf{V}^T) \\
\Rightarrow A_{kl} &= \text{vec}(\mathbf{h}^{l^T}\mathbf{V}^T\mathbf{w}^k) \\
\Rightarrow A_{kl} &= \mathbf{w}^{k^T}\mathbf{V}\mathbf{h}^l \qquad (3.14)
\end{aligned}$$

Similarly if we define $\mathbf{I}^{\text{opr}}$ by,

$$\mathbf{i}^{\text{opr}} = \text{vec}(\mathbf{I}^{\text{opr}T}) \qquad (3.15)$$

then it can be seen that,

$$B_{kl} = \mathbf{w}^{k^T}\mathbf{I}^{\text{opr}}\mathbf{h}^l \qquad (3.16)$$

Let the $n \times n$ matrix **W** be such that the k^{th} column of **W** is $\mathbf{w}^k$.
Let the $m \times m$ matrix **H** be such that the l^{th} column of **H** is $\mathbf{h}^l$. i.e.,

$$\mathbf{W}_{.k} = \mathbf{w}^k \quad \text{and} \quad \mathbf{H}_{.l} = \mathbf{h}^l \qquad (3.17)$$

From (2.2), equations (3.14) and (3.16) can now be expressed in terms of **A**, **B**, **W**, and **H**:

$$\mathbf{A} = \mathbf{W}^T\mathbf{V}\mathbf{H} \qquad (3.18)$$

$$\mathbf{B} = \mathbf{W}^T\mathbf{I}^{\text{opr}}\mathbf{H} \qquad (3.19)$$

3.1.2 Expressions for v and $\mathbf{v}^{\text{zs}}$ in terms of A

Equation (3.18) allows us to easily transform between the matrix representation **V** of **v** and the matrix of eigenvector components **A**. Using the orthogonality property of **W** and **H**, i.e,

$$\mathbf{W}\mathbf{W}^T = \mathbf{I}^n \quad \text{and} \quad \mathbf{H}\mathbf{H}^T = \mathbf{I}^m \qquad (3.20)$$

we can now obtain an expression for $\mathbf{V}$ in terms of A_{kl}, $\mathbf{w}^k$ and $\mathbf{h}^l$:

$$\begin{aligned} \mathbf{WAH}^T &= \mathbf{W}(\mathbf{W}^T\mathbf{VH})\mathbf{H}^T \\ \Rightarrow \mathbf{V} &= \mathbf{WAH}^T \qquad (3.21) \\ \Rightarrow V_{ij} &= \sum_{k=1}^{n} W_{ik} \sum_{l=1}^{m} H_{jl} A_{kl} \\ \Rightarrow V_{ij} &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}[\mathbf{w}^k]_i[\mathbf{h}^l]_j \\ \Rightarrow \mathbf{V} &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}\mathbf{w}^k\mathbf{h}^{l^T} \qquad (3.22) \end{aligned}$$

Applying the Kronecker product identity (2.14) and $\mathbf{v} = \text{vec}(\mathbf{V}^T)$ to (3.22), a similar expression can be derived for $\mathbf{v}$:

$$\begin{aligned} \text{vec}(\mathbf{V}^T) &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}\text{vec}\left((\mathbf{w}^k\mathbf{h}^{l^T})^T\right) \\ \mathbf{v} &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}\text{vec}(\mathbf{h}^l.1.\mathbf{w}^{k^T}) \\ \Rightarrow \mathbf{v} &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \qquad (3.23) \end{aligned}$$

But $\mathbf{v}^{\text{zs}} = \mathbf{T}^{\text{zs}}\mathbf{v} = (\mathbf{R}^n \otimes \mathbf{R}^m)\mathbf{v}$, hence using (3.9) $\mathbf{v}^{\text{zs}}$ can be expressed as follows:

$$\begin{aligned} \mathbf{v}^{\text{zs}} &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}(\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{w}^k \otimes \mathbf{h}^l) \\ &= \sum_{k=1}^{n}\sum_{l=1}^{m} A_{kl}(\mathbf{R}^n\mathbf{w}^k \otimes \mathbf{R}^m\mathbf{h}^l) \\ \Rightarrow \mathbf{v}^{\text{zs}} &= \sum_{k=2}^{n}\sum_{l=2}^{m} A_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \qquad (3.24) \end{aligned}$$

Using an identical derivation we can obtain the following expression for $\mathbf{i}^{\text{opr}}$ in terms of the B_{kl}'s:

$$\mathbf{i}^{\text{opr}} = \sum_{k=2}^{n}\sum_{l=2}^{m} B_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l) \qquad (3.25)$$

If we assume $\mathbf{v}$ is confined to the valid subspace, then

$$\begin{aligned} \mathbf{v} &= \mathbf{v}^{\text{zs}} + \mathbf{s} \\ \Rightarrow \mathbf{s} &= \mathbf{v} - \mathbf{v}^{\text{zs}} \end{aligned}$$

Substituting the expressions from (3.23) and (3.24) we can see that,

$$\mathbf{s} = \sum_{k=2}^{n} A_{k1}(\mathbf{w}^k \otimes \mathbf{h}^l) + \sum_{l=2}^{m} A_{1l}(\mathbf{w}^k \otimes \mathbf{h}^l) + A_{11}(\mathbf{w}^1 \otimes \mathbf{h}^1) \qquad (3.26)$$

3.1.3 Expressions for $|\mathbf{v}|^2$ and $|\mathbf{v}^{\mathrm{zs}}|^2$ in terms of A

We can now make use of the above results together with (2.13) and (2.14) to derive expressions for $|\mathbf{v}|^2$ in terms of A_{kl},

$$\begin{aligned}
|\mathbf{v}|^2 &= \mathbf{v}^T\mathbf{v} & \\
&= \mathrm{vec}(\mathbf{V}^T)^T\mathrm{vec}(\mathbf{V}^T) & \text{since } \mathbf{v} = \mathrm{vec}(\mathbf{V}^T) \\
&= \mathrm{trace}(\mathbf{V}\mathbf{V}^T) & \text{using (2.13)} \\
&= \mathrm{trace}(\mathbf{W}\mathbf{A}\mathbf{H}^T\mathbf{H}\mathbf{A}^T\mathbf{W}^T) & \text{using (3.21)} \\
&= \mathrm{trace}(\mathbf{A}\mathbf{H}^T\mathbf{H}\mathbf{A}^T\mathbf{W}^T\mathbf{W}) & \text{since trace}(\mathbf{AB})\text{=trace}(\mathbf{BA}) \\
&= \mathrm{trace}(\mathbf{A}\mathbf{A}^T) & \text{using (3.20)}
\end{aligned}$$

But, $\mathrm{trace}(\mathbf{A}\mathbf{B}^T) = \mathrm{trace}(\mathbf{B}^T\mathbf{A}) = \sum_{k=1}^n\sum_{l=1}^m A_{kl}B_{kl}$, hence

$$\Rightarrow |\mathbf{v}|^2 = \sum_{k=1}^n\sum_{l=1}^m A_{kl}^2 \tag{3.27}$$

Since $\mathbf{v}^{\mathrm{zs}} = \sum_{k=2}^n\sum_{l=2}^m A_{kl}(\mathbf{w}^k \otimes \mathbf{h}^l)$, it follows that,

$$|\mathbf{v}^{\mathrm{zs}}|^2 = \sum_{k=2}^n\sum_{l=2}^m A_{kl}^2 \tag{3.28}$$

3.1.4 Expressions for E^{opr} in terms of A and B

First using $\mathbf{v} = \mathrm{vec}(\mathbf{V}^T)$, (2.13) and (2.15), E^{opr} can be expressed in terms of the trace of matrix products:

$$\begin{aligned}
E^{\mathrm{opr}} &= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{opr}}\mathbf{v} - \mathbf{i}^{\mathrm{opr}T}\mathbf{v} \\
&= -\tfrac{1}{2}\mathrm{vec}(\mathbf{V}^T)^T(\mathbf{P}^r\otimes\mathbf{Q}^r)\mathrm{vec}(\mathbf{V}^T) - \mathrm{vec}\left(\mathbf{I}^{\mathrm{opr}T}\right)\mathrm{vec}(\mathbf{V}^T) \\
\Rightarrow E^{\mathrm{opr}} &= -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{V}\mathbf{Q}^r\mathbf{V}^T\mathbf{P}^{rT}\right) - \mathrm{trace}\left(\mathbf{I}^{\mathrm{opr}}\mathbf{V}^T\right) \qquad (3.29)
\end{aligned}$$

But,

$$\begin{aligned}
\mathbf{Q}^r\mathbf{V}^T\mathbf{P}^{rT} &= \sum_{k=1}^n\sum_{l=1}^m A_{kl}\mathbf{Q}^r\mathbf{h}^l\mathbf{w}^{k^T}\mathbf{P}^{rT} & \text{from (3.22)} \\
&= \sum_{k=1}^n\sum_{l=1}^m A_{kl}\lambda_k\gamma_l\mathbf{h}^l\mathbf{w}^{k^T} & \text{from (3.2) \& (3.3)} \\
&= \sum_{k=1}^n\sum_{l=1}^m C_{kl}\mathbf{h}^l\mathbf{w}^{k^T} & \text{where } C_{kl} = A_{kl}\lambda_k\gamma_l \\
&= \left(\mathbf{W}\mathbf{C}\mathbf{H}^T\right)^T & \text{by comparison with (3.21)\&(3.22)}
\end{aligned}$$

Thus,

$$E^{\mathrm{opr}} = -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{V}\mathbf{Q}^r\mathbf{V}^T\mathbf{P}^{rT}\right) - \mathrm{trace}\left(\mathbf{I}^{\mathrm{opr}}\mathbf{V}^T\right)$$

$$\begin{aligned}
&= -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{WAH}^T\mathbf{HC}^T\mathbf{W}^T\right) - \mathrm{trace}\left(\mathbf{WBH}^T\mathbf{HA}^T\mathbf{W}^T\right) \\
&= -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{AH}^T\mathbf{HC}^T\mathbf{W}^T\mathbf{W}\right) - \mathrm{trace}\left(\mathbf{BH}^T\mathbf{HA}^T\mathbf{W}^T\mathbf{W}\right) \\
&= -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{AC}^T\right) - \mathrm{trace}\left(\mathbf{BA}^T\right) \\
&= -\sum_{k=1}^{n}\sum_{l=1}^{m}(\tfrac{1}{2}A_{kl}C_{kl} + A_{kl}B_{kl}) \\
\Rightarrow E^{\mathrm{opr}} &= -\sum_{k=1}^{n}\sum_{l=1}^{m}(\tfrac{1}{2}\lambda_k\gamma_l A_{kl}^2 + A_{kl}B_{kl})
\end{aligned}$$

By definition $\lambda_1 = \gamma_1 = 0$ and from (3.13) the first row and column of $\mathbf{B}$ are zero, hence,

$$E^{\mathrm{opr}} = -\sum_{k=2}^{n}\sum_{l=2}^{m}(\tfrac{1}{2}\lambda_k\gamma_l A_{kl}^2 + A_{kl}B_{kl}) \tag{3.30}$$

If we now complete the square, we obtain our final expression for E^{opr}:

$$E^{\mathrm{opr}} = -\tfrac{1}{2}\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\gamma_l\left(A_{kl} + \frac{B_{kl}}{\lambda_k\gamma_l}\right)^2 + \text{a constant} \tag{3.31}$$

where the constant term is equal to $\frac{1}{2}\sum_{k=2}^{n}\sum_{l=2}^{m}\frac{1}{\lambda_k\gamma_l}B_{kl}^2$.

3.2 Properties of the Optimum Solution

After the confinement to the valid subspace has been taken into account, it was shown in section 2.5.2 that,

$$E^{\mathrm{op}} = E^{\mathrm{opr}} + \text{a constant}$$

Hence the valid solution that minimizes E^{opr} will also correspond to the optimum solution of the problem being solved. As a simplification, it will be assumed that $\mathbf{i}^{\mathrm{opr}} = \mathbf{0}$ and hence,

$$\begin{aligned}
B_{kl} &= 0 \quad \forall\, k, l \\
\Rightarrow E^{\mathrm{opr}} &= -\sum_{k=2}^{n}\sum_{l=2}^{m}\tfrac{1}{2}\lambda_k\gamma_l A_{kl}^2
\end{aligned} \tag{3.32}$$

This simplified expression for E^{opr} is valid for the TSP and GPP, and will be used from now on. However, it should be noted that for the Hamilton Path and Graph Matching problems $\mathbf{i}^{\mathrm{opr}} \neq \mathbf{0}$, and in these cases much of the following analysis will need to be extended to take account of the presence of this term. A derivation of the value of $\mathbf{i}^{\mathrm{opr}}$ for various problems is given in Appendix A.2.

The aim of the analysis that follows is to establish a connection between the form of the matrix $\mathbf{A}$, and the valid solution which minimizes E^{opr} as expressed in (3.32). Because of the difficulty in handling the discrete nature of the problem, the first step of the analysis will be to approximate the discrete problem of minimizing E^{opr} over $\mathbf{v} = \mathbf{v}(\mathbf{p})$ with a continuous optimization problem over $\mathbf{v} \in \mathcal{R}^N$. Then, in three stages, a series of increasingly stringent conditions will be imposed which gradually limit the continuous problem to a discrete optimization problem over the $\mathbf{v}(\mathbf{p})$'s. At each stage expressions will be derived for the form of $\mathbf{A}$ that corresponds to the $\mathbf{v}$ which minimizes E^{opr}, given the conditions imposed at that stage.

The three conditions that are sequentially imposed are:

(i) $|\mathbf{v}|^2 = n$

(ii) $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$

(iii) $V_{ij} \geq 0 \ \ \forall \ i,j$

Fig 3.2 shows the relationship between these conditions and the sets of values for $\mathbf{v}$ which satisfy them.

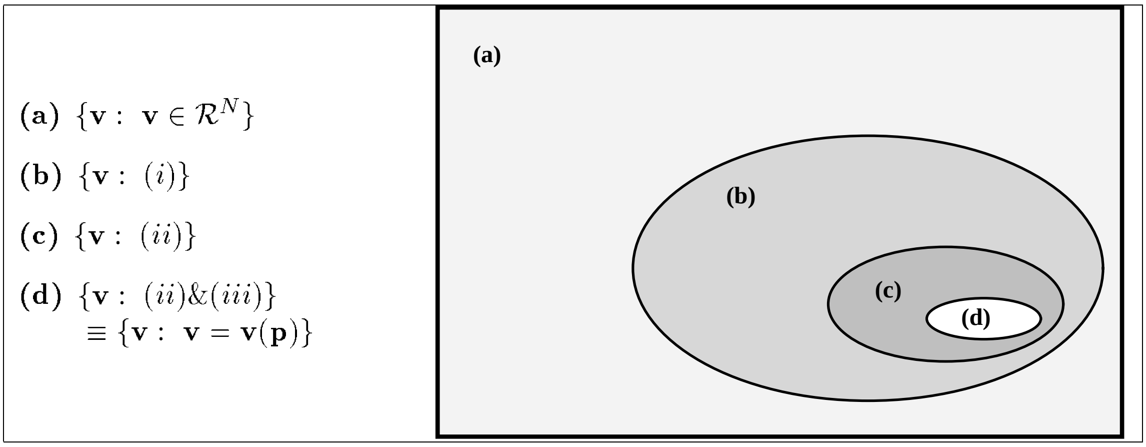

(a) $\{\mathbf{v}: \ \mathbf{v} \in \mathcal{R}^N\}$
(b) $\{\mathbf{v}: \ (i)\}$
(c) $\{\mathbf{v}: \ (ii)\}$
(d) $\{\mathbf{v}: \ (ii)\&(iii)\}$
$\equiv \{\mathbf{v}: \ \mathbf{v} = \mathbf{v}(\mathbf{p})\}$
(a)
(b)
(c)
(d)

Figure 3.1: Relationship between conditions (i),(ii) and (iii) and the sets of values of $\mathbf{v}$ that satisfy them.

3.2.1 Imposing the condition $|\mathbf{v}|^2 = n$

The value of $|\mathbf{v}|^2$ for a valid solution $\mathbf{v}(\mathbf{p})$ is given by the number of elements of $\mathbf{v}(\mathbf{p})$ with a value of 1. Thus, for a n-city TSP and n-node GPP,

$$|\mathbf{v}|^2 = n$$

If we assume $\mathbf{v}$ is confined to the valid subspace then

$$\mathbf{v} = \mathbf{v}^{\mathrm{zs}} + \mathbf{s} \quad \text{where} \quad \mathbf{v}^{\mathrm{zs}^T}\mathbf{s} = 0$$

Hence,

$$|\mathbf{v}|^2 = |\mathbf{v}^{\mathrm{zs}}|^2 + |\mathbf{s}|^2$$

From Table 2.1 it can be seen that for these problems,

$$\begin{aligned} |\mathbf{s}|^2 &= \frac{n}{m} \\ \Rightarrow |\mathbf{v}^{\mathrm{zs}}|^2 &= n - \frac{n}{m} \end{aligned} \tag{3.33}$$

Thus, if $\mathbf{v}$ is a TSP or GPP valid solution, then $\mathbf{v}^{\mathrm{zs}}$ lies on a hypersphere of radius $\sqrt{n - \frac{n}{m}}$. Let us now assume that $\mathbf{v}^{\mathrm{zs}}$ is free to lie anywhere on this hypersphere. Using (3.28) it can be seen that this corresponds to:

$$\sum_{k=2}^{n} \sum_{l=2}^{m} A_{kl}^2 = n - \frac{n}{m} \tag{3.34}$$

Hence, in order to satisfy this equation and minimize (3.32), it follows that

$$A_{kl}^2 = \begin{cases} n - \frac{n}{m} & k = k' \text{ and } l = l' \\ 0 & k \neq k' \text{ or } \ l \neq l' \end{cases} \tag{3.35}$$

where $\chi_{k'l'} = \lambda_{k'}\gamma_{l'}$ is the largest positive eigenvalue of $\mathbf{T}^{\mathrm{opr}}$.

Let $\lambda_2 \cdots \lambda_n$ and $\gamma_2 \cdots \gamma_m$ be given in order of decreasing size. i.e.

$$\lambda_2 > \lambda_3 > \quad \cdots \quad > \lambda_n \tag{3.36}$$

$$\gamma_2 > \gamma_3 > \quad \cdots \quad > \gamma_m \tag{3.37}$$

Assuming that either $\lambda_n \geq 0$ or $\gamma_m \geq 0$, we can see that,

$$k' = l' = 2 \tag{3.38}$$

Otherwise, if both λ_n and γ_m are negative then it is possible that χ_{nm} is the largest eigenvalue.

Unfortunately, (3.35) is derived by allowing $\mathbf{v}^{\mathrm{zs}}$ to lie anywhere on the hypersphere of radius $\sqrt{n - \frac{n}{m}}$, whereas for a valid solution $\mathbf{v}^{\mathrm{zs}} + \mathbf{s}$ must also correspond to a hypercube corner. However, we can still see intuitively that (3.35) confirms that there should be a strong correlation between the eigenvector corresponding to the largest eigenvalue of $\mathbf{T}^{\mathrm{opr}}$, and the region of space that $\mathbf{v}$ must lie in to minimize $\mathbf{T}^{\mathrm{opr}}$.

3.2.2 Imposing the condition $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$

The condition $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$ is satisfied by all valid solutions for which $\mathbf{V}(\mathbf{p})$ is a $n \times n$ permutation matrix. Although this is not the case for the GPP, it is true of a large number of combinatorial optimization problems, such as the TSP, Hamilton Path Problem, and Graph Matching Problem.

Consider now the row and column sums of the square of the elements of $\mathbf{A}$. Taking the row sums first and using (3.14) we obtain,

$$\sum_{l=1}^{m} A_{kl}^2 \quad = \quad \sum_{l=1}^{m} (\mathbf{w}^{k^T} \mathbf{V} \mathbf{h}^l)^2$$

$$\begin{aligned} &= \sum_{l=1}^{m} \mathbf{w}^{k^T}\mathbf{V}\mathbf{h}^l\mathbf{h}^{l^T}\mathbf{V}^T\mathbf{w}^k \\ \Rightarrow \sum_{l=1}^{m} A_{kl}^2 &= \mathbf{w}^{k^T}\mathbf{V}\left(\sum_{l=1}^{m}\mathbf{h}^l\mathbf{h}^{l^T}\right)\mathbf{V}^T\mathbf{w}^k \end{aligned}$$

Since $\mathbf{h}^1 \cdots \mathbf{h}^m$ form an orthogonal set spanning m dimensional space,

$$\sum_{l=1}^{m}\mathbf{h}^l\mathbf{h}^{l^T} = \mathbf{I}^m$$

hence

$$\sum_{l=1}^{m} A_{kl}^2 = \mathbf{w}^{k^T}\mathbf{V}\mathbf{V}^T\mathbf{w}^k \tag{3.39}$$

Using a similar derivation, the column sums reduce to:

$$\sum_{k=1}^{n} A_{kl}^2 = \mathbf{h}^{l^T}\mathbf{V}^T\mathbf{V}\mathbf{h}^l \tag{3.40}$$

If we now assume that $\mathbf{V}$ is a $n \times n$ permutation matrix, and impose the condition $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$, then,

$$\sum_{l=1}^{m} A_{kl}^2 = \mathbf{h}^{l^T}\mathbf{h}^l = 1 \tag{3.41}$$

$$\sum_{k=1}^{n} A_{kl}^2 = \mathbf{w}^{k^T}\mathbf{w}^k = 1 \tag{3.42}$$

This result can now be used to make some predictions about the values of A_{kl}^2 that minimize E^{opr} for the class of problems where $\mathbf{V}(\mathbf{p})$ is a permutation matrix. First, note that we are only interested in rows $2 \cdots n$ and columns $2 \cdots m$, since row and column 1 of $\mathbf{A}$ are fixed by the valid subspace constraint (3.26). It transpires that for the elements in rows $2 \cdots n$ and columns $2 \cdots m$ of $\mathbf{A}$, if the λ_k's and γ_l's are ordered as specified in (3.36) and (3.37) then E^{opr} is minimized when,

$$A_{kl}^2 = \delta_{kl} \tag{3.43}$$

For proof of this see Appendix A.4.

This expression is strictly correct as long as we are only imposing the constraints (3.41) and (3.42). Unfortunately, these are satisfied by any $\mathbf{V}$ for which $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$. While this is true of valid solutions $\mathbf{V}(\mathbf{p})$, it is also true of all $\mathbf{V}$'s which have columns that form an orthogonal basis set.

3.2.3 Imposing the condition $V_{ij} \geq 0 \;\; \forall \; i,j$

In order to strictly enforce the condition that $\mathbf{V}$ is a valid solution hypercube corner, we need to impose a final condition, which is that all the elements of $\mathbf{V}$ must be positive. The proof that this condition is sufficient to ensure $\mathbf{v} = \mathbf{v}(\mathbf{p})$ is as follows:

If $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$ then $\mathbf{V}_{.k}{}^T\mathbf{V}_{.l} = 0 \ \ \forall\, k \neq l$. If $\mathbf{V}_{.k}{}^T\mathbf{V}_{.l} = 0$ and $V_{ij} \geq 0 \ \forall\, i, j$, then it must be true that $V_{ik} = 0$ if $V_{il} \neq 0$. The only way that this can be true for all the columns of $\mathbf{V}$ is if there is a single nonzero element for each row and column of $\mathbf{V}$. Since $\mathbf{V}_{.k}{}^T\mathbf{V}_{.k} = 1$, this element must have a value of 1. Thus $\mathbf{V}$ must be a permutation matrix $\mathbf{V}(\mathbf{p})$.

Naturally, once this extra condition is imposed, it can no longer be assumed that (3.43) holds. However, intuitively we can see that although (3.43) may not strictly hold, in order to minimize E^{opr} it is desirable to keep the A^2_{kl}'s as close to δ_{kl} as possible. With this in mind, we can see that for $p < q$, if there is a choice between violating $A^2_{pp} = 1$ and $A^2_{qq} = 1$ in order to satisfy the constraint that the elements of $\mathbf{V}$ are positive, then it is better to violate $A^2_{qq} = 1$ since $\lambda_p\gamma_p > \lambda_q\gamma_q$. In general we would therefore expect that for good cost valid solutions $A^2_{pp} \approx 1$ for small values of p, but as p increases and $\lambda_p\gamma_p$ decreases, we would expect much greater divergence, with many values of A^2_{pq}, $(p \neq q)$, having significant nonzero values.

3.3 Analysis of Network Dynamics

Through an analysis of the dynamics of an EM network, predictions will now be made of the form of the final solution that the network is likely to achieve. Initially, the relationship between the sign of the eigenvalues of $\mathbf{T}^{\text{opr}}$ and the descent minimization of E^{opr} will be analyzed. This will then be extended to account for the relative magnitudes of these eigenvalues, by analyzing a linearized approximation of the general dynamic equation (1.32). Finally, an intuitive argument will be developed of how $\mathbf{v}$ evolves once the effect of the nonlinear output functions $v_i = g(u_i)$ is accounted for.

3.3.1 Evolution of v and descent minimization of E^{opr}

In section 1.3.5 it was shown that both the Hopfield and MFA networks operate so as to perform a descent minimization of the Energy function/Hamiltonian [2]

$$\begin{aligned} E &= E^{op} + E^{cn} + \eta G && (3.44)\\ \text{where } G &= \sum_i \int_0^{v_i} g^{-1}(\alpha)d\alpha && (3.45)\\ \text{and } \nabla_{\mathbf{v}} G &= \mathbf{u} \end{aligned}$$

If we assume that the minimization of E^{cn} ensures $\mathbf{v}$ is confined to the valid subspace, then using the analysis in section 2.5.2, this reduces to:

$$E = E^{\text{opr}} + \eta G + \text{ a constant}$$

[2] This is only strictly true for the continuous Hopfield network and MFA network without normalization. For the MFA network with normalization, $g()$ is a function of several variables (see Table 1.1), and it is not possible to easily construct a function G with the property $\nabla_{\mathbf{V}} G = \mathbf{u}$. However if we view the normalization modification as just a way of ensuring confinement to the valid subspace, the analysis of the MFA network without normalization can be used to approximate the network with normalization.

Substituting for E^{opr} using (3.31) we get,

$$E = -\tfrac{1}{2}\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\gamma_l\left(A_{kl} + \frac{B_{kl}}{\lambda_k\gamma_l}\right)^2 + \eta G + \text{ a constant} \tag{3.46}$$

Since $g(u)$ is a positive nondecreasing function (see footnote 2), G will be a convex function which is minimized when $\mathbf{v} = \mathbf{0}$. Thus it is the minimization of E^{opr} that causes $\mathbf{v}$ to evolve towards a hypercube corner. Expressions can now be derived for how the descent dynamics of an EM network evolve A_{kl}. In the next section, a linearized analysis of the dynamics will be developed, which although it gives a more detailed picture in the early stages of the network's operation, becomes increasingly inaccurate as the effect of the output nonlinearities becomes more significant. However, the following analysis, which is based only on the assumption that the network operates by descent dynamics (i.e. it evolves $\mathbf{v}$ in a way that always decreases E^{opr}), is always valid, and produces a useful overall picture of how the network operates. We can conveniently divide this minimization into three cases:

(i) $\lambda_k\gamma_l < 0$
From (3.46), it can be seen that in this case, E^{opr} is minimized by setting A_{kl} so that:

$$A_{kl} = -\frac{B_{kl}}{\lambda_k\gamma_l} \tag{3.47}$$

Note that if $\mathbf{i}^{\text{opr}} = 0$ (as in the case of the TSP and GPP) then $B_{kl} = 0$, and E^{opr} is minimized when $A_{kl} = 0$.

(ii) $\lambda_k\gamma_l = 0$
According to equation (3.30), E^{opr} can be expressed as:

$$E = -\sum_{k=2}^{n}\sum_{l=2}^{m}(\tfrac{1}{2}\lambda_k\gamma_l A_{kl}^2 + A_{kl}B_{kl})$$

Thus if $\lambda_k\gamma_l = 0$, then to minimize E^{opr} the network will increase $|A_{kl}|$ whilst ensuring $\text{sign}(A_{kl}) = \text{sign}(B_{kl})$.

(iii) $\lambda_k\gamma_l > 0$
In this case, from (3.46) we can see that E^{opr} is minimised by:

$$A_{kl} \to \infty \tag{3.48}$$

In deriving the above expressions, we have ignored the minimization of G. Following S.Abe [1], one way of dealing with the minimization of G is to perform a Taylor series expansion of it around $\mathbf{v} = \mathbf{v}' + \Delta\mathbf{v}$, and truncate at the quadratic term, giving the following expression:

$$G(\mathbf{v}' + \Delta\mathbf{v}) = G(\mathbf{v}') + \nabla_{\mathbf{v}}G(\mathbf{v}')^T\Delta\mathbf{v} + \tfrac{1}{2}\Delta\mathbf{v}^T\mathcal{H}(G)_{\mathbf{v}=\mathbf{v}'}\Delta\mathbf{v}\cdots$$

where $\mathcal{H}(G)$ is the Hessian of G, given by the matrix $\nabla_{\mathbf{v}}(\nabla_{\mathbf{v}}G)^T$. Recalling that $\nabla_{\mathbf{v}}G = \mathbf{u}$, this reduces to:

$$G(\mathbf{v}' + \Delta\mathbf{v}) = G(\mathbf{v}') + \mathbf{u}'^T\Delta\mathbf{v} + \tfrac{1}{2}\Delta\mathbf{v}^T(\nabla_{\mathbf{v}}\mathbf{u}^T)_{\mathbf{v}=\mathbf{v}'}\Delta\mathbf{v}$$

If this is then substituted for G in (3.46) we obtain an equation which is quadratic in $\Delta\mathbf{v}$:

$$E = -\tfrac{1}{2}\Delta\mathbf{v}^T(\mathbf{T}^{\mathrm{opr}} + \eta\nabla_{\mathbf{v}}\mathbf{u}^T)_{\mathbf{v}=\mathbf{v}'}\Delta\mathbf{v} - (\mathbf{i}^{\mathrm{opr}} + \eta\mathbf{u}' + \mathbf{T}^{\mathrm{opr}}\mathbf{v}')^T\Delta\mathbf{v} + \text{ constant} \qquad (3.49)$$

Unfortunately, $(\nabla_{\mathbf{v}}\mathbf{u}^T)_{\mathbf{v}=\mathbf{v}'}$ is a matrix which is highly dependent on $\mathbf{v}'$, and although it only has elements on the leading diagonal [3], it is still very difficult to analyze. Instead of taking this analysis any further, we can use an intuitive approximation, based on the fact that G is a convex function which is minimized when $\mathbf{v} = \mathbf{0}$. In effect, this simply means that the network will tend to evolve A_{kl} to a value smaller than we would expect given the minimization of E^{opr} alone. A more accurate treatment of the effect of the η term is given in the next section on the linearized dynamics.

3.3.2 Linearized Dynamics

The magnitude of the A_{kl}'s which correspond to the positive eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ cannot be increased indefinitely, since the nonlinear output functions $g()$ will confine $\mathbf{v}$ to the unit hypercube. In addition, the differences in relative magnitude of the positive eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ will affect the rate at which the various A_{kl}'s are increased, and hence the relative magnitude of the A_{kl}'s in the final solution. We can now use an analysis of the linearized dynamics of an EM network to characterise these differences and make predictions of how the A_{kl}'s will evolve in the initial stages of the operation of the network.

As demonstrated in section 1.3.5, the dynamics of both the continuous Hopfield network and the asynchronous discrete MFA network, with or without normalization, can be described by the following equations:

$$\begin{aligned} v_i &= g(u_i) \\ \dot{\mathbf{u}} &= -\eta\mathbf{u} + \frac{1}{T^p}\left(\mathbf{T}\mathbf{v} + \mathbf{i}^b\right) \end{aligned}$$

where Table 1.1 shows settings and substitutions required to convert these equations to describe the appropriate networks. We first need to re-express this dynamic equation so that it is in terms of the $\mathbf{v}^{\mathrm{zs}}$ component only, since $\mathbf{v}^{\mathrm{nz}}$ is fixed by the confinement to the valid subspace:

$$\mathbf{v}^{\mathrm{nz}} = \mathbf{s} = \begin{cases} \frac{1}{m}\mathbf{o}^{nm} & \text{for the GPP} \\ \frac{1}{n}\mathbf{o}^{nn} & \text{for the TSP} \end{cases}$$

If we now multiply both sides of the dynamic equation by $\mathbf{T}^{\mathrm{zs}}$, we can express it in terms of the zerosum subspace components only:

$$\begin{aligned} \mathbf{T}^{\mathrm{zs}}\dot{\mathbf{u}} &= -\eta\mathbf{T}^{\mathrm{zs}}\mathbf{u} + \frac{1}{T^p}\left(\mathbf{T}^{\mathrm{zs}}\mathbf{T}\mathbf{v} + \mathbf{T}^{\mathrm{zs}}\mathbf{i}^b\right) \\ \Rightarrow \dot{\mathbf{u}}^{\mathrm{zs}} &= -\eta\mathbf{u}^{\mathrm{zs}} + \frac{1}{T^p}\left(\mathbf{T}^{\mathrm{zs}}\mathbf{T}\mathbf{v}^{\mathrm{zs}} + \mathbf{T}^{\mathrm{zs}}\mathbf{T}\mathbf{s} + \mathbf{T}^{\mathrm{zs}}\mathbf{i}^b\right) \\ \Rightarrow \dot{\mathbf{u}}^{\mathrm{zs}} &= -\eta\mathbf{u}^{\mathrm{zs}} + \frac{1}{T^p}\left(\mathbf{T}^{\mathrm{zs}}\mathbf{T}\mathbf{T}^{\mathrm{zs}}\mathbf{v}^{\mathrm{zs}} + \mathbf{T}^{\mathrm{zs}}\mathbf{T}\mathbf{s} + \mathbf{T}^{\mathrm{zs}}\mathbf{i}^b\right) \qquad \text{since } \mathbf{T}^{\mathrm{zs}}\mathbf{v}^{\mathrm{zs}} = \mathbf{v}^{\mathrm{zs}} \end{aligned}$$

[3] For the Hopfield network and MFA without normalization network, where u_i is only functional on v_i.

Recalling the definitions of $\mathbf{T}^{\text{opr}}$ (2.55) and $\mathbf{i}^{\text{opr}}$ (2.58) this can be written as:

$$\dot{\mathbf{u}}^{\text{zs}} = -\eta\mathbf{u}^{\text{zs}} + \frac{1}{T^p}\left(\mathbf{T}^{\text{opr}}\mathbf{v}^{\text{zs}} + \mathbf{i}^{\text{opr}}\right) \tag{3.50}$$

Consider the output nonlinearity $v_i = g(u_i)$. For the MFA network with normalization $g()$ is a function of more than one variable, hence to be strictly correct, the expression $v_i = g_i(\mathbf{u})$ should be used. Let,

$$\begin{aligned} [\mathbf{g}(\mathbf{u})]_i &= g_i(\mathbf{u}) & (3.51)\\ \Rightarrow \mathbf{v} &= \mathbf{g}(\mathbf{u}) & (3.52) \end{aligned}$$

and let
$$\mathbf{J} = \nabla_{\mathbf{u}}\mathbf{g}(\mathbf{u}) \tag{3.53}$$

Linearizing $\mathbf{v} = \mathbf{g}(\mathbf{u})$, we get:

$$\Delta\mathbf{v} \approx \mathbf{J}\Delta\mathbf{u} \tag{3.54}$$

Let us assume that the network is started with $\mathbf{v}_{t=0} = \mathbf{s} + \mathbf{v}^{\text{zs}}$, where $|\mathbf{v}^{\text{zs}}| \ll |\mathbf{s}|$. If we substitute $\mathbf{v}^{\text{zs}}$ for $\Delta\mathbf{v}$, and evaluate $\mathbf{J}$ at $\mathbf{v} = \mathbf{s}$, it is shown in Appendix A.3 that equation (3.54) simplifies to:

$$\begin{aligned} \mathbf{u}^{\text{zs}} &= \alpha\mathbf{v}^{\text{zs}}\\ \Rightarrow \dot{\mathbf{u}}^{\text{zs}} &= \alpha\dot{\mathbf{v}}^{\text{zs}} \end{aligned}$$

where $\mathbf{u}^{\text{zs}}$ is the zerosum subspace component of $\mathbf{u}$ and,

$$\alpha = \begin{cases} \dfrac{m^2}{m-1} & \begin{array}{l}\text{for the continuous Hopfield network}\\ \text{and the MFA network without normalization}\end{array}\\ m & \text{for the MFA network with normalization} \end{cases} \tag{3.55}$$

Substituting the above expressions for $\mathbf{u}^{\text{zs}}$ and $\dot{\mathbf{u}}^{\text{zs}}$ into (3.50), we obtain,

$$\begin{aligned} \alpha\dot{\mathbf{v}}^{\text{zs}} &= -\eta\alpha\mathbf{v}^{\text{zs}} + \frac{1}{T^p}\left(\mathbf{T}^{\text{opr}}\mathbf{v}^{\text{zs}} + \mathbf{i}^{\text{opr}}\right)\\ \Rightarrow \dot{\mathbf{v}}^{\text{zs}} &= \mathbf{T}^{\text{lin}}\mathbf{v}^{\text{zs}} + \mathbf{i}^{\text{lin}} & (3.56) \end{aligned}$$

where,

$$\begin{aligned} \mathbf{T}^{\text{lin}} &= \frac{1}{\alpha T^p}\mathbf{T}^{\text{opr}} - \eta\mathbf{I}^{nm} & (3.57)\\ \mathbf{i}^{\text{lin}} &= \frac{1}{\alpha T^p}\mathbf{i}^{\text{opr}} & (3.58) \end{aligned}$$

If we now substitute for $\mathbf{v}^{\text{zs}}$ using (3.24) and $\mathbf{i}^{\text{opr}}$ using (3.25), we can write (3.56) in terms of the A_{kl}'s and B_{kl}'s:

$$\sum_{k=2}^{n}\sum_{l=2}^{m}\frac{\mathrm{d}}{\mathrm{d}t}(A_{kl})\mathbf{x}^{kl} = \sum_{k=2}^{n}\sum_{l=2}^{m}\left[A_{kl}\mathbf{T}^{\text{lin}}\mathbf{x}^{kl} + \frac{1}{\alpha T^p}B_{kl}\mathbf{x}^{kl}\right]$$

$$\begin{aligned}
&= \sum_{k=2}^{n}\sum_{l=2}^{m}\left[A_{kl}\left(\frac{1}{\alpha T^p}\mathbf{T}^{\mathrm{opr}}\mathbf{x}^{kl} - \eta\mathbf{x}^{kl}\right) + \frac{1}{\alpha T^p}B_{kl}\mathbf{x}^{kl}\right] \\
&= \sum_{k=2}^{n}\sum_{l=2}^{m}\left[A_{kl}\left(\frac{1}{\alpha T^p}\chi_{kl} - \eta\right)\mathbf{x}^{kl} + \frac{1}{\alpha T^p}B_{kl}\mathbf{x}^{kl}\right] \\
\Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}(A_{kl}) &= \tilde{\chi}_{kl}A_{kl} + \frac{1}{\alpha T^p}B_{kl} \qquad (3.59)\\
\text{where } \tilde{\chi}_{kl} &= \frac{\chi_{kl}}{\alpha T^p} - \eta \qquad (3.60)
\end{aligned}$$

In effect $\tilde{\chi}_{kl}$ is the eigenvalue of $\mathbf{T}^{\mathrm{lin}}$ corresponding to the eigenvector $\mathbf{x}^{kl}$.

Assuming that

$$\mathbf{v}^{\mathrm{zs}}_{t=0} = \mathbf{v}^{\mathrm{rnd}} = \sum_{k=2}^{n}\sum_{l=2}^{m} A^{(0)}_{kl}\mathbf{x}^{kl} \qquad (3.61)$$

where $\mathbf{v}^{\mathrm{rnd}}$ is a very small random vector in the zerosum subspace, we can now integrate equation (3.59) to obtain an expression for $A_{kl}(t)$. Note that the case where $\tilde{\chi}_{kl} = 0$ has to be treated separately to the case where $\tilde{\chi}_{kl} \neq 0$.

(i) $\tilde{\chi}_{kl} = 0$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}(A_{kl}) &= B_{kl} \\
\Rightarrow \int_{A^{(0)}_{kl}}^{A_{kl}(t)} \mathrm{d}A'_{kl} &= \int_0^t \frac{1}{\alpha T^p}B_{kl}\mathrm{d}t' \\
\Rightarrow A_{kl}(t) &= A^{(0)}_{kl} + \frac{1}{\alpha T^p}B_{kl}t \qquad (3.62)
\end{aligned}$$

(ii) $\tilde{\chi}_{kl} \neq 0$

$$\begin{aligned}
\int_0^t \tilde{\chi}_{kl}\mathrm{d}t' &= \int_{A^{(0)}_{kl}}^{A_{kl}(t)} \frac{\mathrm{d}A'_{kl}}{A'_{kl} + \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}} \\
\Rightarrow \tilde{\chi}_{kl}t &= \log\left(A'_{kl} + \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}\right)\Bigg|_{A^{(0)}_{kl}}^{A_{kl}(t)} \\
\Rightarrow \tilde{\chi}_{kl}t &= \log\left(\frac{A_{kl}(t) + \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}}{A^{(0)}_{kl} + \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}}\right) \\
\Rightarrow A_{kl}(t) &= \left(A^{(0)}_{kl} + \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}\right)\exp(\tilde{\chi}_{kl}t) - \frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}} \qquad (3.63)
\end{aligned}$$

The first thing to note is that if $\tilde{\chi}_{kl} < 0$ then $A_{kl} \to -\frac{B_{kl}}{\alpha T^p\tilde{\chi}_{kl}}$, where $\alpha T^p\tilde{\chi}_{kl} = \chi_{kl} - \frac{\eta}{\alpha T^p}$. This is in agreement with the predictions of case (i) in section 3.2, except that this time the effect of the η term has been accounted for. For the case where $\tilde{\chi}_{kl} > 0$, we can see that as $t \to \infty$, the A_{kl} with the largest positive $\tilde{\chi}_{kl}$ will dominate over all the others. In other words, if we neglect the effect of the nonlinear threshold functions, then as $t \to \infty$

$$\mathbf{v}^{\mathrm{zs}}(t) \to \left(A_{k'l'}^{(0)} + \frac{B_{k'l'}}{\alpha T^p \tilde{\chi}_{k'l'}} \right) \exp(\tilde{\chi}_{k'l'} t) \tag{3.64}$$

where $\tilde{\chi}_{k'l'}$ is the largest positive eigenvalue of $\mathbf{T}^{\mathrm{lin}}$.

3.3.3 Effect of the nonlinear output functions $g()$

The above results can now be extended to take account of the nonlinear output functions $v_i = g(u_i)$. Let us assume that the eigenvalues of $\mathbf{P}^r$ and $\mathbf{Q}^r$ are ordered according to (3.36) and (3.37), i.e.,

$$\lambda_2 > \lambda_3 > \cdots > \lambda_n \quad \text{and} \quad \gamma_2 > \gamma_3 > \cdots > \gamma_m$$

Unless both λ_n and γ_m are negative, then χ_{22} will be the largest positive eigenvalue of $\mathbf{T}^{\mathrm{opr}}$, and consequently $\tilde{\chi}_{22}$ will be the largest positive eigenvalue of $\mathbf{T}^{\mathrm{lin}}$. The following analysis will be based on this assumption, although with minor revision it can be modified to account of both λ_n and γ_m being negative.

According to the integral of the linearized dynamic equation (3.64), the network will initially evolve $\mathbf{v}^{\mathrm{zs}}$ in the direction of $\mathbf{x}^{22}$, which corresponds to the network increasing just A_{22}^2. Thus we would expect the final solution that the network achieves to have a large $\mathbf{x}^{22}$ component. This ties in with the predictions of section 3.2.1, where it was shown that if $\mathbf{v}^{\mathrm{zs}}$ is free to point anywhere on the hypersphere of radius $n - \frac{n}{m}$, then E^{opr} is minimized when:

$$A_{kl} = \begin{cases} n - \frac{n}{m} & k = l = 2 \\ 0 & k \geq 2 \ \&\ l \geq 3 \ \text{ or } \ k \geq 3 \ \&\ l \geq 2 \end{cases}$$

To develop this analysis further, we need to extend equations (3.41) and (3.42) to deal with the cases where we can only assume that $\mathbf{v}$ lies in the valid subspace.

Let $\mathbf{L}$ be a permutation matrix such that if $\mathbf{x} = \mathbf{L}\mathbf{w}^k$ then,

$$x_1 > x_2 > \cdots > x_n$$

Thus from (3.39),

$$\begin{aligned} \sum_{l=1}^{n} A_{kl}^2 &= \mathbf{w}^{k^T} \mathbf{V}\mathbf{V}^T \mathbf{w}^k \\ &= \mathbf{w}^{k^T} \mathbf{L}^T \mathbf{L}\mathbf{V}\mathbf{V}^T \mathbf{L}^T \mathbf{L}\mathbf{w}^k \\ \Rightarrow \sum_{l=1}^{n} A_{kl}^2 &= \mathbf{x}^T \mathbf{D}\mathbf{x} \\ \text{where } \mathbf{D} &= \mathbf{L}\mathbf{V}\mathbf{V}^T \mathbf{L}^T \end{aligned}$$

Since $\mathbf{L}$ is a permutation matrix $\mathbf{L}\mathbf{o}^n = \mathbf{L}^T\mathbf{o}^n = \mathbf{o}^n$. If $\mathbf{V}$ lies in the valid subspace, then it must have column and row sums of 1. Thus $\mathbf{V}\mathbf{o}^n = \mathbf{V}^T\mathbf{o}^n = \mathbf{o}^n$ and it follows that,

$$\mathbf{D}\mathbf{o}^n = \mathbf{L}\mathbf{V}\mathbf{V}^T\mathbf{L}^T\mathbf{o}^n = \mathbf{o}^n$$

Similarly,

$$\mathbf{o}^{nT}\mathbf{D} = \mathbf{o}^{nT}$$

Hence $\mathbf{D}$ will have column and row sums that satisfy the equality conditions of theorem A.1 in Appendix A.4. Since the elements of $\mathbf{V}$ are all positive the elements of $\mathbf{D}$ must also be positive, so $\mathbf{D}$ also satisfies the inequality conditions of theorem A.1. Given that $\mathbf{x}$ satisfies equation (A.5), theorem A.1 can now be invoked to show that $\mathbf{x}^T\mathbf{D}\mathbf{x}$ is maximized when $\mathbf{D} = \mathbf{I}^n$, which is only the case when $\mathbf{V} = \mathbf{V}(\mathbf{p})$. From (3.41), $\mathbf{x}^T\mathbf{D}\mathbf{x} = 1$ when $\mathbf{V} = \mathbf{V}(\mathbf{p})$, thus unless $\mathbf{v}$ is a valid solution, if $\mathbf{v}$ lies in the valid subspace then,

$$\sum_{l=1}^{n} A_{kl}^2 < 1 \tag{3.65}$$

and using a similar analysis for $\sum_{k=1}^{n} A_{kl}^2$, we obtain:

$$\sum_{k=1}^{n} A_{kl}^2 < 1 \tag{3.66}$$

This leads to the phenomenum of blocking, which is that if $A_{kl}^2 \rightarrow 1$ then in order to satisfy the above inequalities, the network dynamics must keep the magnitude of all the other elements of $\mathbf{A}$ on row k and column l very small.

We can see now that although the network dynamics will initially favour introducing only the A_{22} component, as $A_{22}^2 \rightarrow 1$, the network cannot increase $\mathbf{v}$ any further in the direction of $(\mathbf{w}^2 \otimes \mathbf{h}^2)$, and the second row and column of $\mathbf{A}$ will be effectively blocked. Assuming the dynamics are performing a descent of E^{opr}, it now becomes favourable for the dynamics to cause A_{33} to be increased. Similarly once $A_{33}^2 \rightarrow 1$ the other elements of the third row and column of $\mathbf{A}$ will be blocked from increasing and the dynamics will move on to A_{44} and so on. Thus, eventually the network dynamics will favour solutions which satisfy

$$A_{kl}^2 = \delta_{kl}$$

as closely as possible, with the preferred violations being for k's and l's which are large. Therefore the predicted form of the solution achieved by network dynamics agrees with the predicted form of the optimum solution given in section 3.2. Whilst bearing in mind that the above analysis relies on a series of approximations, it can be seen that this establishes a concrete theoretical link between the form of the optimum solution, and the form of the solution that is likely to be achieved by an EM network.

Dependency on direction along each eigenvector

Before moving on to a detailed study of the Euclidean TSP, it is necessary to address a key point that has so far been overlooked; this is that to increase the magnitude of A_{22}, the network can make it either more positive or more negative. This poses an interesting problem: what if, as in the case of Graph Matching problems, there is a unique valid solution corresponding to the optimum solution. Although this solution may well have large components along $(\mathbf{w}^2 \otimes \mathbf{h}^2)$ etc, the sign of these components is going to be fixed.

Thus, unless $\mathbf{i}^{\mathrm{opr}}$ has a significant component along $(\mathbf{w}^2 \otimes \mathbf{h}^2)$ which determines the sign of A_{22}, in its initial stages there is a 50% chance of the network dynamics picking the wrong sign.

This phenomenon has been investigated in [20], where it was concluded that the only obvious solution was to run the network twice, reversing the sign of $\mathbf{v}^{\mathrm{zs}}_{t=0}$ on each run so that both possibilities are tested. Usefully, however, for many problems (e.g the TSP, GPP, and Hamilton Path problem) this difficulty does not arise because although $\mathbf{i}^{\mathrm{opr}} = \mathbf{0}$, there is a many to one mapping between valid solutions and solution cost. For these problems, as will be shown in the next section for the TSP, changing the sign of A_{22} etc, simply switches between equivalent valid solutions. Thus, we can see that in contrast to the conclusions of V.Wilson and G.Pawley[48], who proposed that their very poor results for using Hopfield and Tank's TSP network were due to the cyclic degeneracy of its valid solutions, the opposite is in fact the case: the cyclic degeneracy enhances the performance of the network.

Chapter 4

The Euclidean TSP

This chapter is concerned with developing a more specific and detailed analysis for the Euclidean TSP, which can be used to confirm the analysis of previous chapter. Initially, a set of general expressions for the eigenvalues and eigenvectors of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ and $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$ will be derived, which will then be extended to deal with the issue of cyclic denegeracy in TSP solutions. These expressions are then employed together with the results of computer simulations of EM networks solving Hopfield and Tank's 30 city TSP [25] to confirm the theory developed in both this and the previous chapter.

4.1 Derivation of the eigenvalues of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ and $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$

4.1.1 Diagonalisation of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$

For the Euclidean TSP, $\mathbf{P}$ is a negated distance matrix. Consquently $\mathbf{P}$ is highly specific to the particular instance of the problem being solved. However it is possible to make some general observations about the eigenvalue spectrum of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$. The distance matrix is also closely related to the distance squared matrix, and since it is possible to derive analytic expressions for the latter matrix, in the analysis that follows both cases will be presented.

Let the k dimensional vector $\mathbf{y}^i$ be the position of the i^{th} point in a k dimensional space, out of an array of n points. Let $\mathbf{Y}$ be a $n \times k$ matrix where the i^{th} row of $\mathbf{Y}$ is $\mathbf{y}^i$

Negated Distance Squared Matrix

In this case,

$$\begin{aligned} P_{ij} &= -|\mathbf{y}^i - \mathbf{y}^j|^2 \\ &= -(\mathbf{y}^i - \mathbf{y}^j)^T(\mathbf{y}^i - \mathbf{y}^j) \\ &= -\mathbf{y}^{i^T}\mathbf{y}^i + 2\mathbf{y}^{i^T}\mathbf{y}^j - \mathbf{y}^{j^T}\mathbf{y}^j \end{aligned}$$

Let $a_i = \mathbf{y}^{i^T}\mathbf{y}^i$. We can now write $\mathbf{P}$ in terms of $\mathbf{a}$, $\mathbf{Y}$ and $\mathbf{o}^n$,

$$\mathbf{P} = -\mathbf{o}^n\mathbf{a}^T + 2\mathbf{Y}\mathbf{Y}^T - \mathbf{a}\mathbf{o}^{nT}$$

Thus $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ is given by,

$$\begin{aligned} \mathbf{R}^n\mathbf{P}\mathbf{R}^n &= -\mathbf{R}^n(\mathbf{o}^n\mathbf{a}^T - 2\mathbf{Y}\mathbf{Y}^T + \mathbf{a}\mathbf{o}^{nT})\mathbf{R}^n \\ &= +2\mathbf{R}^n\mathbf{Y}\mathbf{Y}^T\mathbf{R}^n \\ \Rightarrow \mathbf{R}^n\mathbf{P}\mathbf{R}^n &= +2\sum_{i=1}^{k}\mathbf{z}^i\mathbf{z}^{i^T} \qquad (4.1) \\ \text{where} \quad \mathbf{z}^i &= \mathbf{R}^n\mathbf{Y}_{.i} \\ & \quad \text{i.e. } \mathbf{z}^i \text{ is the } i^{\text{th}} \text{ column of } \mathbf{R}^n\mathbf{Y} \end{aligned}$$

It can be seen from (4.1)that, $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ is equal to the sum of the outer products of k, n-dimensional vectors. This means, as long as the $\mathbf{z}^i$'s are linearly independent, that $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ will have k non-zero eigenvalues which will all be positive, and the corresponding eigenvectors will span the subspace spanned by $\mathbf{z}^1 \cdots \mathbf{z}^k$.

Negated Distance Matrix

In this case

$$\begin{aligned} P_{ij} &= -|\mathbf{y}^i - \mathbf{y}^j| \\ &= -\sqrt{(\mathbf{y}^i - \mathbf{y}^j)^T(\mathbf{y}^i - \mathbf{y}^j)} \end{aligned}$$

Unfortunately it is not possible to take the theoretical analysis any further because of the presence of the square root operator. Nevertheless on the basis of large scale experimental evidence, the following behaviour has been observed:

$$\mathbf{x}^T\mathbf{R}^n\mathbf{P}\mathbf{R}^n\mathbf{x} \geq 0 \quad \forall\, \mathbf{x} \in \Re^n \qquad (4.2)$$

In other words $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ is positive semi-definite if $\mathbf{P}$ is a negated Euclidean distance matrix.

By definition (see (3.7)), $\lambda_1 = 0$ since $\mathbf{R}^n\mathbf{w}^1 = \mathbf{0}$. Let the other λ_k's be ordered according to the order given in (3.36). i.e.,

$$\lambda_2 > \lambda_3 > \cdots > \lambda_n \geq 0 \qquad (4.3)$$

The eigenvalues for $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ where $\mathbf{P}$ is the 30×30 negated distance matrix for Hopfield and Tank's 30 city TSP[25], are shown in Fig 4.1.

4.1.2 Diagonalisation of $\mathbf{R}^m\mathbf{Q}\mathbf{R}^m$

From Table 2.6 for a n city TSP, $\mathbf{Q}$ is a $n \times n$ matrix given by,

$$Q_{ij} = \delta_{j,i\oplus 1} + \delta_{j,i\ominus 1} \quad \text{where} \quad \begin{aligned} i \oplus 1 &= \begin{cases} i+1 & i \neq n \\ 1 & i = n \end{cases} \\ i \ominus 1 &= \begin{cases} i-1 & i \neq 1 \\ n & i = 1 \end{cases} \end{aligned}$$

Let $\hat{\mathbf{h}}^l$ be the n-element complex harmonic vector defined by,

$$\hat{h}^l_q = \tfrac{1}{\sqrt{n}} \exp\left(\jmath(l-1)(q-1)\right) \tag{4.4}$$

where

$$\begin{aligned} \jmath &= \sqrt{-1}\frac{2\pi}{n} \\ l,q &\in \{1,2,\cdots n\} \end{aligned}$$

Consider $\mathbf{Q}\hat{\mathbf{h}}^l$,

$$\begin{aligned} [\mathbf{Q}\hat{\mathbf{h}}^l]_p &= \sum_{q=1}^{n} Q_{pq}\hat{h}^l_q \\ &= \tfrac{1}{\sqrt{n}}\sum_{q=1}^{n}(\delta_{q,p\oplus 1}+\delta_{q,p\ominus 1})\exp\left(\jmath(l-1)(q-1)\right) \\ &= \tfrac{1}{\sqrt{n}}[\exp\left(\jmath(l-1)(p-2)\right)+\exp\left(\jmath(l-1)(p)\right)] \\ &= \tfrac{1}{\sqrt{n}}\exp\left(\jmath(l-1)(p-1)\right)[\exp\left(-\jmath(l-1)\right)+\exp\left(+\jmath(l-1)\right)] \\ &= 2\cos\left(\tfrac{2\pi}{n}(l-1)\right)\hat{h}^l_p \end{aligned}$$

$$\Rightarrow \mathbf{Q}\hat{\mathbf{h}}^l = 2\cos\left(\tfrac{2\pi}{n}(l-1)\right)\hat{\mathbf{h}}^l \tag{4.5}$$

Hence $\hat{\mathbf{h}}^l$ is an eigenvector of $\mathbf{Q}$ with an eigenvalue of $\gamma_l = 2\cos\left(\frac{2\pi}{n}(l-1)\right)$.

To simplify the later analysis, we now convert the complex eigenvector set $\hat{\mathbf{h}}^1\cdots\hat{\mathbf{h}}^n$, to a real eigenvector set $\mathbf{h}^1\cdots\mathbf{h}^n$, by the following transformation:

$$\mathbf{h}^l = \tfrac{1}{2}[\mu(l)\hat{\mathbf{h}}^l + (\mu(l)\hat{\mathbf{h}}^l)^*] \tag{4.6}$$

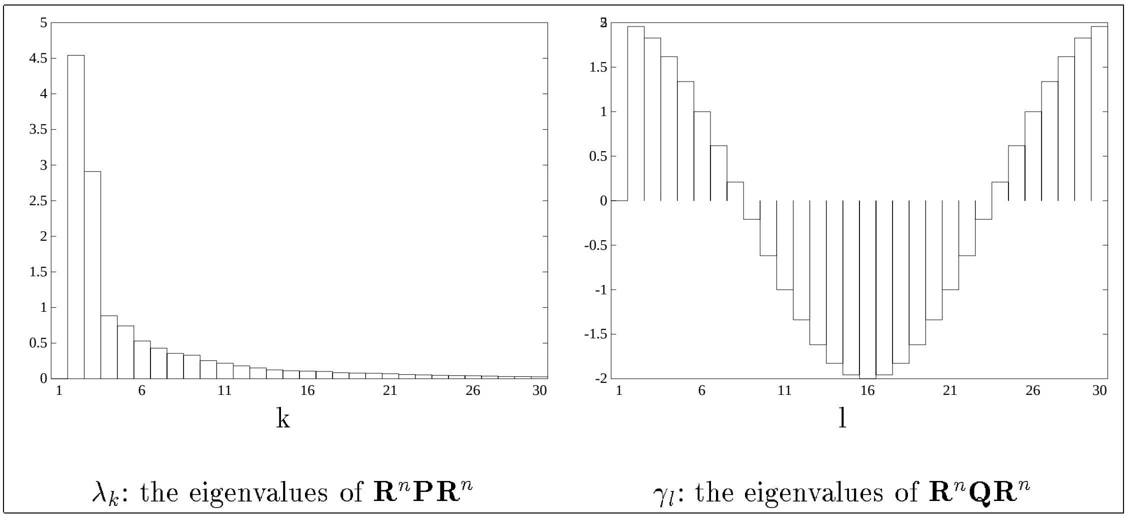
$$\text{where } \mu(l) = \begin{cases} 1 & l-1 = 0, n/2 \\ \sqrt{2} & 0 < l-1 < n/2 \\ -\sqrt{-2} & n/2 < l-1 < n \end{cases} \tag{4.7}$$

Thus,

$$\Rightarrow h^l_p = \begin{cases} \frac{1}{\sqrt{n}}\cos\left(\frac{2\pi}{n}(l-1)(p-1)\right) & l-1 = 0, n/2 \\ \frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{2\pi}{n}(l-1)(p-1)\right) & 0 < l-1 < n/2 \\ \frac{\sqrt{2}}{\sqrt{n}}\sin\left(\frac{2\pi}{n}(n-l+1)(p-1)\right) & n/2 < l-1 < n \end{cases} \tag{4.8}$$

Since $2\cos\left(\frac{2\pi}{n}(l-1)\right)$ and $\mathbf{Q}$ are real, then $\mathbf{Q}(\hat{\mathbf{h}}^l)^* = (\mathbf{Q}\hat{\mathbf{h}}^l)^* = 2\cos\left(\frac{2\pi}{n}(l-1)\right)(\hat{\mathbf{h}}^l)^*$, and it follows that,

$$\mathbf{Q}\mathbf{h}^l = \mathbf{Q}\left(\tfrac{1}{2}\mu(l)\hat{\mathbf{h}}^l + \tfrac{1}{2}(\mu(l)\hat{\mathbf{h}}^l)^*\right) = 2\cos\left(\tfrac{2\pi}{n}(l-1)\right)\mathbf{h}^l$$



λ_k: the eigenvalues of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$

γ_l: the eigenvalues of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$

Figure 4.1: The eigenvalues of $\mathbf{R}^n\mathbf{P}\mathbf{R}^n$ and $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$ for Hopfield's [25] 30 city TSP

From (4.8) we can see that,

$$\mathbf{h}^1 = \frac{1}{\sqrt{n}}\mathbf{o}^n$$

and since $\mathbf{Q}$ is symmetric,

$$\mathbf{h}^{1^T}\mathbf{h}^l = 0 = \mathbf{o}^{n^T}\mathbf{h}^l \quad \forall\, l \neq 1$$

According to (2.21), if $\mathbf{o}^{n^T}\mathbf{h}^l = 0$ then $\mathbf{R}^n\mathbf{h}^l = \mathbf{h}^l$, for $l = 2 \cdots n$ and,

$$\mathbf{R}^n\mathbf{Q}\mathbf{R}^n\mathbf{h}^l = \mathbf{R}^n\mathbf{Q}\mathbf{h}^l = \mathbf{R}^n\gamma_l\mathbf{h}^l = \gamma_l\mathbf{h}^l$$

Thus $\mathbf{h}^l$ is also an eigenvector of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$. In summary we can state:

$$\mathbf{R}^n\mathbf{Q}\mathbf{R}^n\mathbf{h}^l = \gamma_l\mathbf{h}^l$$

where,

$$\gamma_l = \begin{cases} 0 & l = 1 \\ 2\cos\left(\frac{2\pi}{n}(l-1)\right) & l > 1 \end{cases} \tag{4.9}$$

The eigenvalues of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$ for a 30 city TSP are shown in Fig 4.1.

4.2 Degeneracy of the Eigenvalues

There is a slight complication with the analysis of the eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ for the TSP. This is that many of its eigenvalues are two-fold degenerate, which, as will be shown later, is an artifact of the cyclic degeneracy of the TSP solutions. The degeneracy itself arises from the fact that,

$$\cos\left(\tfrac{2\pi}{n}(l-1)\right) = \cos\left(\tfrac{2\pi}{n}(n-l+1)\right) = \cos\left(\tfrac{2\pi}{n}((n-l+2)-1\right)$$

Hence the eigenvalues of $\mathbf{R}^n\mathbf{Q}\mathbf{R}^n$ (4.9) are degenerate, i.e.,

$$\begin{aligned} \gamma_l &= \gamma_{\tilde{l}} \\ \Rightarrow \chi_{kl} &= \chi_{k\tilde{l}} \\ \text{where} \qquad \tilde{l} &= n-l+2 \\ \text{and} \qquad 2 &\leq l < \frac{n+2}{2} \end{aligned} \tag{4.10}$$

Thus, if l satisfies (4.10) then χ_{kl} will be twofold degenerate. Consequently when we talk of the component of $\mathbf{v}$ along an eigenvector which corresponds to a two-fold degenerate eigenvalue, we should really be considering the component in the corresponding eigenplane.

4.2.1 Representing the component in each eigenplane by a complex number Z_{kl}

Let l satisfy (4.10) and $\tilde{l} = n - l + 2$. It can be shown from (4.4) and (4.8) that,

$$\begin{aligned} \mathbf{h}^l &= \mathrm{Re}(\mu(l)\hat{\mathbf{h}}^l) \\ \mathbf{h}^{\tilde{l}} &= \mathrm{Im}(\mu(l)\hat{\mathbf{h}}^l) \end{aligned}$$

The component of $\mathbf{v}$ along $\mathbf{x}^{kl}$ is given by (3.14),

$$A_{kl} = \mathbf{w}^{k^T}\mathbf{V}\mathbf{h}^l$$

Hence,

$$\begin{aligned} A_{kl} &= \mathrm{Re}(\mathbf{w}^{k^T}\mathbf{V}\mu(l)\hat{\mathbf{h}}^l) \\ A_{k\tilde{l}} &= \mathrm{Im}(\mathbf{w}^{k^T}\mathbf{V}\mu(l)\hat{\mathbf{h}}^l) \\ \Rightarrow A_{kl} &= \mathrm{Re}(Z_{kl}) && (4.11) \\ \Rightarrow A_{k\tilde{l}} &= \mathrm{Im}(Z_{kl}) && (4.12) \\ \text{where} \qquad Z_{kl} &= \mathbf{w}^{k^T}\mathbf{V}\mu(l)\hat{\mathbf{h}}^l && (4.13) \end{aligned}$$

Thus we can conveniently view the complex number Z_{kl} as representing the magnitude ($|Z_{kl}|$) and direction ($\arg(Z_{kl})$) of the component of $\mathbf{v}$ in the eigenplane spanned by $\mathbf{x}^{kl}$ and $\mathbf{x}^{k\tilde{l}}$.

4.2.2 Cyclic Degeneracy of TSP Valid Solutions

We can now show that valid solutions which are cyclic permutations of each other contain components of exactly the same magnitude but of differing direction in each eigenplane of

$\mathbf{T}^{\text{lin}}$. This result ties in with the fact that cyclic permutations of a tour should not affect its length.

To analyse the effect of cyclicly permuting the columns of $\mathbf{V}$, let us define a forward cyclic permutation matrix, $\mathbf{C}$:

$$[\mathbf{C}]_{ij} = \delta_{i,j\oplus 1} \tag{4.14}$$

Since $\mathbf{C}$ is a cyclic matrix, $\hat{\mathbf{h}}^l$ is also an eigenvector of it:

$$\begin{aligned}
[\mathbf{C}\hat{\mathbf{h}}^l]_p &= \sum_{q=1}^{n} C_{pq}\hat{h}^l_q \\
&= \sum_{q=1}^{n} \delta_{p,q\oplus 1}\exp\left(\jmath(l-1)(q-1)\right) \\
&= \exp\left(\jmath(l-1)(p-2)\right) \\
&= \exp\left(-\jmath(l-1)\right)\exp\left(\jmath(l-1)(p-1)\right) \\
&= \exp\left(-\jmath(l-1)\right)\hat{h}^l_p
\end{aligned}$$

Thus $\hat{\mathbf{h}}^l$ is an eigenvector of $\mathbf{C}$ with an eigenvalue of $\exp\left(-\jmath(l-1)\right)$.

Let $\mathbf{V}^\circ$ be a permuted version of $\mathbf{V}$, where the columns of $\mathbf{V}$ have been cyclicly shifted p columns forward. i.e,

$$\mathbf{V}^\circ = \mathbf{V}\mathbf{C}^p \tag{4.15}$$

Consider now the complex eigenplane component Z_{kl} of $\mathbf{V}^\circ$. Let this be denoted $Z^\circ{}_{kl}$:

$$\begin{aligned}
Z^\circ{}_{kl} &= \mathbf{w}^{k^T}\mathbf{V}^\circ\mu(l)\hat{\mathbf{h}}^l \\
&= \mathbf{w}^{k^T}\mathbf{V}\mathbf{C}^p\mu(l)\hat{\mathbf{h}}^l
\end{aligned}$$

But $\mathbf{C}\hat{\mathbf{h}}^l = \exp\left(-\jmath(l-1)\right)\hat{\mathbf{h}}^l$, hence

$$\begin{aligned}
\mathbf{C}^p\hat{\mathbf{h}}^l &= \exp\left(-\jmath(l-1)p\right)\hat{\mathbf{h}}^l \\
\Rightarrow Z^\circ{}_{kl} &= \mathbf{w}^{k^T}\mathbf{V}\exp\left(-\jmath(l-1)p\right)\mu(l)\hat{\mathbf{h}}^l \\
\Rightarrow Z^\circ{}_{kl} &= \exp\left(-\jmath(l-1)p\right)Z_{kl}
\end{aligned}$$

Therefore,

$$\begin{aligned}
|Z^\circ{}_{kl}| &= |Z_{kl}| &\quad (4.16)\\
\arg(Z^\circ{}_{kl}) &= \arg(Z_{kl}) + \tfrac{2\pi}{n}(l-1)p &\quad (4.17)
\end{aligned}$$

Equations (4.16) and (4.17) prove that cyclicly permuting the columns of $\mathbf{V}$ does not affect the magnitude of each eigenplane component, but changes the angle of them by $\frac{2\pi}{n}(l-1)p$. [1]

[1] Reversing the order of a tour also leaves its cost unaffected. By defining a matrix $\mathbf{C}^\diamond$ such that $\mathbf{V}(\mathbf{p})\mathbf{C}^\diamond$ reverses the order of the tour specified by $\mathbf{V}(\mathbf{p})$, and using a similar analysis to the one used for $\mathbf{C}^p$, it can be shown that $|Z^\diamond_{kl}| = |Z_{kl}|$ whilst $\arg(Z^\diamond_{kl}) = -\arg(Z_{kl}) - \frac{2\pi}{n}(l-1)p$.

4.2.3 Eigenplanes, Solution Quality and Network Dynamics

For the TSP, many eigenvalues of $\mathbf{T}^{\text{lin}}$ correspond to an eigenplane, hence the analysis in section 3.3.3 has to be altered to take this fact into account. Instead of analyzing the network dynamics in terms of $|A_{kl}|^2$, we have to consider the changes of $|Z_{kl}|^2$. Note that from (4.11) and (4.12),

$$|Z_{kl}|^2 = \begin{cases} A_{kl}^2 & l = 1 \text{ or } l = \frac{n+2}{2} \\ A_{kl}^2 + A_{k\tilde{l}}^2 & 1 < l < \frac{n+2}{2} \end{cases} \qquad \text{where } \tilde{l} = n - l + 2$$

Thus, recalling that $B_{kl} = 0$ for the TSP (see Appendix A.2), equation (3.31) can be written,

$$E^{\text{opr}} = -\sum_{k=2}^{n}\sum_{l=2}^{m} \lambda_k \gamma_l A_{kl}^2 = -\sum_{k=2}^{n}\sum_{l=2}^{\frac{n+2}{2}} \lambda_k \gamma_l |Z_{kl}|^2 \tag{4.18}$$

and the conditions (3.41) and (3.42) become,

$$\sum_{k=1}^{n} |Z_{kl}|^2 = \begin{cases} \sum_{k=1}^{n} A_{kl}^2 & = 1 \quad l = 1 \text{ or } l = \frac{n+2}{2} \\ \sum_{k=1}^{n} \left(A_{kl}^2 + A_{k\tilde{l}}^2\right) & = 2 \quad 1 < l < \frac{n+2}{2} \end{cases} \tag{4.19}$$

$$\sum_{l=0}^{\frac{n+2}{2}} |Z_{kl}|^2 = \sum_{l=1}^{n} A_{kl}^2 = 1 \tag{4.20}$$

In section 3.2 it was shown that the conditions (3.41) and (3.42) are sufficient to ensure that one of the valid solution constraints, $\mathbf{V}\mathbf{V}^T = \mathbf{I}^n$ is enforced. With just this constraint, it can be demonstrated (see Appendix A.4) that E^{opr} is minimized when,

$$A_{kl}^2 = \delta_{kl}$$

In terms of the Z_{kl}'s (see Appendix A.4.2 for proof) this becomes,

$$|Z_{kl}|^2 = \begin{cases} 1 & k = \text{integer part of } \frac{l+2}{2} \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

(Note that as with $\mathbf{A}$, the elements of the first row and column of $\mathbf{Z}$ are fixed by the fact that $\mathbf{v}$ must lie in the valid subspace.)

Once the constraint that the elements of $\mathbf{v}$ have to be positive, is applied, then (4.21) no longer strictly holds. However, as in the case of (3.43), we can see that as far as possible the network will try to satisfy (4.21) for $|Z_{kl}|^2$'s where k and l are small.

In terms of the network dynamics, instead of their being a choice of increasing $|A_{kl}|^2$ by making A_{kl} more positive or more negative, $|Z_{kl}|^2$ can be increased by increasing Z_{kl} in any direction in the Z_{kl} plane. According to (4.17) the cyclic permutations of a valid solution are spread out equally in the complex Z_{kl} plane, thus even though $\arg(Z_{kl})_{t=0}$ may be random, it is not going to change the solution quality, only which of the n equivalent cyclic permutations is eventually chosen.

4.3 A detailed study of Hopfield and Tank's 30 city TSP

The aim in this section is to use the results of a series of computer simulations of a continuous Hopfield network [2] solving Hopfield and Tank's 30 city TSP [25] to confirm the analysis presented in this and the previous chapter.

4.3.1 Correlations between solution cost and $|Z_{kl}|^2$

As an overall confirmation, first consider the correlation for various valid solutions between the $|Z_{kl}|^2$'s corresponding to the largest eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ and the tour length of the solution. Fig 4.2 shows the plots obtained for 1555 valid solutions which have tour lengths ranging from 4.26 (the optimum) to 6.0886. The top plot shows the value of $|Z_{22}|^2+|Z_{32}|^2$, and it is clear that there is a strong correlation between this value and the tour length. Recalling from (4.19) and (4.20) that,

$$\begin{aligned}\sum_{k=1}^{n}|Z_{kl}|^2 &= \begin{cases} 1 & l=1 \text{ or } l=\frac{n+2}{2} \\ 2 & 1<l<\frac{n+2}{2}\end{cases} \\ \sum_{l=0}^{\frac{n+2}{2}}|Z_{kl}|^2 &= 1\end{aligned}$$

we can see that $|Z_{22}|^2+|Z_{32}|^2 \leq 2$, and that $|Z_{kl}|^2 < 1 \ \forall k,l$. However, according to the top plot $|Z_{22}|^2+|Z_{32}|^2 \rightarrow 1.82$, hence both $|Z_{22}|$ and $|Z_{32}|$ must be approaching their maximum value of 1. All this is in complete agreement with the theoretical predictions of this and the previous chapter. Firstly, Z_{22} is the component in the eigenplane corresponding to the largest eigenvalue of $\mathbf{T}^{\mathrm{opr}}$, so we would expect it to be close to its maximum value of 1 for good cost solutions. Secondly, given that for a valid solution $\sum_{k=1}^{n}|Z_{k2}|^2=2$, we would also expect Z_{32} to have a value close to 1, since $\lambda_3\gamma_2$ is much larger than $\lambda_p\gamma_2$ for $p>3$. This is clearly illustrated in Table 4.1 which lists, in magnitude order, the 15 largest positive eigenvalues of $\mathbf{T}^{\mathrm{opr}}$, and which shows that $\lambda_3\gamma_2$ is 3.3 times larger than $\lambda_4\gamma_2$.

According to (4.21), in order to minimize E^{opr} whilst satisfying (4.19) and (4.20) it is desirable that,

$$|Z_{kl}|^2 = \begin{cases} 1 & k = \text{integer part of } \frac{l+2}{2} \\ 0 & \text{otherwise}\end{cases}$$

The second plot of Fig 4.2 shows the sum of all $|Z_{kl}|^2$'s (except for Z_{11} which always equals 1 for any $\mathbf{v}$ on the valid subspace) for which k equals the integer part of $\frac{l+2}{2}$. Although this equation is not strictly satisfied because we also need to impose the constraint that the elements of $\mathbf{v}$ are positive, it is clear form the second plot of Fig 4.2 that there is a good correlation between the value of this sum and the corresponding tour length. The theoretical maximum value for this sum is 29 (30-$|Z_{11}|^2$) which would be achieved if (4.21) is being strictly satisfied. However, the actual value reached for good cost solutions (i.e

[2] Employing a $\mathbf{T}^{\mathrm{cn}}$ specified in section 2.4

Rank	Eigenvalue of $\mathbf{T}^{\text{opr}}$ given by $\lambda_k\gamma_l$	k	l	λ_k	γ_l
1	8.88507	2	2	4.54178	1.9563
2	8.29825	2	3	4.54178	1.82709
3	7.34876	2	4	4.54178	1.61803
4	6.07809	2	5	4.54178	1.33826
5	5.69122	3	2	2.90918	1.9563
6	5.31534	3	3	2.90918	1.82709
7	4.70715	3	4	2.90918	1.61803
8	4.54178	2	6	4.54178	1
9	3.89324	3	5	2.90918	1.33826
10	2.90918	3	6	2.90918	1
11	2.80698	2	7	4.54178	0.618034
12	1.79797	3	7	2.90918	0.618034
13	1.73019	4	2	0.88442	1.9563
14	1.61592	4	3	0.88442	1.82709
15	1.45177	5	2	0.7421	1.9563

Table 4.1: 15 Largest eigenvalues of $\mathbf{T}^{\text{opr}}$ for Hopfield and Tank's 30 city TSP

those with tour lengths < 4.4) is about 5.9. This essentially shows the degree to which (4.21) is violated in order to keep the elements of $\mathbf{v}$ positive. Further evidence of this violation is provided by the final plot in Fig 4.2 which shows the value of $|Z_{12,6}|^2 + |Z_{13,6}|^2$. According to (4.21), in order to minimize E^{opr} this value should approach 2. Clearly this is not the case, and in addition there is very little correlation (if any) between this value and the tour length.

4.3.2 Evolution of the elements of Z

Fig 4.3 show plots of $|Z_{kl}|^2$ obtained at four points in the evolution of $\mathbf{v}$ from its initial random value to final valid solution. Each plot is labelled by the $|\mathbf{v}|^2$, since the value of $|\mathbf{v}|^2$ is a good measure of the stage the network has reached (the value of $|\mathbf{v}|$ ranges from $|\mathbf{v}|^2_{t=0} = |\mathbf{s}| = 1$ to $|\mathbf{v}|^2_{t\to\infty} = |\mathbf{v}(\mathbf{p})| = n$ as it evolves from its initial value to the final solution). The corresponding plots of $\mathbf{V}$ are shown in Fig 4.4.

The plots in Fig 4.3 again confirm the previous predictions relating to how the Z_{kl}'s should evolve. Firstly, from the $|\mathbf{v}|^2 = 2.247$ plot, we can see that $|Z_{22}|$ is increased first until it starts to reach its maximum value of 1, then as shown in the $|\mathbf{v}|^2 = 4.054$ plot, the Z_{32}, Z_{43} and Z_{53} components are put in. As $|\mathbf{v}|^2$ increases, the other components of $\mathbf{Z}$ are gradually introduced until a final valid solution is reached. The $|\mathbf{v}|^2 = 28.826$ plot of Fig 4.3 shows that the network has as far as possible tried to satisfy (4.21), but because of the need to keep the elements of $\mathbf{v}$ positive, (4.21) tends to be violated for Z_{kl}'s where $\lambda_k\gamma_l$ is small.

4.3.3 Geometrical Interpretation

The strong correlations between $|Z_{22}|^2+|Z_{32}|^2$ and the tour length can also be interpreted geometrically. Recalling that $\mathbf{y}^i$ is the position of the i^{th} city and is the i^{th} row of the $n \times k$ matrix $\mathbf{Y}$, let $\tilde{\mathbf{Y}}$ be defined by,

$$\tilde{\mathbf{Y}} = \mathbf{V}^T\mathbf{Y} \tag{4.22}$$

Consider what $\tilde{\mathbf{Y}}$ represents for a Euclidean TSP where the cities lie within the unit square. At $t=0$, if we assume that $\mathbf{v} = \mathbf{s} + \mathbf{v}^{\text{rnd}}$ where $|\mathbf{v}^{\text{rnd}}| \ll |\mathbf{s}|$ then $\mathbf{v}_{t=0} \approx \mathbf{s}$. This corresponds to $\mathbf{V} \approx \frac{1}{n}\mathbf{O}^{nn}$ for the TSP. Hence,

$$\begin{aligned} \tilde{\mathbf{Y}}_{t=0} &\approx \tfrac{1}{n}\mathbf{O}^{nn}\mathbf{Y} \\ \Rightarrow \tilde{\mathbf{Y}}_{t=0} &\approx \mathbf{o}^n[\bar{x},\bar{y}] \end{aligned} \tag{4.23}$$

where $\bar{x}$ and $\bar{y}$ are the average of the x and y co-ordinates of the cities. i.e. $[\bar{x},\bar{y}]$ is the centre of gravity of the cities. If each row of $\tilde{\mathbf{Y}}$ is interpreted as a point, then $\tilde{\mathbf{Y}}$ represents n points all at $[\bar{x},\bar{y}]$. Now consider what happens as $t \to \infty$ and $\mathbf{V}$ approaches a valid solution $\mathbf{V}(\mathbf{p})$.

$$\tilde{\mathbf{Y}}_{t\to\infty} = \mathbf{V}(\mathbf{p})^T\mathbf{Y} \tag{4.24}$$

Since multiplication by $\mathbf{V}(\mathbf{p})^T$ simply permutes the rows of $\mathbf{Y}$ according to the permutation represented by $\mathbf{p}$, we can see that the rows of $\tilde{\mathbf{Y}}_{t\to\infty}$ correspond to the coordinates of the cites in the order of the tour represented by $\mathbf{p}$. Thus if each row of $\tilde{\mathbf{Y}}$ is interpreted as a point, and the points are joined up in order of row number, we will get a plot of the final tour. What is even more interesting is the plot we get when $\mathbf{v}$ is between its initial value of $\mathbf{s}+\mathbf{v}^{\text{rnd}}$ and final solution $\mathbf{v}(\mathbf{p})$. An example for Hopfield and Tank's 30 city TSP is illustrated in Fig 4.5. This figure show various plots of the tours achieved by joining the points corresponding to the rows of $\tilde{\mathbf{Y}}$.

It can be seen in Fig 4.5 that as $|\mathbf{v}|$ increases, the corresponding plot changes from an elongated ellipsoid to a slightly distorted circle at $|\mathbf{v}|^2 = 4.05449$. This circle is then further distorted, until, as $|\mathbf{v}| \to 30$, the plot corresponds to a valid TSP tour. From Fig 4.3, we can see that the intermediate distorted circle stage ($|\mathbf{v}|^2 = 4.05449$) only contains significant components in Z_{22}, Z_{32}, Z_{43} and Z_{53} with the dominant components being Z_{22} and Z_{32}. This means that as long as the network dynamics put in these components first, it will always go through an intermediate distorted circle stage. This distorted circle represents the average of all the tours which follow a reasonably circular path (i.e a path which does not cross over itself). For the case of a 2D Euclidean TSP, paths which satisfy this condition are likely to have a good cost. Conversely, tours with a good cost are going to have a high correlation with the distorted circle and hence will have large components in Z_{22} and Z_{32}. In summary, we can see that the inherent geometrical properties of the Euclidean TSP provide further confirmation that the eigenvectors corresponding to the largest eigenvectors of $\mathbf{T}^{\text{lin}}$ are strongly correlated with the optimum solution.

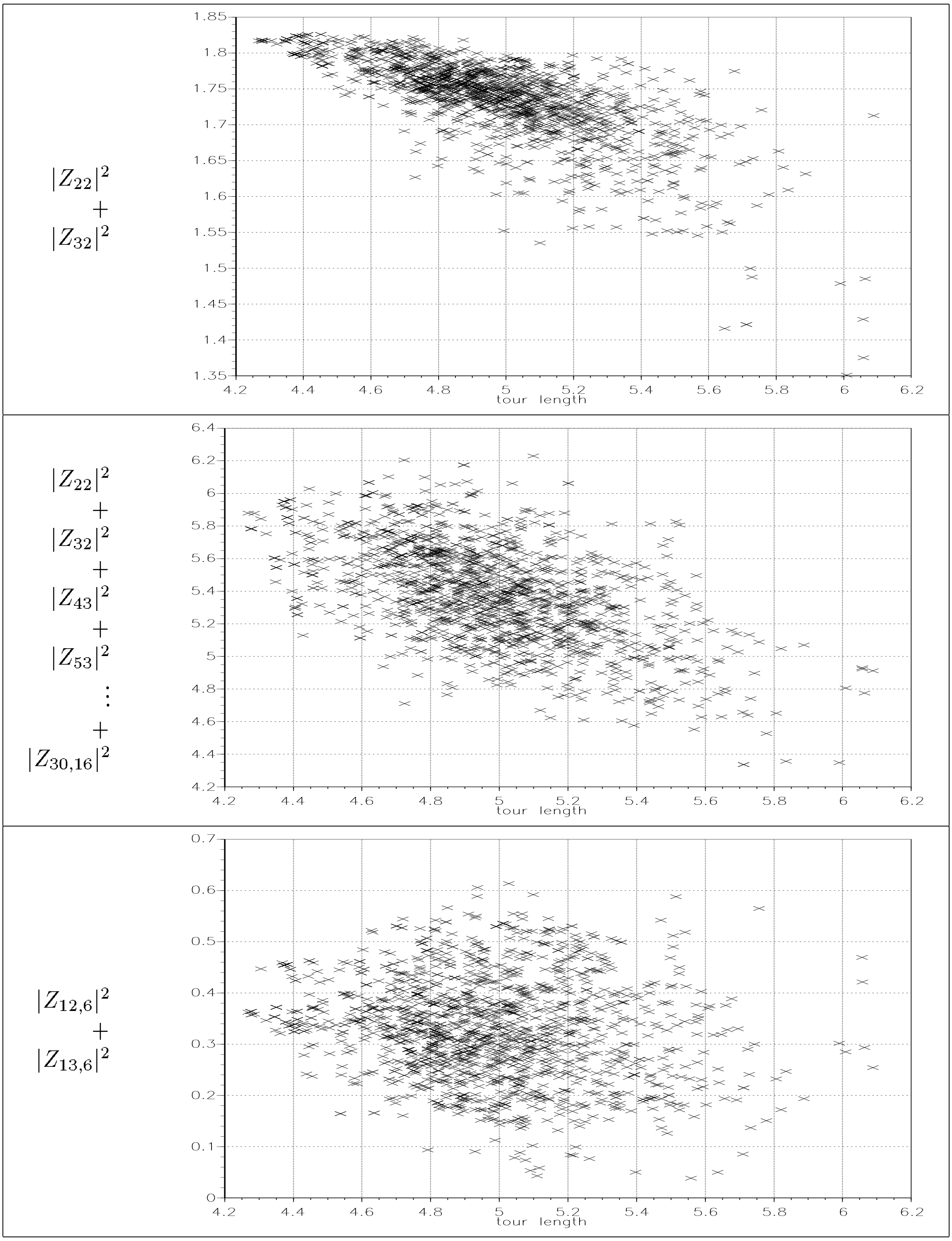
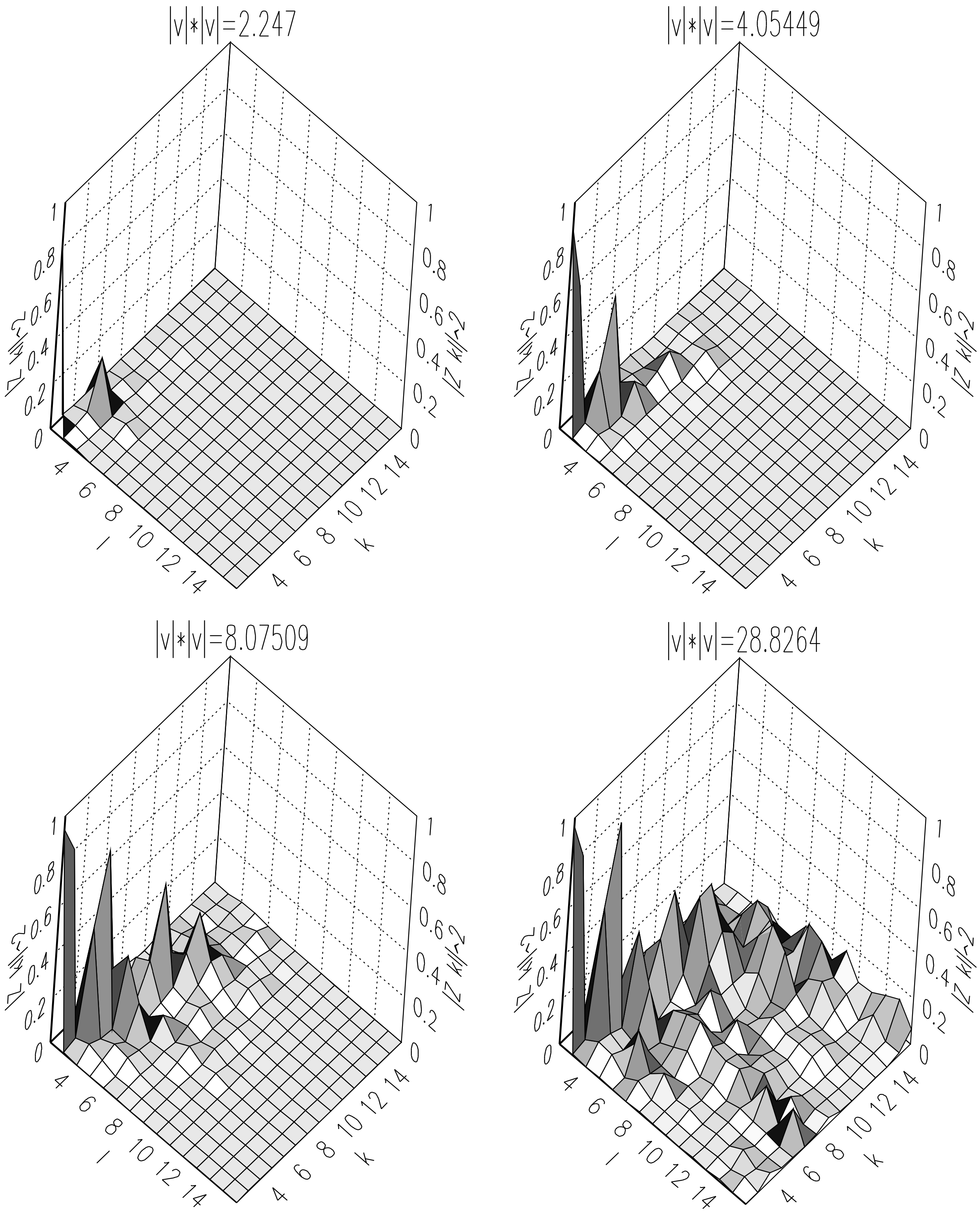


Figure 4.2: Plots of Tour length for Hopfield & Tank's 30 city TSP versus the sum of the magnitude of the components of $\mathbf{v}(\mathbf{p})$ in various eigenplanes of $\mathbf{T}^{\text{lin}}$. (The optimum length is 4.26)

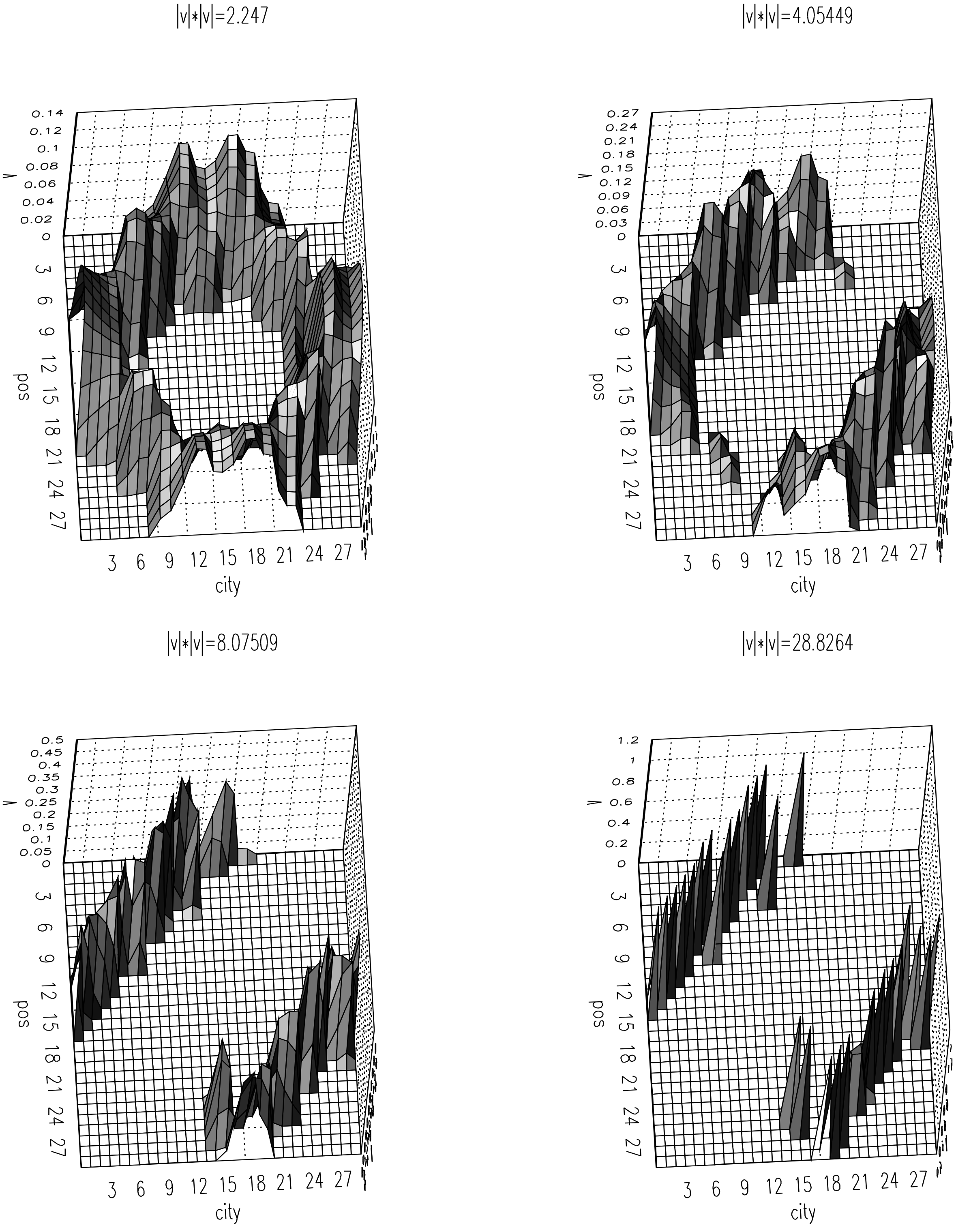

|v|*|v|=2.247

|v|*|v|=4.05449

|v|*|v|=8.07509

|v|*|v|=28.8264

Figure 4.3: Plots showing how for Hopfield and Tank's [25] 30 city TSP, the matrix $\mathbf{Z}$ evolves with increasing $|\mathbf{v}|^2$ ($Z_{kl} = \mathbf{w}^{k^T}\mathbf{V}\mu(l)\hat{\mathbf{h}}^l$).



|v|*|v|=2.247

|v|*|v|=4.05449

|v|*|v|=8.07509

|v|*|v|=28.8264

Figure 4.4: Plots showing how for Hopfield and Tank's [25] 30 city TSP, the matrix representation $\mathbf{V}$, evolves with increasing $|\mathbf{v}|^2$: Cost of final solution is 4.356

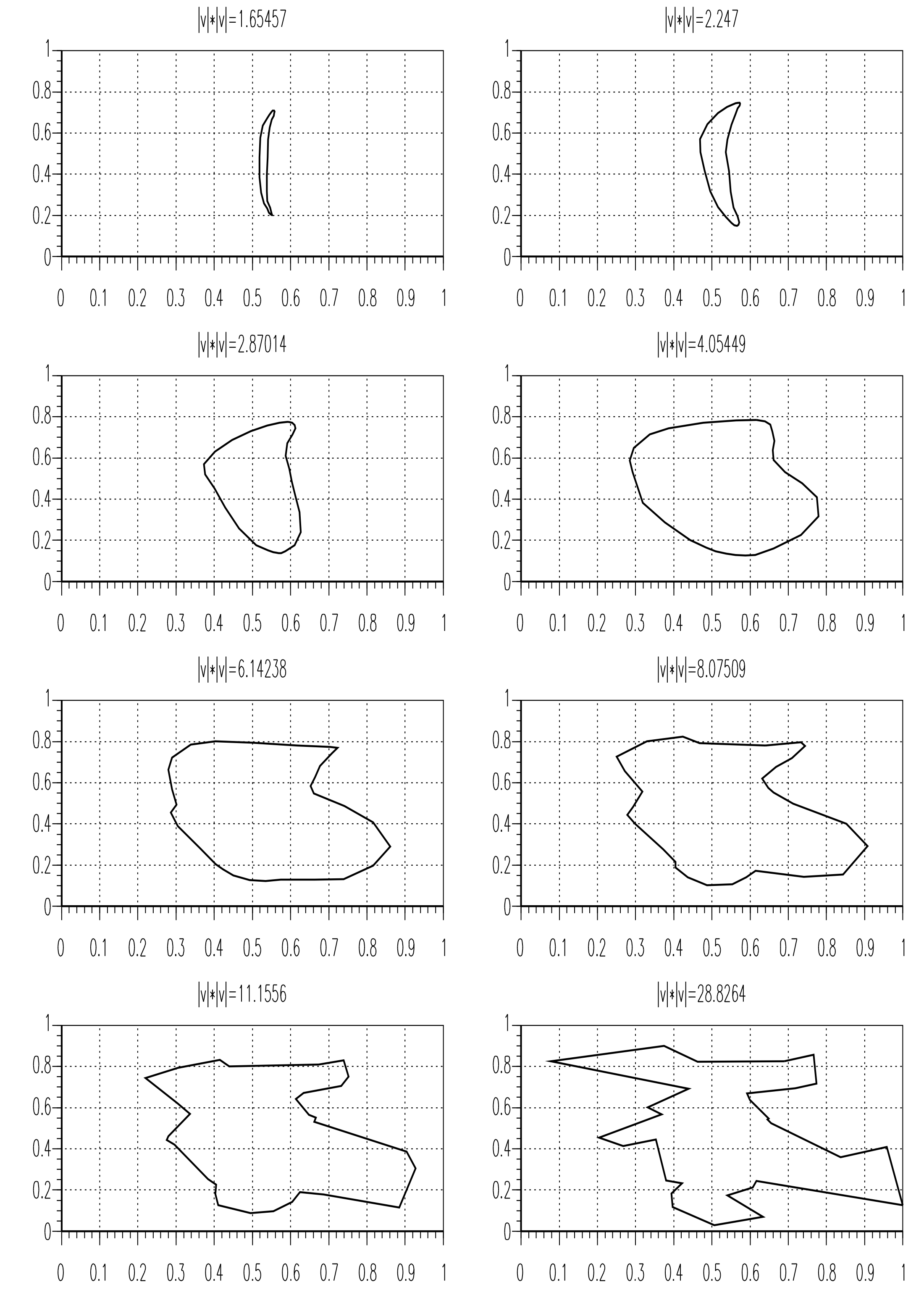


Figure 4.5: Plots generated by joining the points specified by the rows of $\tilde{\mathbf{Y}} = \mathbf{V}^T\mathbf{Y}$ for Hopfield and Tank's [25] 30 city TSP

Chapter 5

Improving the Performance of EM networks

The purpose of this chapter is to draw together the theory developed in the previous chapters, with the aim of specifying modifications that improve the performance of EM networks. The first set of modifications are designed to enhance the average cost of the solutions achieved by these networks, and also to ensure that the network converges exactly to a hypercube corner. The second set employ the 'subspace projection' method to greatly improve the efficiency of computer simulations of the continuous Hopfield network.

5.1 Enhancing the Solution Quality

5.1.1 Critical Temperature and the eigenvalues of $\mathbf{T}^{\mathrm{lin}}$

Before describing the main technique for improving the solution cost, we need to digress slightly to examine the relationship between the sign of the eigenvalues of $\mathbf{T}^{\mathrm{lin}}$ and the phenomenon of critical temperature in MFA networks. This analysis will then form the theoretical basis for the later design modifications.

In the linearized analysis of section 3.3.2, it was shown that if $\tilde{\chi}_{kl} < 0$ then,

$$A_{kl} \rightarrow -\frac{B_{kl}}{\alpha T^p \tilde{\chi}_{kl}} \tag{5.1}$$

Hence, if all the $\tilde{\chi}_{kl}$'s are negative then $\mathbf{v}^{\mathrm{zs}}$ will converge to:

$$\mathbf{v}^{\mathrm{zs}} \rightarrow -\sum_{k=2}^{n}\sum_{l=2}^{m}\frac{B_{kl}}{\alpha T^p \tilde{\chi}_{kl}}\mathbf{x}^{kl}$$

For problems like the TSP and GPP, where $B_{kl} = 0$, this reduces to:

$$\mathbf{v}^{\mathrm{zs}} \rightarrow \mathbf{0}$$

From the definition of $\tilde{\chi}_{kl}$, given in (3.60), it can be seen that all the $\tilde{\chi}_{kl}$'s will be negative if:

$$T^p > T^{\text{crit}} \tag{5.2}$$

where T^{crit} is given by:

$$T^{\text{crit}} = \frac{\lambda_{k'}\gamma_{l'}}{\alpha\eta} \tag{5.3}$$

and $\lambda_{k'}\gamma_{l'}$ is the largest positive eigenvalue of $\mathbf{T}^{\text{opr}}$. Essentially, $\mathbf{T}^{\text{crit}}$ is the critical temperature at which the largest eigenvalue of $\mathbf{T}^{\text{lin}}$ becomes positive, and the network dynamics will start moving $\mathbf{v}$ away from the equilibrium solution (5.1).

Annealing, Critical Temperature and Solution Quality

We are now in a position to examine temperature annealing in MFA networks and the effect it has on the final solution quality. For convenience it will be assumed, as in section 3.2, that λ_k and γ_l are ordered according to:

$$\lambda_2 > \lambda_3 > \cdots > \lambda_n \quad \text{and} \quad \gamma_2 > \gamma_3 > \cdots > \gamma_m$$

and that consequently $\tilde{\chi}_{22}$ is the largest eigenvalue [1] of $\mathbf{T}^{\text{lin}}$. Clearly, just below the critical temperature only $\tilde{\chi}_{22}$ will be positive, and the network will start introducing only the A_{22} component of $\mathbf{v}^{\text{zs}}$. However, as shown in section 3.3.3, the magnitude of A_{22} cannot exceed 1, so as $A_{22} \to 1$, the network dynamics will no longer be able to change $\mathbf{v}^{\text{zs}}$ unless the temperature is lowered and more of the eigenvalues of $\mathbf{T}^{\text{lin}}$ are made positive. Although this is a very simplified picture (in reality the nonlinear output functions have a very significant effect, and as $\mathbf{v}^{\text{zs}}$ changes, so does α), it is consistent with the fact that the temperature has to be continually lowered (annealed) to ensure that a MFA network converges to a final valid solution hypercube corner. We can also see that the annealing process will ensure that the network dynamics are even more likely to favour the eigenvectors corresponding to the largest eigenvalues of $\mathbf{T}^{\text{lin}}$, since in the initial stages of its operation these are the only ones that are positive. This also explains why, if the MFA network without normalization is started at a low enough temperature to ensure that annealing is not required, the results it achieves for the TSP are much worse than if annealing is used [43].

5.1.2 Matrix Graduated Non-Convexity (MGNC)

The phenomenon of critical temperature and annealing in MFA point to the need to be able to adjust the eigenvalues of $\mathbf{T}^{\text{lin}}$, without affecting its eigenvectors, as the key to improving the average solution quality in the continuous Hopfield network. The simplest approach would be to adjust the parameter β which controls the slope of the hyperbolic tangent nonlinearity (1.25) in the continuous Hopfield network. From Table 1.1 it can be seen that increasing β is equivalent to reducing the temperature T^p in MFA networks. Alternatively

[1] The case where both λ_n and γ_m are negative, making it possible for $\lambda_n\gamma_m$ to be the largest eigenvalue, does not affect the overall analysis. However, some modifications may have to be made to properly account for it.

we can adjust η, which is a free parameter in the continuous Hopfield network (see (1.26)). From (3.6) we can see that by changing η we can also make all the eigenvalues of $\mathbf{T}^{\text{lin}}$ more positive or more negative. However, the linearized assumption on which (3.60) is based breaks down once $|\mathbf{v}^{\text{zs}}|$ becomes large enough, and hence the sign of the eigenvalues, will no longer be a simple function of η. Matrix Graduated Non-Convexity [2] is essentially a more robust way of controlling the sign and relative magnitude of these eigenvalues, which is not affected by the nonlinearity of the network dynamics. It operates by adding a scalar multiple of a matrix $\mathbf{\Phi}$ to $\mathbf{T}^{\text{op}}$, so that $\mathbf{T}^{\text{op}}$ is given by,

$$\mathbf{T}^{\text{op}} = (\mathbf{P} \otimes \mathbf{Q}) + \kappa\mathbf{\Phi} \tag{5.4}$$

The matrix $\mathbf{\Phi}$ must have two essential features:

(i) The eigenvectors of $(\mathbf{P}^r \otimes \mathbf{Q}^r)$ should also be the eigenvectors of $\mathbf{T}^{\text{zs}}\mathbf{\Phi}\mathbf{T}^{\text{zs}}$.

(ii) The addition of $\kappa\mathbf{\Phi}$ to $(\mathbf{P} \otimes \mathbf{Q})$ should not affect the relative values of $E^{\text{op}}(\mathbf{v}(\mathbf{p}))$, although the absolute values may vary by a fixed constant. If this condition is not satisfied then the addition of $\kappa\mathbf{\Phi}$ may change the optimum solution, and hence the ability of the network to correctly solve the problem.

In deriving (2.55) it was shown that,

$$\begin{aligned} \mathbf{T}^{\text{opr}} &= \mathbf{T}^{\text{zs}}\mathbf{T}^{\text{op}}\mathbf{T}^{\text{zs}} \\ \Rightarrow \mathbf{T}^{\text{opr}} &= (\mathbf{P}^r \otimes \mathbf{Q}^r) + \kappa\mathbf{T}^{\text{zs}}\mathbf{\Phi}\mathbf{T}^{\text{zs}} \end{aligned} \tag{5.5}$$

where, as defined in (2.56), (2.57) and (2.34)

$$\mathbf{P}^r = \mathbf{R}^n\mathbf{P}\mathbf{R}^n, \quad \mathbf{Q}^r = \mathbf{R}^m\mathbf{Q}\mathbf{R}^m, \text{ and } \mathbf{T}^{\text{zs}} = (\mathbf{R}^n \otimes \mathbf{R}^m)$$

Thus, from (3.5) and (3.4), condition (i) means that,

$$\mathbf{T}^{\text{zs}}\mathbf{\Phi}\mathbf{T}^{\text{zs}}\mathbf{x}^{kl} = \phi_{kl}\mathbf{x}^{kl} \tag{5.6}$$

Using (3.9) it can be shown that,

$$\mathbf{T}^{\text{zs}}\mathbf{x}^{kl} = \begin{cases} 0 & k = 1 \text{ or } l = 1 \\ \mathbf{x}^{kl} & k \geq 2 \ \& \ l \geq 2 \end{cases}$$

From this, (5.6) and (5.5), it follows that,

$$\mathbf{T}^{\text{opr}}\mathbf{x}^{kl} = \begin{cases} \mathbf{0} & k = 1 \text{ or } l = 1 \\ (\lambda_k\gamma_l + \kappa\phi_{kl})\mathbf{x}^{kl} & k \geq 2 \ \& \ l \geq 2 \end{cases} \tag{5.7}$$

Hence if $\mathbf{x}^{kl}$ is an eigenvector of $\mathbf{\Phi}$, then the addition of $\kappa\mathbf{\Phi}$ to $\mathbf{T}^{\text{op}}$ changes the eigenvalues of $\mathbf{T}^{\text{opr}}$ to $\lambda_k\gamma_l + \kappa\phi_{kl}$ without affecting the eigenvectors of $\mathbf{T}^{\text{opr}}$.

[2] The name MGNC comes from the term Graduate Non-Convexity employed by A.Blake & D.Zisserman [11] to describe an algorithm that gradually changes the convexity of a function. In the case of MGNC, the convexity of E^{op} is gradually altered by adding a gradually increasing scalar multiple of the matrix $\mathbf{\Phi}$ to $\mathbf{T}^{\text{op}}$.

Naturally there are many possible forms for Φ which satisfy (5.6) and condition (ii). Which particular version should be used depends on the problem being solved (GPP, TSP, Hamilton Path problem etc), and may also depend upon the particular instance of the problem. Table 5.1 list three of the simplest and most general forms for Φ that satisfy (5.6).

Version	Φ	$\mathbf{T}^{\mathrm{zs}}\Phi\mathbf{T}^{\mathrm{zs}}$	ϕ_{kl}	$\lambda_k\gamma_l + \kappa\phi_{kl}$
1	$\mathbf{I}^{nm}$	$\mathbf{T}^{\mathrm{zs}}$	1	$\lambda_k\gamma_l + \kappa$
2	$(\mathbf{P} \otimes \mathbf{I}^m)$	$(\mathbf{R}^n\mathbf{P}\mathbf{R}^n \otimes \mathbf{R}^m)$	λ_k	$\lambda_k(\gamma_l + \kappa)$
3	$(\mathbf{I}^n \otimes \mathbf{Q})$	$(\mathbf{R}^n \otimes \mathbf{R}^m\mathbf{Q}\mathbf{R}^m)$	γ_l	$(\lambda_k + \kappa)\gamma_l$

Table 5.1: Versions of Φ that leave $E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}))$ unchanged and do not affect the eigenvectors of $\mathbf{T}^{\mathrm{opr}}$

Clearly, version 1 of Φ will not affect the relative values of $E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}))$ since for version 1,

$$\begin{aligned} E^{\mathrm{op}}(\mathbf{v}(\mathbf{p})) &= -\tfrac{1}{2}\mathbf{v}(\mathbf{p})^T((\mathbf{P}\otimes\mathbf{Q}) + \kappa\mathbf{I}^{nm})\mathbf{v}(\mathbf{p}) - \mathbf{i}^{\mathrm{op}\mathbf{v}}(\mathbf{p}) \\ &= -\tfrac{1}{2}\mathbf{v}(\mathbf{p})^T(\mathbf{P}\otimes\mathbf{Q})\mathbf{v}(\mathbf{p}) - \tfrac{1}{2}\kappa\mathbf{v}(\mathbf{p})^T\mathbf{v}(\mathbf{p}) - \mathbf{i}^{\mathrm{op}\mathbf{v}}(\mathbf{p}) \\ &= -\tfrac{1}{2}\mathbf{v}(\mathbf{p})^T(\mathbf{P}\otimes\mathbf{Q})\mathbf{v}(\mathbf{p}) - \mathbf{i}^{\mathrm{op}\mathbf{v}}(\mathbf{p}) - \frac{n\kappa}{2} \end{aligned}$$

Versions 2 and 3 are more specific, and only satisfy condition (ii) for those problems where $\mathbf{V}(\mathbf{p})\mathbf{V}(\mathbf{p})^T = \mathbf{I}^n$. Whilst this is not true for the GPP, it does apply to problems such as the TSP, Hamilton Path problem and Graph Matching problem for which $\mathbf{V}(\mathbf{p})$ is a permutation matrix. The proof is as follows:

First note that since $\mathbf{v}(\mathbf{p}) = \mathrm{vec}(\mathbf{V}(\mathbf{p})^T)$, from (2.15) it follows that,

$$E^{\mathrm{op}} = -\tfrac{1}{2}\mathrm{trace}\left(\mathbf{V}(\mathbf{p})\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) - \mathbf{i}^{\mathrm{op}T}\mathbf{v} \tag{5.8}$$

But,

$$\begin{aligned} (\mathbf{P}\otimes\mathbf{Q}) + \kappa(\mathbf{P}\otimes\mathbf{I}^m) &= \left(\mathbf{P}\otimes(\mathbf{Q} + \kappa\mathbf{I}^m)\right) \\ (\mathbf{P}\otimes\mathbf{Q}) + \kappa(\mathbf{I}^n\otimes\mathbf{Q}) &= \left((\mathbf{P} + \kappa\mathbf{I}^n)\otimes\mathbf{Q}\right) \end{aligned}$$

Note also that since trace$(\mathbf{AB})$ = trace$(\mathbf{BA})$ then

$$\mathrm{trace}\left(\mathbf{V}(\mathbf{p})\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) = \mathrm{trace}\left(\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\mathbf{V}(\mathbf{p})\right)$$

Hence, assuming $\mathbf{V}(\mathbf{p})\mathbf{V}(\mathbf{p})^T = \mathbf{I}^n$, we can see that when $\mathbf{T}^{\mathrm{op}} = (\mathbf{P}\otimes\mathbf{Q}) + \kappa\Phi$, and $\mathbf{V} = \mathbf{V}(\mathbf{p})$, the trace() term for version 2 is given by,

$$\mathrm{trace}\left(\mathbf{V}(\mathbf{p})\left(\mathbf{Q} + \kappa\mathbf{I}^m\right)\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) = \mathrm{trace}\left(\mathbf{V}(\mathbf{p})\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right)$$

$$\begin{aligned}
&+\text{trace}\left(\kappa\mathbf{V}(\mathbf{p})\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) \\
=\;& \text{trace}\left(\mathbf{V}(\mathbf{p})\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) + \kappa\text{trace}\left(\mathbf{P}^T\right)
\end{aligned}$$

and for version 3 it is given by

$$\begin{aligned}
\text{trace}\left(\mathbf{Q}\mathbf{V}(\mathbf{p})^T\left(\mathbf{P}^T + \kappa\mathbf{I}^n\right)\mathbf{V}(\mathbf{p})\right) &= \text{trace}\left(\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\mathbf{V}(\mathbf{p})\right) \\
&\quad +\text{trace}\left(\kappa\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{V}(\mathbf{p})\right) \\
&= \text{trace}\left(\mathbf{V}(\mathbf{p})\mathbf{Q}\mathbf{V}(\mathbf{p})^T\mathbf{P}^T\right) + \kappa\text{trace}\left(\mathbf{Q}\right)
\end{aligned}$$

But trace$(\mathbf{P})$ and trace$(\mathbf{Q})$ are independent of $\mathbf{V}(\mathbf{p})$, hence condition (ii) is satisfied, and the relative values of $E^{\text{op}}(\mathbf{v}(\mathbf{p}))$ are left unaffected.

5.1.3 Selecting and Using the MGNC matrix Φ

As mentioned in the previous subsection, the selection of an appropriate Φ is dependent both on the problem being solved and the particular instance of the problem. The aim now is to discuss which of the versions listed in Table 5.1 is appropriate to which problems, and also how they should be used.

Version 1

Of the three versions listed in Table 5.1, only version 1 is generally applicable. Its operation is to simply add a multiple (κ) of the identity matrix to $\mathbf{T}^{\text{op}}$, which corresponds to adding κ to all the eigenvalues (χ_{kl}) of $\mathbf{T}^{\text{opr}}$ for which $k \neq 1$ & $l \neq 1$. Thus with version 1,

$$\mathbf{T}^{\text{opr}}\mathbf{x}^{kl} = \mathbf{T}^{\text{opr}}(\mathbf{w}^k \otimes \mathbf{h}^l) = (\lambda_k\gamma_l + \kappa)(\mathbf{w}^k \otimes \mathbf{h}^l) \qquad (5.9)$$

If κ is just larger than $-\lambda_2\gamma_2$, then only $\lambda_2\gamma_2 + \kappa$ will be positive, so in a way analogous to the use of critical temperature in MFA networks κ can be used to ensure that the network will only introduce the $\mathbf{x}^{22}$ component at first. If $\mathbf{v}$ stops evolving before a valid solution has been reached, then in a similar way to the annealing of temperature in MFA networks, κ can be increased, making all the eigenvalues more positive. This similarity between increasing κ and annealing temperature has also been noticed by S.P.Eberhardt et al [16] and H. Yanai & Y. Sawada [51]. These authors have proposed a method of changing the elements on the leading diagonal of $\mathbf{T} = \mathbf{T}^{\text{cn}} + \mathbf{T}^{\text{op}}$, which they termed hysteretic annealing. However, like many other researchers who have proposed similar schemes, their main justification is that experimentally it improves the stability and quality of the final result of the network. No attempt is made to prove theoretically the link with temperature annealing in MFA, as has been accomplished in the previous subsection. The advantage of this approach is that it shows there are alternative ways (i.e. versions 2 and 3) of achieving the same effect, which as will be shown next, are more effective for certain problems.

Versions 2 and 3

Unlike version 1, versions 2 and 3 are only applicable to problems such as the TSP, where $\mathbf{V}(\mathbf{p})\mathbf{V}(\mathbf{p})^T = \mathbf{I}^n$. However, in such cases versions 2 and 3 allow us to control the eigenvalues in $\mathbf{T}^{\text{opr}}$ in a way that is much more effective in ensuring the network dynamics try to satisfy equation (3.43), i.e.

$$A_{kl}^2 = \delta_{kl}$$

The importance of this is that (as shown in section 3.2) theoretically the optimum solution should have A_{kl}'s which satisfy (3.43) as closely as possible. Therefore, if the network dynamics find a solution which also has this property, it is reasonable to expect it to be a good solution as well.

To see how the use of version 2 or 3 can improve the performance of the network, consider the value of $\lambda_k\gamma_l + \kappa\phi_{kl}$ given in the fifth column of Table 5.1. For version 2 this is,

$$\lambda_k(\gamma_l + \kappa)$$

Since the eigenvalues of $\mathbf{T}^{\text{opr}}$ have changed from $\lambda_k\gamma_l$ to $\lambda_k(\gamma_l + \kappa)$, it can be seen from (3.30) that,

$$E^{\text{opr}} = -\sum_{k=2}^{n}\sum_{l=2}^{m}\left(\tfrac{1}{2}\lambda_k(\gamma_l + \kappa)A_{kl}^2 + A_{kl}B_{kl}\right) \tag{5.10}$$

For simplicity let us assume that a continuous Hopfield network with $\eta = 0$ and $T^p = 1$ is being used. Further, let us assume, as in the case of the Euclidean TSP (see (4.2)), that $\lambda_k \geq 0 \;\; \forall k$. Clearly, the sign of $\lambda_k(\gamma_l + \kappa)$, and hence the sign of χ_{kl}, will be given by the sign of $(\gamma_l + \kappa)$. Thus, if $\gamma_l + \kappa < 0$, then from (3.47) we can see that as $t \rightarrow \infty$:

$$\begin{aligned} A_{kl} \quad &\rightarrow \quad -\frac{B_{kl}}{\alpha\tilde{\chi}_{kl}} \\ \Rightarrow \sum_{k=1}^{n} A_{kl}^2(t) \quad &\rightarrow \quad \sum_{k=1}^{n}\left(\frac{B_{kl}}{\alpha\tilde{\chi}_{kl}}\right)^2 \end{aligned} \tag{5.11}$$

For problems such as the TSP and GPP, where $B_{kl} = 0$, this reduces to:

$$\sum_{k=1}^{n} A_{kl}^2(t) \rightarrow 0 \tag{5.12}$$

In section 3.2 it was shown that in its initial stages an EM network will tend to introduce the A_{22} component first, increasing it until $A_{22}^2 \approx 1$. At this point the constraint (3.66) will prevent A_{22}^2 from exceeding 1 and since, according to (3.41),

$$\sum_{k=1}^{n} A_{kl}^2 = 1$$

it also means that A_{k2}^2 for $k > 2$ will remain small. It was then proposed in section 3.3.3 that the network dynamics would be forced into increasing the magnitude of the elements in the third column of $\mathbf{A}$ (i.e. the A_{k3}'s) whilst specifically favouring A_{33} since

$\lambda_3 > \lambda_k$, $k > 3$. Unfortunately, in practice, the nonlinear output functions $v_i = g(u_i)$ in an EM network cause the elements in the third, and less so the fourth column of **A** to start increasing well before $A_{22} \approx 1$. We can now see that with the use of version 2 MGNC, this behaviour can be prevented by setting κ so that

$$\gamma_2 > -\kappa > \gamma_3$$

If κ is set in such a way, then from (5.11) it follows that

$$\sum_{k=1}^{n} A_{kl}^2(t) \rightarrow 0 \quad l = 3 \cdots n$$

Hence the network dynamics cannot increase the magnitude of the elements in any of the columns of **A** except the second one. Once $A_{22}^2 \approx 1$ the value of κ can be altered so that,

$$\gamma_3 > -\kappa > \gamma_4$$

which will allow the network to start increasing the magnitude of the elements in the third column of **A**. This procedure is repeated until a final valid solution is reached. By exactly controlling the order in which the network increases the magnitude of the elements in each column of **A**, we are counteracting the nonlinear tendencies of an EM network, which may cause the network to start changing the wrong elements of **A**.

The use of version 3 MGNC is very similar to version 2. The key difference is that the eigenvalues of $\mathbf{T}^{\text{opr}}$ are given by,

$$(\lambda_k + \kappa)\gamma_l$$

Using a similar analysis to that developed above, it can be shown that version 3 operates on the rows of **A** in the same way as version 2 operates on the columns.

An illustration of using version 2 MGNC for the Euclidean TSP

For the Euclidean TSP we can use either version 2 or version 3 MGNC. However, whilst the $\lambda_k \geq 0$ the relative values of the λ_k's vary for each problem instance. In contrast the γ_l's are fixed and are given by (4.9):

$$\gamma_l = \begin{cases} 0 & l = 1 \\ 2\cos\left(\frac{2\pi}{n}(l-1)\right) & l > 1 \end{cases}$$

Thus by setting κ to

$$\kappa = -2\cos\left(\frac{2\pi}{n}(l-1)\right) \tag{5.13}$$

we can ensure that only,

$$\gamma_p + \kappa \geq 0 \text{ and } \gamma_q + \kappa \geq 0$$

where $p \leq l$ and $q \geq n + 2 - l$

The algorithm used to increment κ is basically as outlined above. The initial value of κ is set so that $\gamma_2 \geq -\kappa \geq \gamma_3$. The network is then allowed to run until $|\mathbf{v}|$ converges to a fixed point. At this point κ is increased so that $\gamma_3 \geq -\kappa \geq \gamma_4$, and the process repeated until we get convergence to a final solution.

Due to the two fold degeneracy of γ_l's, it is better to analyse the dynamics in terms of the complex eigenvector component Z_{kl}, as defined and used in section 4. Thus, instead of

$$\sum_{k=1}^{n} A_{kl}^2 = 1$$

we see from (4.19)

$$\sum_{k=1}^{n} |Z_{kl}|^2 = \begin{cases} \sum_{k=1}^{n} A_{kl}^2 & = 1 \quad l = 1 \text{ or } l = \frac{n+2}{2} \\ \sum_{k=1}^{n} \left(A_{kl}^2 + A_{k\tilde{l}}^2 \right) & = 2 \quad 1 < l < \frac{n+2}{2} \end{cases}$$

Fig 5.1 shows plots of $\sum_{k=1}^{n} |Z_{kl}|^2$ against l for 4 different values of $|\mathbf{v}|^2$, together with the corresponding plots of $\gamma_l + \kappa$. These are obtained from a continuous Hopfield network, employing version 2 MGNC, being used to solve Hopfield and Tank's 30 city TSP. For details of the implementation see Appendix B. It can be clearly seen in Fig 5.1 that for any value of $|\mathbf{v}|^2$, if $\gamma_l + \kappa < 0$ then $\sum_{k=1}^{n} Z_{kl}^2 \approx 0$. Also it can be seen that the final value of $\sum_{k=1}^{n} Z_{kl}^2$ as $\mathbf{v} \to \mathbf{v}(\mathbf{p})$ is in agreement with (4.19).

The actual network used is the same as the one used to obtain the plots of Figs 4.3, 4.4 and 4.5, and the values of $|\mathbf{v}|^2$ shown in Fig 5.1 correspond to the values shown in the earlier figures.

Fig 5.2 shows a similar set of plots, except that this time no MGNC has been used. Two important differences can be seen if these plots are compared with Fig 5.1. Firstly, for $4 < |\mathbf{v}|^2 < 10$, the plots in 5.1 are both much smoother and monotonically decreasing with increasing l. In contrast, the plots of Fig 5.2 show irregularities and at some points are not monotonically decreasing. On the other hand, the overall forms of the plots in Figs 5.1 and 5.2 are quite similar. We can see therefore that the use of version 2 MGNC is simply reinforcing the tendency of the network dynamics (which was predicted in section 3.3.3) to increase the magnitude of the elements of Z_{kl} by column, in order of the magnitude of the γ_l's. In practice, however, as is shown in the final experimental results section of this chapter, this extra reinforcement translates directly into a higher final solution quality.

The second main difference is that the network which produced the plots in Fig 5.2 does not actually converge to a hypercube corner. In fact it stabilizes at $|\mathbf{v}|^2 = 14.2$, as shown in the last plot of Fig 5.2. The reason for this is that, with $\kappa = 0$, because $\gamma_l = 2\cos\left(\frac{2\pi}{n}(l-1)\right)$ for $l = 2 \cdots n$, it follows that $\lambda_k \gamma_l$ is negative for

$$\frac{n}{4} < l - 1 < \frac{3n}{4} \tag{5.14}$$

This means we would expect that for all l's that satisfy (5.14):

$$\sum_{k=1}^{n} Z_{kl}^2 \to 0$$

which is approximately what can be seen in the last plot of Fig 5.2.

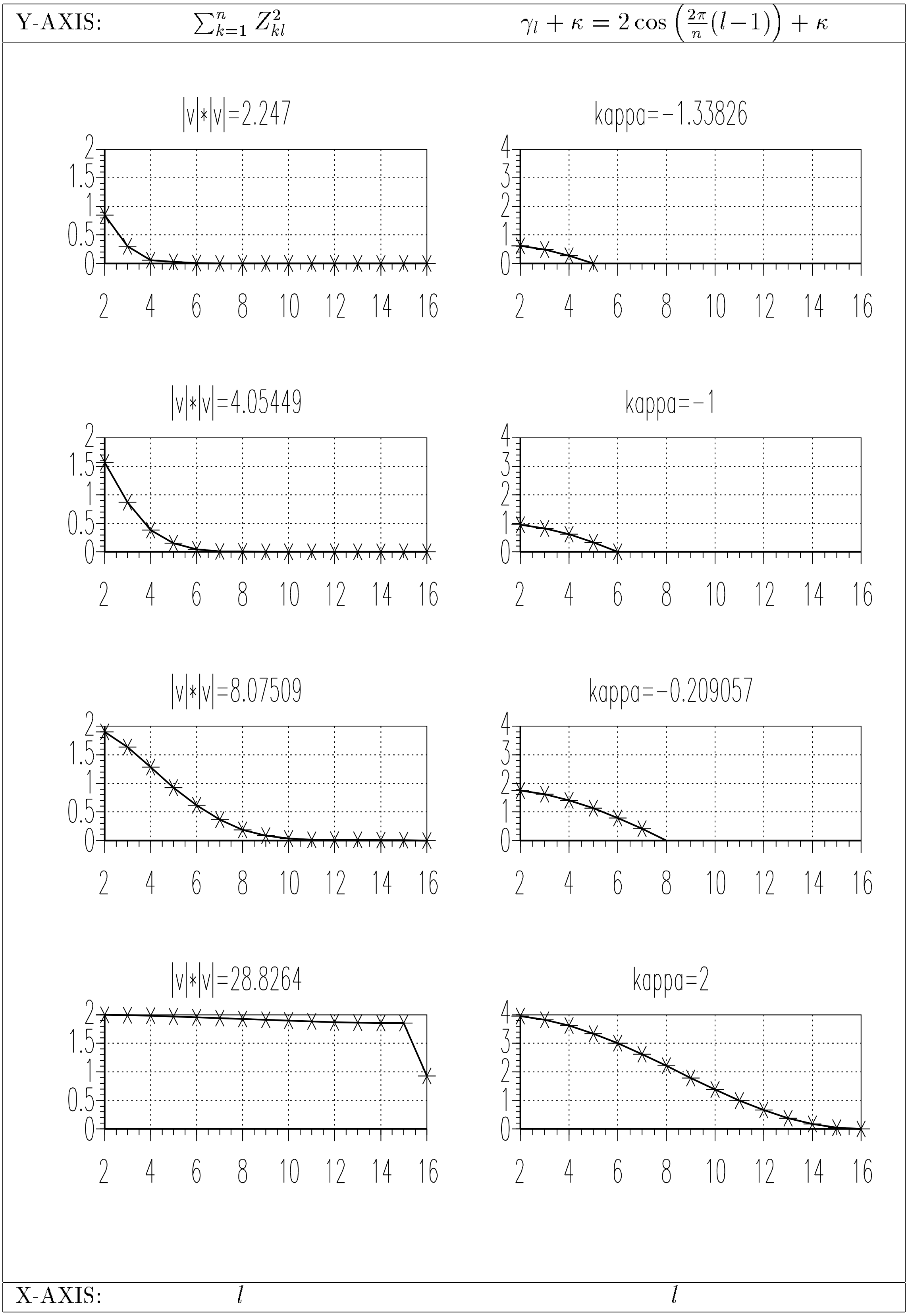


Figure 5.1: Plots of $\sum_{k=1}^{n} Z_{kl}^2$ and $\gamma_l + \kappa$ against l for a continuous Hopfield network with version 2 MGNC, solving Hopfield and Tank's 30 city TSP.

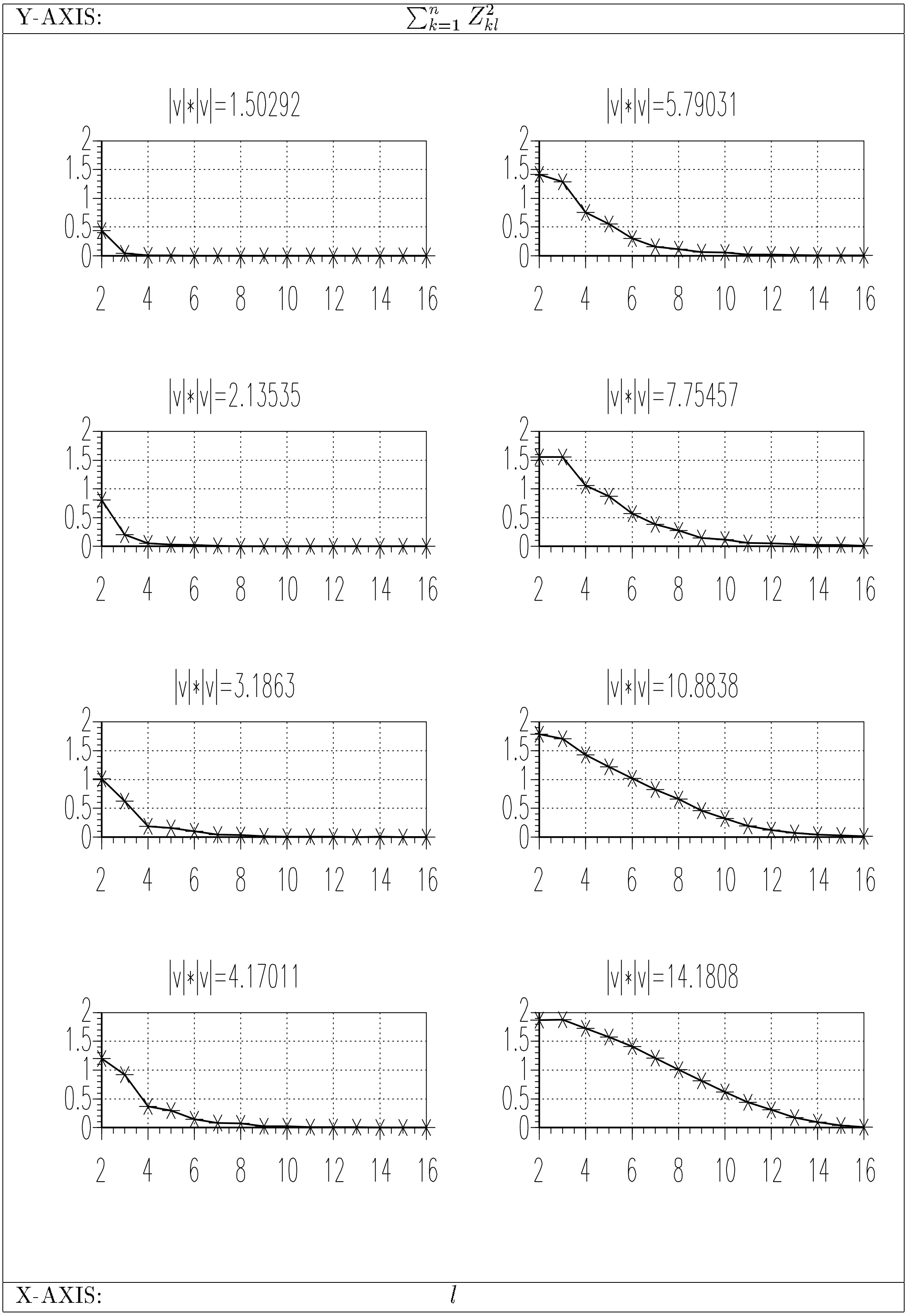


Figure 5.2: Plots of $\sum_{k=1}^{n} Z_{kl}^2$ against $|\mathbf{v}|^2$ for a continuous Hopfield network without any MGNC, solving Hopfield and Tank's 30 city TSP.

5.1.4 Ensuring Convergence to a Hypercube Corner with MGNC

Until now it has been assumed that an EM network should eventually converge to a hypercube corner. However, we can now see from the MGNC analysis that, as shown in Fig 5.2, this is by no means guaranteed. For $\mathbf{v}$ to be a valid solution, then according to (4.19)

$$\sum_{k=1}^{n} |Z_{kl}|^2 = \begin{cases} 1 & l = 1 \text{ or } l = \frac{n+2}{2} \\ 2 & 1 < l < \frac{n+2}{2} \end{cases}$$

Unfortunately, for problems like the Euclidean TSP, $\lambda_k \gamma_l < 0$ if l satisfies (5.14), and as shown in Fig 5.2 the network dynamics will not be able to introduce all the components required for $\mathbf{v}$ to be a valid solution. The only obvious way round this is to use some form of MGNC to shift the values of all the eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ so that they are positive. In the case of the Euclidean TSP this can be achieved by using version 2 MGNC with $\kappa = 2$ (as with the last plot of Fig 5.1). For problems where version 1 MGNC has to be used, the eigenvalues of $\mathbf{T}^{\mathrm{opr}}$ are given by,

$$\lambda_k \gamma_l + \kappa$$

Thus to ensure all these eigenvalues are positive, κ has to be set so that

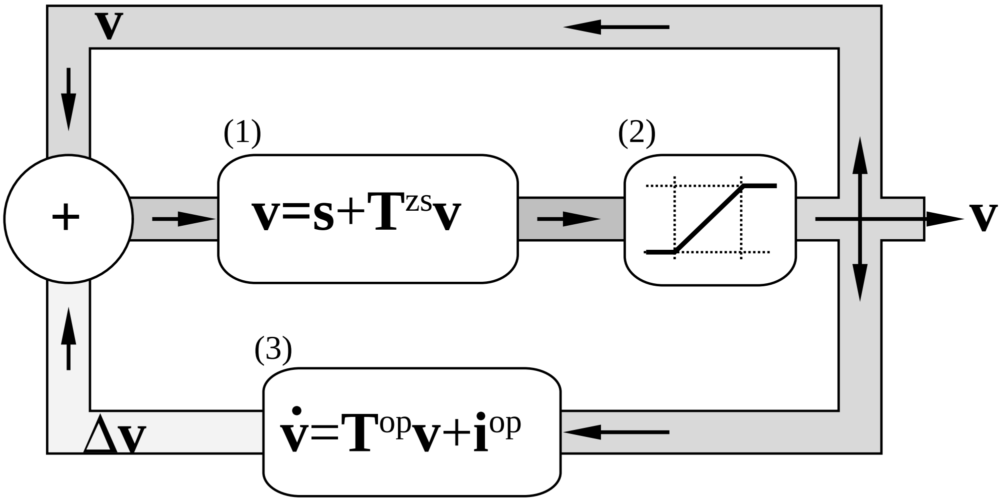
$$\lambda_k \gamma_l + \kappa > 0 \;\; \forall \; k, l$$

In chapter 2 it was proposed that the interference between constraint and optimization terms E^{cn} and E^{op} was the cause of the poor solution quality reported by many researchers [25, 48, 26] for solving the TSP with EM networks. Specifically, it was shown that the interference is due to $\mathbf{T}^{\mathrm{cn}}$ having a degenerate positive eigenvalue (2α for C.Peterson [43] and $2A$ for J.J.Hopfield [25]) corresponding to the zerosum subspace (see Table 2.3). If a correctly formulated E^{cn} term is used, then (see section 2.4) $\mathbf{T}^{\mathrm{cn}}$ should have an eigenvalue of zero in the zerosum subspace. However, we can interpret the presence of the positive eigenvalue in $\mathbf{T}^{\mathrm{cn}}$ as a kind of 'back door' MGNC version 1, with $\kappa = 2\alpha$ for C.Peterson and $\kappa = 2A$ for J.J.Hopfield. Given the need for MGNC to ensure convergence to a hypercube corner, we can see that this 'back door' MGNC is essential if the network is to converge to a valid solution. Unfortunately, as shown in the previous section, setting κ to a high value results in the poor final solutions reported in [25, 48, 26]. Thus we can see that the conflicting needs of keeping κ high to ensure convergence to a hypercube corner, and keeping it low to ensure that the network dynamics initially evolve $\mathbf{v}$ in the right direction, can only be resolved by an algorithm, such as MGNC, which gradually increase the value of κ.

5.2 Valid Subspace Projection

In computer simulations of the continuous Hopfield network, a discrete (Euler) approximation to the continuous differential equation (1.26) has to be used in which,

$$\begin{aligned} \Delta\mathbf{v} &= \mathbf{g}(\mathbf{u} + \Delta\mathbf{u}) - g(\mathbf{u}) \\ \text{where } \Delta\mathbf{u} &= -\Delta t \nabla E = -\Delta t \nabla (E^{\mathrm{cn}} + E^{\mathrm{op}}) \end{aligned}$$



(1) Projection of $\mathbf{v}$ onto valid subspace $\qquad \mathbf{v} = \mathbf{s} + \mathbf{T}^{\mathrm{zs}}\mathbf{v}$

$$\text{where} \;\; \mathbf{s} = (\mathbf{a} \otimes \mathbf{o}^m) + (\mathbf{o}^n \otimes \mathbf{b}) \;\; \text{and} \;\; \mathbf{T}^{\mathrm{zs}} = (\mathbf{R}^n \otimes \mathbf{R}^m)$$

(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

$$\Delta\mathbf{v} = \Delta t \frac{d}{dt}(\mathbf{v}) = \Delta t \dot{\mathbf{v}} = \Delta t(\mathbf{T}^{\mathrm{op}}\mathbf{v} + \mathbf{i}^{\mathrm{op}})$$

Figure 5.3: Schematic diagram of Subspace Projection network implementation

The need to ensure E^{cn} dominates over E^{op} means that most of the change in $\mathbf{v}$ caused by $\Delta\mathbf{v}$ (via $\Delta\mathbf{u}$) will be concerned with the minimisation of E^{cn}, and very little with the minimisation of E^{op}. This makes the simulation very inefficient, since most of the time will be spent enforcing the valid subspace confinement. Unfortunately it is not possible to increase Δt too much because this might cause the network to 'overshoot' the valid subspace, with the degree of overshoot increasing at each time step, and hence causing the network to become unstable [3].

In order to overcome this problem I propose the subspace projection network implementation shown in Fig 5.3.

The key advantage of the subspace projection network is that it confines $\mathbf{v}$ to the valid subspace directly by the projection operation (1), rather than by the indirect minimisation of E^{cn}. The nonlinear operation (2) then ensures $\mathbf{v}$ is kept within the unit hypercube, by applying a 'symmetric ramp' threshold function to each of the elements of $\mathbf{v}$. i.e,

$$v_q \quad \rightarrow \quad g(v_q)$$

[3] This is similar to the phenomena of 'stiffness' in dynamical systems

$$\text{where } g(v_q) \;=\; \begin{cases} 1 & \text{if} \quad v_q > 1 \\ v_q & \text{if} \quad 0 \leq v_q \leq 1 \\ 0 & \text{if} \quad v_q < 0 \end{cases}$$

The proof that repeated action of operation (1) followed by operation (2) will confine $\mathbf{v}$ to both the valid subspace and unit hypercube is given in Appendix A.5.

The minimisation of E^{op} is achieved by operation (3). This computes the change in $\mathbf{v}$ as:

$$\Delta\mathbf{v} = \Delta t \frac{d}{dt}(\mathbf{v})$$

where

$$\frac{d}{dt}(\mathbf{v}) = \dot{\mathbf{v}} = \mathbf{T}^{\mathrm{op}}\mathbf{v} + \mathbf{i}^{\mathrm{op}} \tag{5.15}$$

Recalling from (1.35) that,

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

it can be seen that,

$$\begin{aligned} \Delta\mathbf{v} &= -\Delta t \nabla E^{\mathrm{op}} \\ \Rightarrow \Delta\mathbf{v}^T \nabla E^{\mathrm{op}} &\leq 0 \end{aligned}$$

Hence operation (3) will always change $\mathbf{v}$ in a way that decreases E^{op}. In contrast to the continuous Hopfield network, $\Delta\mathbf{v}$ is exclusively ‘concerned’ with the minimization of E^{op}, and consequently computer simulations of the modified network converge in far fewer iterations than equivalent simulations of the Hopfield network. Typically, for 30-city TSPs, simulations of the modified network converge in $\sim$ 500 iterations as opposed to $\sim$ 20000 iterations for the continuous Hopfield network.

The modified network has the further important advantage of having just one parameter, Δt. This compares with at worst five parameters (A, B, C, D and Δt) for the network as formulated by Hopfield and Tank in their 1985 paper [25], and at best two parameters (θ and Δt) for the continuous Hopfield network as formulated with the E^{cn} specified in section 2.4.

5.3 Comparison of Results

The following results are mainly aimed at providing an overall confirmation of the effectiveness of the improvements proposed in this chapter, and are not intended to be a large scale experimental evaluation, which is beyond the scope of this essentially theoretical dissertation. The actual results compare the performance of the three networks listed below for solving Hopfield and Tank's 30 city TSP [25].

(i) The Subspace Projection Network with version 2 MGNC.

Type of Network	Minimum Length	Average Length	Maximum Length	Standard Deviation
Subspace Projection Network with version 2 MGNC	4.298	4.3971	4.500	0.077
Subspace Projection Network without MGNC	4.350	4.763	5.267	0.230
MFA network with normalization	4.277	4.543	6.521	0.437

Table 5.2: The Minimum, Mean, Maximum and Standard deviation of the results shown in Fig 5.4. The optimum tour length, according to [25], is 4.268

(ii) The Subspace Projection Network with no effective MGNC apart from the minimum necessary to ensure convergence to a hypercube corner (see section 5.1.4).

(iii) The asynchronous discrete MFA network with normalization.

The implementation details of the Subspace Projection Network are given in Appendix B, and the implementation of the asynchronous discrete MFA network follows that of C.Peterson in [43]. Figure 5.4 shows the results obtained from 100 runs of each network, plotted as frequency histograms of the final solution tour length.

The three dotted lines on each plot show respectively the minimum, average and maximum tour length obtained out of the 100 runs. These values are enumerated in Table 5.3.

It can clearly be seen that the use of MGNC version 2 greatly improves the performance of the Subspace Projection network. Since this network is just an efficient computer simulation of the continuous Hopfield network, this also confirms the effectiveness of MGNC for improving the performance of the continuous Hopfield network. If we are only interested in the value of the minimum tour length, then the MFA network with normalization achieves the best results. However, this is at the expense of a higher mean value and much higher variance than is achieved using the Subspace Projection network with version 2 MGNC. It should be noted that the relative performance of the Subspace Projection networks versus the MFA network may change for problems other than the Euclidean TSP. Nevertheless, given the degree of improvement shown in Fig 5.4, and its theoretical basis, the beneficial effect of MGNC is unlikely to be affected.

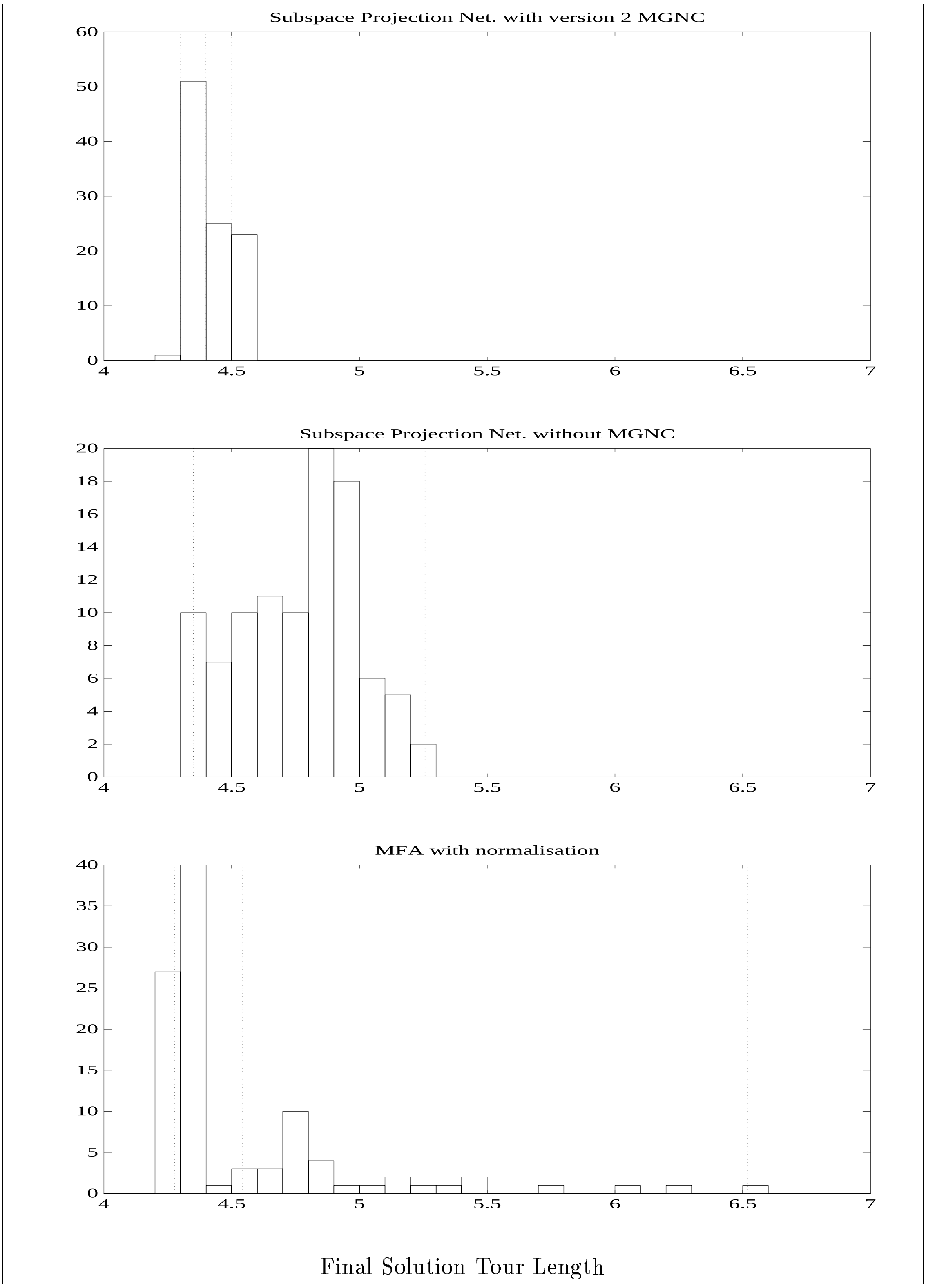


Figure 5.4: Frequency Histogram Plots of Final Solution Tour length for Hopfield & Tank's 30 city TSP, obtained from a 100 runs of the Subspace Projection network with and without MGNC, and the MFA network with normalization.

Chapter 6

The Viterbi Algorithm and EM Networks

So far in this thesis EM networks have been used to solve problems which, although benchmark examples of hard combinatorial optimization problems, are mainly of theoretical interest. The aim of this chapter is to show how these networks can be used to solve problems of practical significance. Specifically it will be shown that EM networks can implement the Viterbi Algorithm, which is widely used to compute the recognition probabilities for Hidden Markov Model (HMM) based Speech Recognition systems [36].

The Viterbi algorithm itself is an optimal serial algorithm for calculating the most probable state sequences of an HMMs. The main idea behind mapping it onto EM networks is to treat it as a form of combinatorial optimization over the space of state sequences. This combinatorial optimization problem can then be mapped onto the network in a way analogous to the other combinatorial optimization problems that have been dealt with in this thesis. C.Chiu et al [12, 13] have developed a way of mapping dynamic programming problems onto the continuous Hopfield network, and since the Viterbi algorithm is a form of dynamic programming, their mapping could also be potentially used. In practice, however, the expressions developed in this chapter are quite different to the ones that would be obtained if this were done. One reason for the difference is that the Viterbi algorithm and standard dynamic programming, although similar, are not identical because of the presence of output probabilities in HMMs. However, the main difference is that C.Chiu et al are handicapped by the fact they are not aware of the valid subspace analysis or the Kronecker product based mathematical framework. For these reasons, their E^{cn} term is incorrectly formulated and does not confine $\mathbf{v}$ to the valid subspace. It also means that they have various weighting parameters for which appropriate values have to be found, and upon the choice of which the performance of their network is crucially dependent.

The following description of the implementation of the Viterbi algorithm by EM networks is divided into three sections. The first introduces the key concepts and notation conventions relevant to the Hidden Markov Model and the Viterbi algorithm. The second section develops and justifies the expressions required to map the Viterbi algorithm onto

EM networks, and finally the last section presents and discusses the experimental results.

6.1 HMMs and the Viterbi Algorithm: Concepts and Notation

Let t be a discrete time variable where $t \in \{1, \ldots, n\}$.

Let $q_1, q_2, \ldots, q_m$ be the states of an HMM with m states.

Let $\mathbf{A}$ be the $m \times m$ matrix of interstate transition probabilities:
$A_{ij} = \Pr(q_j \text{ at } t{+}1 | q_i \text{ at } t)$

Let $\omega_1, \omega_2 \ldots, \omega_k$ be a set of k output symbols.

Let $\mathbf{B}$ be the $m \times k$ output symbol probability matrix: $B_{ij} = \Pr[w_j \text{ at } t | q_i \text{ at } t]$.
For continuous density HMMs, $\mathbf{B}$ is replaced by a set of means and covariance matrices which are associated with each state of the HMM.

Let $\mathbf{O}$ be a n-element vector which denotes a sequence of n output symbols:
$O_t = k$ if the output symbol at time t is ω_k.
In the case of continuous density HMMs, instead of O_t being a discrete symbol number, it is a continuous valued vector.

Let $\mathbf{p}$ be a n-element vector which denotes a sequence of n states:
$p_t = i$ if at time t the HMM is in state q_i.

In terms of the above notation, the Viterbi algorithm reduces to finding a sequence of states $\mathbf{p}$ which maximizes the joint likelihood that the HMM specified by $\mathbf{A}$ and $\mathbf{B}$ generated $\mathbf{O}$. Let $\mathcal{L}(\mathbf{O}, \mathbf{p}|\mathbf{A}, \mathbf{B})$ denote this joint likelihood. Using this expression, the Viterbi algorithm reduces to the following combinatorial optimization problem:

$$\arg\max_{\mathbf{p}} \mathcal{L}(\mathbf{O}, \mathbf{p}|\mathbf{A}, \mathbf{B})$$

This probability can be decomposed into two parts:

$$\mathcal{L}(\mathbf{O}, \mathbf{p}|\mathbf{A}, \mathbf{B}) = \mathcal{L}(\mathbf{O}|\mathbf{A}, \mathbf{B}, \mathbf{p}) \mathcal{L}(\mathbf{p}|\mathbf{A}, \mathbf{B}) \tag{6.1}$$

and, given the definitions of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{O}$ and $\mathbf{p}$ specified above, $\mathcal{L}(\mathbf{O}|\mathbf{A}, \mathbf{B}, \mathbf{p})$ and $\mathcal{L}(\mathbf{p}|\mathbf{A}, \mathbf{B})$ can be further decomposed and expressed as logs.

$$\begin{aligned}
\mathcal{L}(\mathbf{O}|\mathbf{A}, \mathbf{B}, \mathbf{p}) &= \prod_{t=1}^{m} B_{p_t O_t} && (6.2) \\
\Rightarrow \log(\mathcal{L}(\mathbf{O}|\mathbf{A}, \mathbf{B}, \mathbf{p})) &= \sum_{t=1}^{m} \log\left(B_{p_t O_t}\right) && (6.3) \\
\mathcal{L}(\mathbf{p}|\mathbf{A}, \mathbf{B}) &= \prod_{t=1}^{m-1} A_{p_t p_{t+1}} && (6.4) \\
\Rightarrow \log(\mathcal{L}(\mathbf{p}|\mathbf{A}, \mathbf{B})) &= \sum_{t=1}^{m-1} \log\left(A_{p_t p_{t+1}}\right) && (6.5)
\end{aligned}$$

6.2 Mapping the Viterbi Algorithm onto EM Networks

6.2.1 Representing Valid Solutions

From the definition of the n-element vector $\mathbf{p}$ given above, we can see that the valid solutions of the Viterbi algorithm implementation for a HMM with m states, and recognising an output symbol string of n symbols, are identical to the valid solutions of a GPP with n nodes and m partitions. However, unlike the GPP, where each partition has to have an equal number of nodes, there is no restriction on the number of times an HMM state can be visited. Thus $\mathbf{v}(\mathbf{p})$ and $\mathbf{V}(\mathbf{p})$ are given by (1.3), (1.4) and (1.5)

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \boldsymbol{\delta}^{p_1}(m) \\ \boldsymbol{\delta}^{p_2}(m) \\ \dots \\ \boldsymbol{\delta}^{p_n}(m) \end{bmatrix} \qquad \mathbf{V}(\mathbf{p}) = \begin{bmatrix} \boldsymbol{\delta}^{p_1}(m)^T \\ \boldsymbol{\delta}^{p_2}(m)^T \\ \dots \\ \boldsymbol{\delta}^{p_n}(m)^T \end{bmatrix}$$
$$\text{where} \quad [\boldsymbol{\delta}^{p}(m)]_i = \begin{cases} 1 & \text{if} \quad i = p \\ 0 & \text{if} \quad i \neq p \end{cases} \quad p \in \{1, \dots, m\}$$

For a 4 state HMM, with the sequence of states $q_1q_2q_3q_3q_4q_4$, the vectors $\mathbf{p}$ and $\mathbf{v}(\mathbf{p})$, and the matrix $\mathbf{V}(\mathbf{p})$ would expand as follows,

$$\begin{aligned} \mathbf{p}^T &= [1, 2, 3, 3, 4, 4] \\ \mathbf{v}(\mathbf{p})^T &= \begin{matrix} [1\,0\,0\,0 & 0\,1\,0\,0 & 0\,0\,1\,0 & 0\,0\,1\,0 & 0\,0\,0\,1 & 0\,0\,0\,1] \\ [\boldsymbol{\delta}^{p_1 T} & \boldsymbol{\delta}^{p_2 T} & \boldsymbol{\delta}^{p_3 T} & \boldsymbol{\delta}^{p_4 T} & \boldsymbol{\delta}^{p_5 T} & \boldsymbol{\delta}^{p_6 T}] \end{matrix} \\ \mathbf{V}(\mathbf{p}) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

6.2.2 Expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$

The expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$ that will now be developed, are based on the valid subspace analysis of chapter 3. However, unlike the cases considered in that chapter, only the row constraints on $\mathbf{V}(\mathbf{p})$ need to be enforced, since a particular HMM state many be visited any number of times. Thus the matrix form of the valid subspace equation (2.37) becomes:

$$\begin{aligned} \mathbf{V} &= \mathbf{V}\mathbf{R}^m + \tfrac{1}{m}\mathbf{O}^{nm} \qquad (6.6) \\ \Rightarrow \mathbf{V} &= \mathbf{I}^n\mathbf{V}\mathbf{R}^m + \tfrac{1}{m}\mathbf{O}^{nm} \end{aligned}$$

Recalling from (2.29) that the vector of row sums ($\mathbf{r}$) is given by

$$\mathbf{r} = \mathbf{V}\mathbf{o}^m$$

it can be seen that if $\mathbf{V}$ satisfies (6.6) then $\mathbf{r} = \frac{1}{m}\mathbf{O}^{nm}\mathbf{o}^m = \mathbf{o}^m$. Hence if $\mathbf{V}$ corresponds to a hypercube corner, then only one element per row of $\mathbf{V}$ can have a value of 1, which is the exact condition for $\mathbf{V}$ to represent a valid solution of the Viterbi implementation.

Applying the vec() operator and using (2.14), this can be transformed into a vector equation in terms of the zerosum subspace projection matrix $\mathbf{T}^{\mathrm{zs}}$:

$$\begin{aligned} \mathbf{v} &= \mathbf{T}^{\mathrm{zs}}\mathbf{v} + \mathbf{s} \\ \text{where} \quad \mathbf{s} &= \frac{1}{m}(\mathbf{o}^n \otimes \mathbf{o}^m) && (6.7) \\ \mathbf{T}^{\mathrm{zs}} &= (\mathbf{I}^n \otimes \mathbf{R}^m) && (6.8) \end{aligned}$$

In section 2.4 robust expressions were developed for E^{cn}, $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$, which ensured that an EM network would correctly confine $\mathbf{v}$ to the valid subspace. If the above values for $\mathbf{T}^{\mathrm{zs}}$ and $\mathbf{s}$ are substituted into these expressions, we get the following equations:

$$\begin{aligned} \mathbf{T}^{\mathrm{cn}} &= \theta(\mathbf{T}^{\mathrm{zs}} - \mathbf{I}^{nm}) \\ &= \theta\left((\mathbf{I}^n \otimes \mathbf{R}^m) - (\mathbf{I}^n \otimes \mathbf{I}^m)\right) \\ &= \theta\left((\mathbf{I}^n \otimes (\mathbf{I}^m - \frac{1}{m}\mathbf{O}^m)) - (\mathbf{I}^n \otimes \mathbf{I}^m)\right) \\ \Rightarrow \mathbf{T}^{\mathrm{cn}} &= -\theta\frac{1}{m}(\mathbf{I}^n \otimes \mathbf{O}^m) && (6.9) \\ \mathbf{i}^{\mathrm{cn}} &= \theta\frac{1}{m}\mathbf{o}^{nm} && (6.10) \end{aligned}$$

6.2.3 Expressions for E^{op}, $\mathbf{T}^{\mathrm{op}}$ and $\mathbf{i}^{\mathrm{op}}$

For E^{op} to function correctly as an optimization energy term for the Viterbi implementation, $\mathbf{T}^{\mathrm{op}}$ must be a symmetric matrix and $\mathbf{i}^{\mathrm{op}}$ a vector such that $E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}))$ is a monotonic function of $-\mathcal{L}(\mathbf{O},\mathbf{p}|\mathbf{A},\mathbf{B})$. In other words

$$\text{if} \quad \mathcal{L}(\mathbf{O},\mathbf{p}^1|\mathbf{A},\mathbf{B}) \geq \mathcal{L}(\mathbf{O},\mathbf{p}^2|\mathbf{A},\mathbf{B}) \quad \text{then} \quad E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}^1)) \leq E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}^2))$$

This can be achieved by setting $\mathbf{T}^{\mathrm{op}}$, and $\mathbf{i}^{\mathrm{op}}$ according to:

$$\begin{aligned} \mathbf{T}^{\mathrm{op}} &= \mathbf{T}^{\mathrm{pq}} + \mathbf{T}^{\mathrm{pq}T} \\ \mathbf{i}^{\mathrm{op}} &= \mathrm{vec}(\mathbf{I}^{\mathrm{op}T}) \\ \text{where} \quad \mathbf{T}^{\mathrm{pq}} &= (\mathbf{P} \otimes \mathbf{Q}) && (6.11) \\ \text{and} \quad P_{xy} &= \delta_{x+1,y} \\ Q_{ij} &= \log(A_{ij}) \\ I^{\mathrm{op}}_{ij} &= \log(B_{jO_i}) \end{aligned}$$

The proof that $E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}))$ will have the appropriate properties is as follows:

First using $\mathbf{v} = \mathrm{vec}(\mathbf{V}^T)$, (2.13) and (2.15), E^{op} can be expressed in terms of the trace of matrix products:

$$\begin{aligned}
E^{\mathrm{op}} &= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{op}}\mathbf{v} - \mathbf{i}^{\mathrm{op}T}\mathbf{v} \\
&= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{pq}}\mathbf{v} - \tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{pq}T}\mathbf{v} - \mathbf{i}^{\mathrm{op}T}\mathbf{v} \\
&= -\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{pq}}\mathbf{v} - \tfrac{1}{2}(\mathbf{v}^T\mathbf{T}^{\mathrm{pq}}\mathbf{v})^T - \mathbf{i}^{\mathrm{op}T}\mathbf{v} \\
&= -\mathbf{v}^T\mathbf{T}^{\mathrm{pq}}\mathbf{v} - \mathbf{i}^{\mathrm{op}T}\mathbf{v} \\
&= -\mathrm{vec}(\mathbf{V}^T)^T(\mathbf{P}\otimes\mathbf{Q})\mathrm{vec}(\mathbf{V}^T) - \mathrm{vec}\left(\mathbf{I}^{\mathrm{op}T}\right)\mathrm{vec}(\mathbf{V}^T) \\
\Rightarrow E^{\mathrm{op}} &= -\mathrm{trace}\left(\mathbf{V}\mathbf{Q}\mathbf{V}^T\mathbf{P}^T\right) - \mathrm{trace}\left(\mathbf{I}^{\mathrm{op}}\mathbf{V}^T\right) \qquad (6.12)
\end{aligned}$$

Let $\mathbf{V} = \mathbf{V}(\mathbf{p})$ and $\mathbf{E} = \mathbf{V}\mathbf{Q}\mathbf{V}^T$.
According to (1.6) $[\mathbf{V}(\mathbf{p})]_{ij} = [\boldsymbol{\delta}^{p_i}(m)]_j$, hence from (2.2),

$$\begin{aligned}
E_{ij} &= \boldsymbol{\delta}^{p_i T}\mathbf{Q}\boldsymbol{\delta}^{p_j} \\
\Rightarrow E_{ij} &= Q_{p_i p_j}
\end{aligned}$$

But $\mathrm{trace}\left(\mathbf{E}\mathbf{P}^T\right) = \sum_{k=1}^{n}\sum_{l=1}^{n} E_{kl}P_{kl}$, thus,

$$\begin{aligned}
\mathrm{trace}\left(\mathbf{V}\mathbf{Q}\mathbf{V}^T\mathbf{P}^T\right) &= \mathrm{trace}\left(\mathbf{E}\mathbf{P}^T\right) \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} E_{kl}P_{kl} \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} Q_{p_k p_l}P_{kl}
\end{aligned}$$

If we now substitute the values for $\mathbf{P}$ and $\mathbf{Q}$ given in (6.11), it follows that if $\mathbf{V} = \mathbf{V}(\mathbf{p})$ then,

$$\begin{aligned}
\mathrm{trace}\left(\mathbf{V}\mathbf{Q}\mathbf{V}^T\mathbf{P}^T\right) &= \sum_{k=1}^{n}\sum_{l=1}^{n} \log\left(A_{p_k p_l}\right)\delta_{k+1,l} \\
\Rightarrow \mathrm{trace}\left(\mathbf{V}\mathbf{Q}\mathbf{V}^T\mathbf{P}^T\right) &= \sum_{k=2}^{n} \log\left(A_{p_k p_{k+1}}\right) \qquad (6.13)
\end{aligned}$$

Substituting for $\mathbf{I}^{\mathrm{op}}$ with (6.11), and using (1.6), we can express the $\mathrm{trace}\left(\mathbf{I}^{\mathrm{op}}\mathbf{V}^T\right)$ term as a summation,

$$\begin{aligned}
\mathrm{trace}\left(\mathbf{I}^{\mathrm{op}}\mathbf{V}^T\right) &= \sum_{k=1}^{n}\sum_{l=1}^{n} \log\left(B_{lO_k}\right)[\boldsymbol{\delta}^{p_k}(m)]_l \\
\Rightarrow \mathrm{trace}\left(\mathbf{I}^{\mathrm{op}}\mathbf{V}^T\right) &= \sum_{k=1}^{n} \log\left(B_{p_k O_k}\right) \qquad (6.14)
\end{aligned}$$

If the expressions for trace $\left(\mathbf{V}\mathbf{Q}\mathbf{V}^T\mathbf{P}^T\right)$ and trace $\left(\mathbf{I}^{\mathrm{op}}\mathbf{V}^T\right)$ given in (6.13) and (6.14) are substituted in (6.12), we can see from (6.3) and (6.5) that,

$$\begin{aligned}
E^{\mathrm{op}}(\mathbf{v}(\mathbf{p})) &= -\sum_{k=2}^{n}\log\left(A_{p_k p_{k+1}}\right) - \sum_{k=1}^{n}\log\left(B_{p_k o_k}\right) \\
&= -\log(\mathcal{L}(\mathbf{O}|\mathbf{A},\mathbf{B},\mathbf{p})) - \log(\mathcal{L}(\mathbf{p}|\mathbf{A},\mathbf{B})) \\
\Rightarrow E^{\mathrm{op}}(\mathbf{v}(\mathbf{p})) &= -\log(\mathcal{L}(\mathbf{O},\mathbf{p}|\mathbf{A},\mathbf{B})) \qquad (6.15)
\end{aligned}$$

Hence $E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}))$ satisfies the condition:

$$\text{if} \quad \mathcal{L}(\mathbf{O},\mathbf{p}^1|\mathbf{A},\mathbf{B}) \geq \mathcal{L}(\mathbf{O},\mathbf{p}^2|\mathbf{A},\mathbf{B}) \quad \text{then} \quad E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}^1)) \leq E^{\mathrm{op}}(\mathbf{v}(\mathbf{p}^2))$$

In addition, it can be seen that the mapping used neatly splits up the joint likelihood function into an output sequence part, $\mathcal{L}(\mathbf{O}|\mathbf{A},\mathbf{B},\mathbf{p})$, which is handled by the input bias term $\mathbf{i}^{\mathrm{op}}$, and a state sequence part $\mathcal{L}(\mathbf{p}|\mathbf{A},\mathbf{B})$ which is handled by the quadratic term $\mathbf{v}(\mathbf{p})\mathbf{T}^{\mathrm{op}}\mathbf{v}(\mathbf{p})$. Since the input bias term is an external input to the network, it is highly appropriate that it carries all the information about the external output sequence which the HMM is trying to recognise.

6.3 Experimental Results and Discussion

6.3.1 An artificial small scale experiment

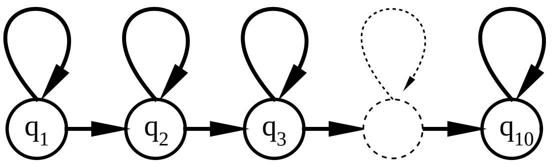
The first experiment is of small scale, designed to confirm the theory presented above and highlight some of the features of using EM networks. The actual experiment consists of finding the optimum state sequence for a 10 state left to right HMM (shown in Fig 6.1) to generate a sequence of 20 output symbols. The experimental results were generated from computer simulation of a Subspace Projection network (see section 5.2), using version 1 MGNC to enforce convergence to a hypercube corner.

As a simplification, the total number of possible output symbols was restricted to 20 and the output sequence was assumed to be $\omega_1\omega_2\cdots\omega_{20}$: hence $\mathbf{O} = [1, 2, \ldots, 20]^T$. The transition matrix $\mathbf{A}$ is shown in Fig 6.1, and the output probability matrix $\mathbf{B}$ was generated randomly subject to the constraint that the sum of each row was one (i.e the total output probability per state was one). In addition, to ensure that the HMM started in state q_1 and ended in state q_{10}, all the elements of the first and last rows and columns of $\mathbf{B}$ were set to zero, except for the element B_{11} and B_{mk} which were set to one. Fig 6.3.1 shows the evolution of the network output from an initial random state to the final solution.

The output shown in Figure 6.3.1 clearly shows the network achieving a valid solution. Further the state sequence represented by this solution,

$$q_1q_2q_3q_4q_5q_6q_6q_6q_6q_6q_7q_7q_8q_8q_9q_9q_9q_9q_9q_{10}$$

is in fact the optimum, since a standard dynamic programming based Viterbi algorithm gives exactly the same state sequence, for the $\mathbf{A}$ and $\mathbf{B}$ matrices used in the simulation.



$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & \cdots & 0 \\ 0 & 0 & 0.5 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0.5 \end{bmatrix}$$

Figure 6.1: Diagram of left to right 10 state HMM with a 10×10 state transition matrix $\mathbf{A}$

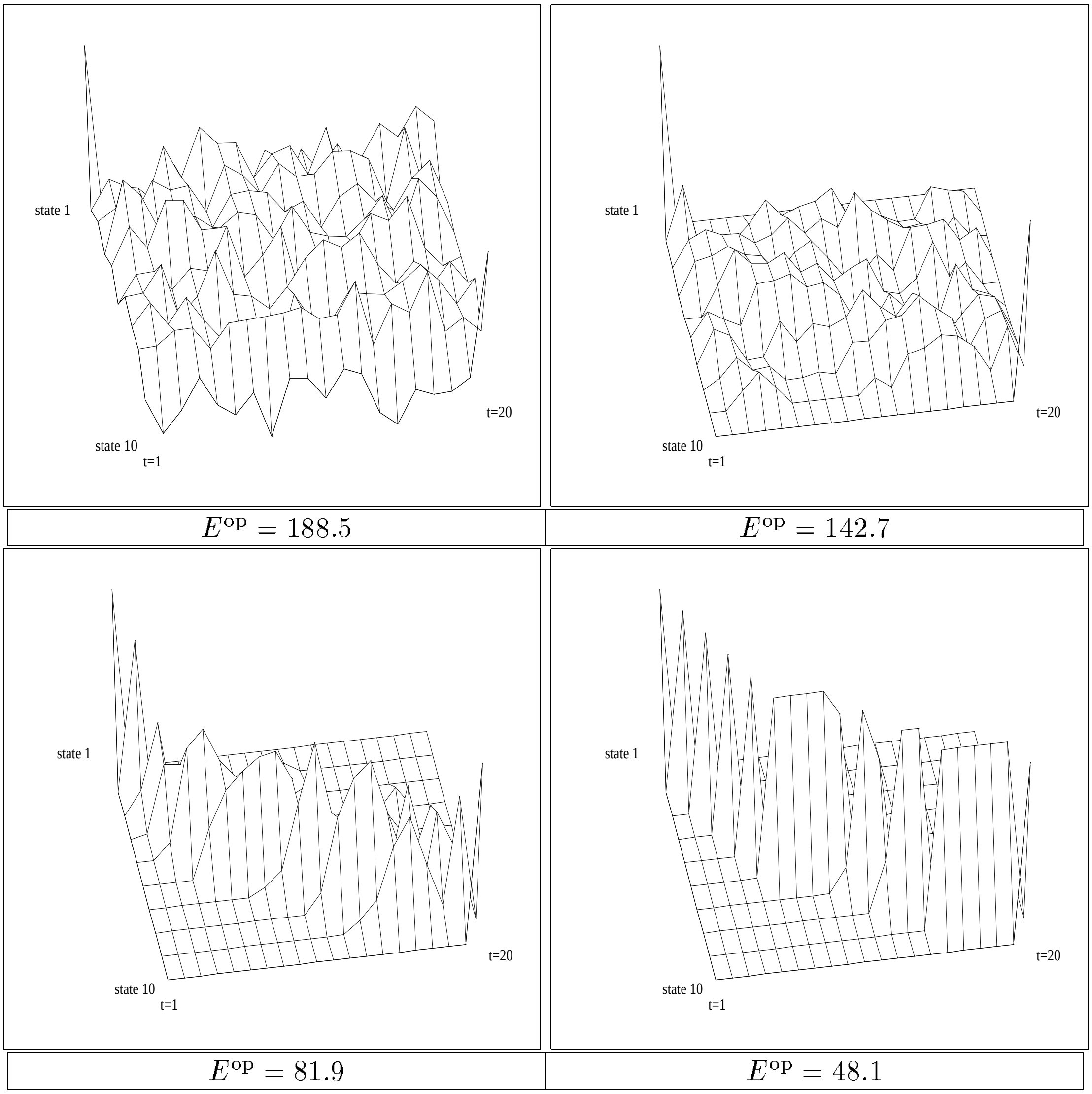
This experiment provides a useful illustration of one of the key features intrinsic to using EM networks. This is the ability of the network to keep active more than one solution. As an illustration, in the top right plot it can be seen that two partial solutions are active simultaneously, of which one eventually dominates.

6.3.2 Preliminary results of a large scale Speech Recognition Experiment

Since the aim of this chapter is to show that EM networks can be to applied to problems of practical significance, it is desirable to test the performance of the Viterbi implementation on a realistic speech recognition experiment. Unfortunately, more research is necessary to iron out the difficulties in handling the log of the zero probabilities in the state transition matrix (see discussion). Consequently, the results that will be presented are of a preliminary nature.

The actual experiment consists of using a set of tri-state HMM phoneme recognition models to recognise the phonemes in the utterance ‘The drunkard is a social outcast’. This sentence and the sentences used to train the HMMs are drawn from the DARPA TIMIT database [35]. The output probabilities for each state in the tri-state state models are computed using continuous density single mixture gaussians, and the tri-state models are connected to each other using a looped phoneme model. Thus, the final state of one phoneme model is connected to the first state of another model with a transition probability given by the bigram probability of the two phonemes. Following Kai-Fu Lee [33, 34], 48 tri-state phoneme models were used, producing an overall HMM with 144 states.

Normally, when such a pre-trained looped phoneme model is being used to recognise a sentence, the Viterbi algorithm is used to decide which frames of the utterance correspond to which states of the underlying 144 state HMM. Once this is done, the utterance can be



$E^{\mathrm{op}} = 188.5$

$E^{\mathrm{op}} = 142.7$

$E^{\mathrm{op}} = 81.9$

$E^{\mathrm{op}} = 48.1$

Figure 6.2: 3D Mesh Plots showing the evolution of **V** in a simulation of the Subspace Projection network implementing the Viterbi algorithm for a 10 state HMM and sequence of 20 output symbols. The final solution shown in the bottom right corner, is identical to the solution obtained by dynamic programming, i.e. it is the global optimum. (N.B. The vertical axis scale changes for each mesh plot.)

phonetically labelled by matching these states to their corresponding tri-state phoneme models. The bottom part of Figs 6.3 and 6.4 shows the result of this Viterbi match for the pre-trained models used in the experiment. The Viterbi algorithm for the 144 state HMM was then mapped onto a Subspace Projection Network using the expressions derived in this chapter. Fig 6.3 shows the network output **V** at an intermediate stage (iteration 100) and close to the final solution (iteration 1000). The Y-axis is the HMM state, broken down into tri-state phoneme models, whilst the X-axis represents the frames of the utterance. Each frame consists of the 12 order LPC reflection and delta LPC reflection coefficients for a 10ms section of the utterance. For the utterance 'The drunkard is a social outcast' there are 243 frames (2.43s of speech), thus **V** is a 243×144 matrix. What is actually shown in Fig 6.3 is the transposed version of **V**, where the elements of **V** are represented by a gray-scale (black=1, white=0).

Due to the difficulty of picking out the gray-scale values of each of the 243×144 elements of **V**, Fig 6.4 shows the same information as Fig 6.3, except that the gray-scale corresponding to the sum of the 3 states in each tri-state model is displayed.

6.3.3 Discussion

Comparing the network outputs shown in Fig 6.4 with the actual Viterbi alignment, we can see that the network is gradually evolving **V** so that it resembles something that is quite close to the desired solution. In addition, from Fig 6.3, we can see that the network is correctly going through each 3 state model in the order of the states. Unfortunately, these Figs also show that the final solution is far from the being identical to the Viterbi solution. The main defect seems to be that instead of assigning a large section of frames to one phoneme model, the network breaks up the large sections by assigning small portions of them to other phoneme models. This may be due to a variety of factors, amongst which the handling of log zero is likely to be the main cause . If any of the bigram probabilities or phoneme model HMM transition probabilities are zero, then the connection matrix for the network implementation must have an entry corresponding to log zero. Since minus infinity cannot be used, an arbitrary value has to be employed instead. It turns out that the performance of the network is heavily dependent on this arbitrary value, and so far no systematic way of assigning it has yet been found.

Although these results are of a preliminary experiment, taken together with the results of the previous small scale experiment, they do show that there is significant scope for implementing the Viterbi algorithm with EM networks. Given the many advantages of these networks in terms of inherent parallel structure and possible hardware implementations, they provide a useful indication of the practical potential of EM networks.

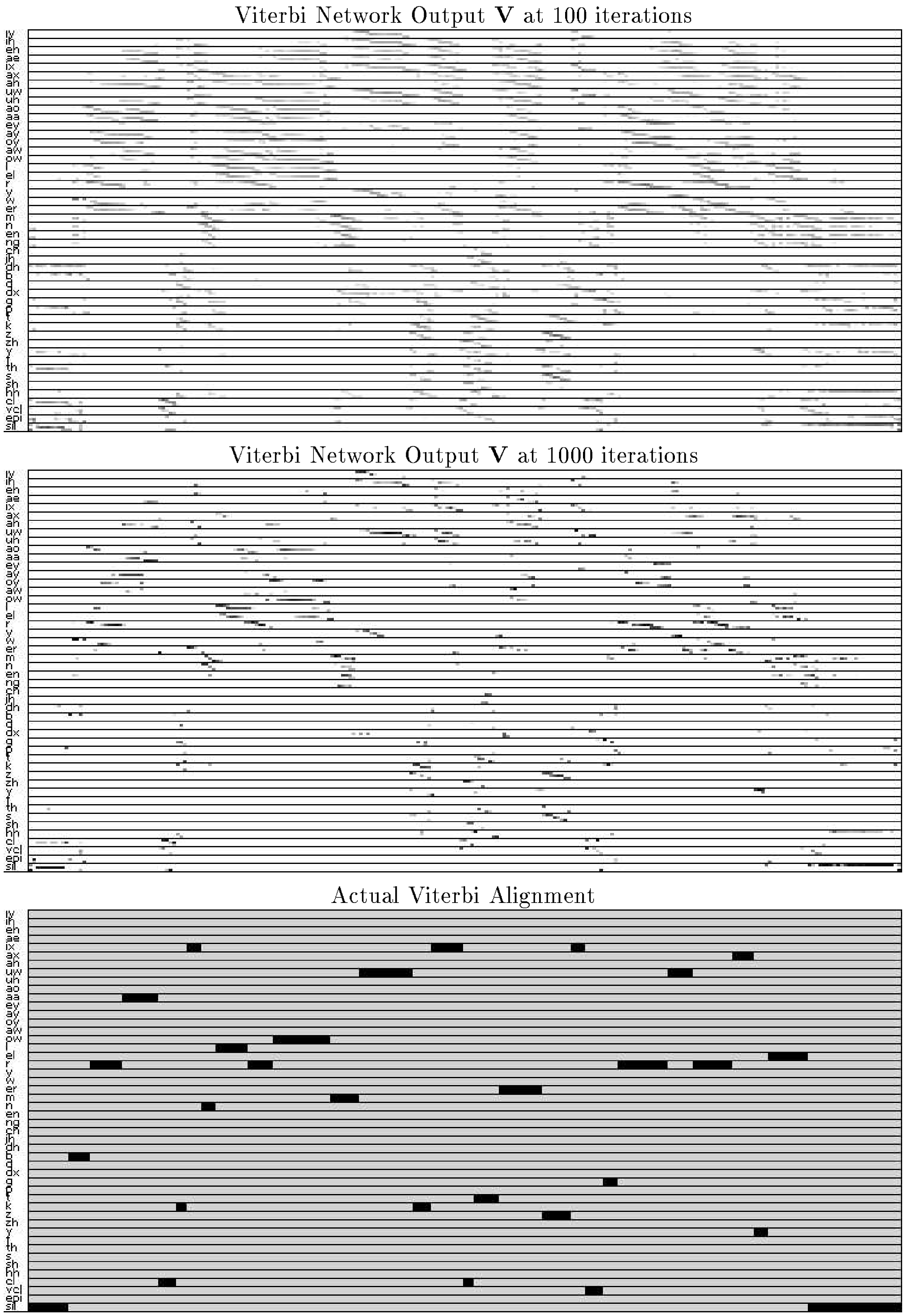
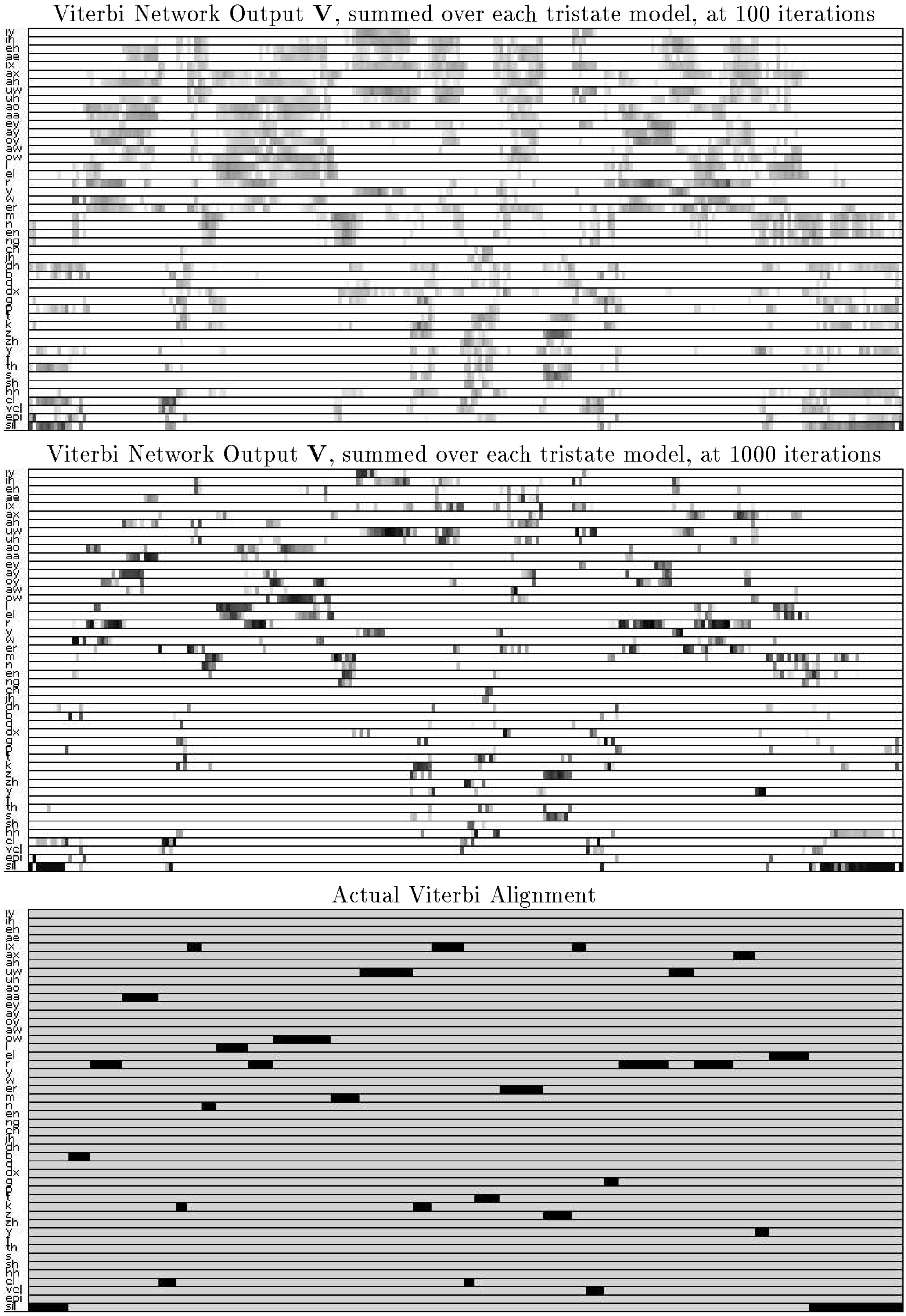


Figure 6.3: The output of a Subspace Projection network implementation of the Viterbi Algorithm, for a 48, tristate HMM, looped phoneme model recognising the utterance 'The drunkard is a social outcast'



Viterbi Network Output **V**, summed over each tristate model, at 100 iterations

Viterbi Network Output **V**, summed over each tristate model, at 1000 iterations

Actual Viterbi Alignment

Figure 6.4: The output summed over each phoneme model, of a Subspace Projection network implementation of the Viterbi Algorithm, for a 48, tristate HMM, looped phoneme model recognising the utterance 'The drunkard is a social outcast'

Chapter 7

Future Lines of Research and Conclusions

7.1 Summary of Thesis

The two main classes of neural network, the Hopfield network and Mean Field Annealing network, that can be used to solve combinatorial optimization problems, were derived and presented using a unified theoretical approach. A new mathematical framework was proposed which provides a much simpler and more compact notation for describing these networks, than previously employed. The unified theoretical approach and new mathematical framework were then used to develop a comprehensive theoretical analysis of the how Hopfield and Mean Field Annealing networks solve combinatorial optimization problems.

In the first part of this analysis, it was shown that all the output vectors of these networks that correspond to valid solutions of a particular problem, lie in a common vector subspace, termed the valid subspace. By formulating constraint energy terms that strictly confined the network output to this subspace, it was demonstrated that the commonly reported problem of convergence to invalid solutions could be corrected.

Predictions were then made of the likely forms of the network output that should correspond to the optimum solution of a particular combinatorial optimization problem. Similar predictions were also made for the likely solution that should be achieved given the dynamics of these networks. By establishing the conditions under which these two solutions agreed, a firm theoretical link between solution quality and network dynamics was developed. Confirmation of this was then achieved in a detailed study of the Euclidean Travelling Salesman problem. Based on this link, and the concept of annealing in Mean Field Annealing networks, the technique of Matrix Graduated Non-Convexity was proposed as a way of significantly improving the average quality of the solutions produced by these networks. To improve the efficiency of computer simulations of the continuous Hopfield network, a network, termed the Subspace Projection network, was described, which confined its output to the valid subspace by direct projection. This network was shown to be functionally equivalent to a continuous Hopfield network, and in computer simulations speed up factors of the order of 100 were obtained with it.

Finally, as a demonstration of the possible practical applications in speech recognition of these networks, an implementation of the Viterbi algorithm was described. In an initial small scale experiment using a 10 state Hidden Markov Model, the Hopfield network implementation of the Viterbi algorithm achieved exactly the same solution as a conventional serial implementation of the algorithm, whilst exhibiting interesting possibilities for the use of its partial solutions. A larger scale experiment was then performed in which a looped phoneme Hidden Markov Model [34] with 144 states was used (via the Viterbi algorithm) to recognise the phonemes in the sentence 'The drunkard is a social outcast'. Although this was a preliminary experiment, the results were promising enough to indicate that these networks had significant practical potential in Hidden Markov Model based speech recognition systems.

7.2 Future Lines of Research

Although the main research objectives of this thesis have been largely achieved, further experimental and theoretical extensions are possible.

The new mathematical framework, valid subspace analysis and MGNC algorithm, allow many problems for which previously poor experimental results have been reported, to be solved with much increased accuracy. Thus, a series of experiments applying the new framework and analysis to such problems would be a desirable research objective. Another area of experimental exploration would be the effect of using MGNC with MFA networks. Although, as shown in section 5.1.4, MFA networks employ a form of 'back door' MGNC to ensure convergence to a hypercube corner, the use of MGNC to improve solution quality has not yet been investigated. In terms of the MGNC algorithm itself, further experimental investigation of the effectiveness of different MGNC schemes would also be useful.

The theoretical extensions are concerned with developing a deeper study of the class of optimization problems for which Hopfield and MFA networks do not achieve good solutions. Naturally these networks solve different problems with a varying degree of accuracy, and it may be possible to classify combinatorial optimization problems, according to how well these networks are able to solve them. A further area of research might then be to see if a more sophisticated version of MGNC can improve the performance of these networks for those problems which at the moment are difficult to solve accurately.

An important priority must also be to further extend the results on implementing the Viterbi algorithm using Hopfield and MFA networks. As yet, these results are of a preliminary nature, and a full exploration of the effects of using different values for the log of zero is necessary to iron out the difficulties associated with it. An interesting outcome of chapter 6 is that the implementation of the Viterbi algorithm on these networks also allows Viterbi training to be mapped onto the same framework, opening up the possibilities of implementing both HMM training and recognition on these networks.

The final area of research, and perhaps the most important in terms of practical possibilities, would be to investigate the implications that the theoretical framework and modifications have for the hardware implementation of these networks. Apart from the Subspace Projection Network, the techniques and modifications proposed in the thesis do not change the structure of either the Hopfield, or MFA network. However, it may be possible to use the analysis of the valid subspace, or the reduction (via Kronecker products) of the order $(nm)^2$ matrix-vector multiplication $(\mathbf{P} \otimes \mathbf{Q})\mathbf{v}$ to an order $nm(n+m)$ matrix-matrix multiplication $\mathbf{PVQ}$ (see Appendix B), to improve the hardware implementation.

7.3 Conclusions

The research into the application of the Hopfield and Mean Field Annealing networks for solving combinatorial optimization problems, has in many ways followed a general trend of neural network research: after much initial promise, extensive research has shown that in practice these networks have failed to fulfil their original expectations. Unfortunately, in the case of the Hopfield network, this failure has extended to the point where some researchers, notable V.Wilson & G.S.Pawley [48] have concluded that these networks cannot be used at all; the solutions produced are either invalid or of an unacceptably poor cost. A more careful examination of much of the research that has led to these conclusions, reveals that a highly experimental approach has been adopted by most researchers, and instead of analyzing the root causes of these problems, they have simply assumed that the networks are fatally flawed. The case of Mean Field Annealing networks, is more promising, mainly because of the use of the more sophisticated Pott's glass model in statistical mechanics, to develop the normalization modification. However, without this modification, the performance of Hopfield and MFA networks are very similar, and even with the normalization modification, the MFA network still suffers from certain problems, such as a high degree of parameter sensitivity.

The original motivation of the thesis was to develop a theoretical analysis of the root causes of the problems of convergence to invalid and poor cost solutions. However, with the new mathematical framework and theoretical analysis that have been developed, it has proved possible not only to explain the root causes of these deficiencies, but also to propose robust methods of correcting them. In addition, the analysis has allowed modifications to be proposed which significantly improve the performance of these networks, over and above that obtained by simply correcting the deficiencies. Importantly, all the new techniques and modifications that have been proposed leave the basic structure of these networks unchanged. Part of the original promise of these networks was their potential for large scale, highly parallel, hardware implementation. However, even these advantages were not enough to compensate for the perceived deficiencies of these networks. With the removal of these deficiencies, and the improvement in performance available with the analysis of this thesis, the potential of hardware implementation once again makes the development of extremely fast, highly parallel hardware for solving combinatorial optimization problems a viable proposition.

With this in mind, it was demonstrated in the last chapter that the Viterbi algorithm could be implemented with a Hopfield or MFA network. Although the results for the large scale phoneme recognition experiment are at a preliminary stage, and an highly efficient serial implementation already exists for the Viterbi algorithm, the promising performance of the network together with its hardware possibilities makes this an application of significant potential.

In conclusion, by explaining and correcting the deficiencies that many researchers have encountered, the theoretical work in this thesis has allowed a much more realistic assessment to be made of the potential of neural networks for solving combinatorial optimization problems. What emerges is a new mathematical and theoretical framework with which neural networks can be reliably and accurately used to solve a wide range of combinatorial optimization problems.

Appendix A

Proofs

A.1 Derivation of Table 2.4.

The first step in the derivation of this Table is to substitute the expressions for $\mathbf{T}^{\mathrm{cn}}$ and $\mathbf{i}^{\mathrm{cn}}$ in Table 2.3 into (1.34), to give the following expressions for E^{cn}:

Author	E^{cn}
S.Abe	$An\frac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{nzo}}\mathbf{v} + An\mathbf{v}^T\mathbf{T}^{\mathrm{o}}\mathbf{v} + 2A\mathbf{o}^{nn^T}\mathbf{v}$
C.Peterson	$-\alpha\mathbf{v}^T\mathbf{T}^{\mathrm{zs}}\mathbf{v} + \alpha(n-2)\frac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{nzo}}\mathbf{v} + \alpha(n-1)\mathbf{v}^T\mathbf{T}^{\mathrm{o}}\mathbf{v} + \alpha\mathbf{o}^{nn^T}\mathbf{v}$
J.Hopfield	$-A\mathbf{v}^T\mathbf{T}^{\mathrm{zs}}\mathbf{v} + A(n-2)\frac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{nzo}}\mathbf{v} + (2A(n-1)+Cn^2)\frac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{o}}\mathbf{v} + C\mathbf{o}^{nn^T}\mathbf{v}$

Table A.1: E^{cn} for the TSP expressed in terms of $\mathbf{T}^{\mathrm{zs}}$, $\mathbf{T}^{\mathrm{nzo}}$, $\mathbf{T}^{\mathrm{o}}$ and $\mathbf{o}^{nn}$

Using the fact that $\mathbf{T}^{\mathrm{zs}}$, $\mathbf{T}^{\mathrm{nzo}}$, and $\mathbf{T}^{\mathrm{o}}$ are projection matrices, it can be seen that:

$$\begin{aligned}
\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{zs}}\mathbf{v} &= \tfrac{1}{2}\left|\mathbf{v}^{\mathrm{zs}}\right|^2 \\
\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{o}}\mathbf{v} - \frac{\gamma}{n}\mathbf{o}^{nn^T}\mathbf{v} + \frac{\gamma^2}{2} &= \tfrac{1}{2}\left|\mathbf{v}^{\mathrm{o}} - \frac{\gamma}{n}\mathbf{o}^{nn}\right|^2 \\
\tfrac{1}{2}\mathbf{v}^T\mathbf{T}^{\mathrm{nzo}}\mathbf{v} &= \tfrac{1}{2}\left|\mathbf{v}^{\mathrm{nzo}}\right|^2
\end{aligned}$$

Table 2.4 is obtained by substituting the RHS of the above expressions for all the occurrences of the LHS in Table A.1.

A.2 Classifying problems by the value of $\mathbf{i}^{\rm opr}$

In this appendix it is shown that combinatorial optimization problems which are solvable by EM networks can be classified according to whether $\mathbf{i}^{\rm opr} = 0$. First consider the expression for $\mathbf{i}^{\rm opr}$ given in (2.58).

$$\begin{aligned} \mathbf{i}^{\rm opr} &= \mathbf{T}^{\rm zs}\mathbf{T}^{\rm op}\mathbf{s} + \mathbf{T}^{\rm zs}\mathbf{i}^{\rm op} \\ &= ((\mathbf{R}^n\mathbf{P}\mathbf{o}^n) \otimes (\mathbf{R}^m\mathbf{Q}\mathbf{a})) + \mathbf{T}^{\rm zs}\mathbf{i}^{\rm op} \end{aligned}$$

Apart from the Viterbi implementation of Chapter 6, $\mathbf{i}^{\rm op} = \mathbf{0}$ for all the combinatorial optimization problems considered in this thesis (i.e. those described in section 1.1). Thus the value of $\mathbf{i}^{\rm opr}$ is dictated by the value of $\mathbf{T}^{\rm zs}\mathbf{T}^{\rm op}\mathbf{s}$. For the problems described in section 1.1, $\mathbf{a} = \frac{1}{m}\mathbf{o}^m$, hence from (2.21):

$$\begin{array}{llll} \text{if} & \mathbf{P}\mathbf{o}^n = \lambda\mathbf{o}^n & \text{then} & \mathbf{R}^n\mathbf{P}\mathbf{o}^n = \mathbf{R}^n\lambda\mathbf{o}^n = \mathbf{0} \\ \text{if} & \mathbf{Q}\mathbf{o}^m = \gamma\mathbf{o}^m & \text{then} & \mathbf{R}^m\mathbf{Q}\mathbf{o}^m = \mathbf{R}^m\lambda\mathbf{o}^m = \mathbf{0} \end{array}$$

Consequently, if $\mathbf{o}^n$ is an eigenvector of $\mathbf{P}$, or $\mathbf{o}^m$ is an eigenvector of $\mathbf{Q}$, then

$$((\mathbf{R}^n\mathbf{P}\mathbf{o}^n) \otimes (\mathbf{R}^m\mathbf{Q}\mathbf{a})) = \mathbf{0}$$

For the TSP and GPP, (see Table 2.6 and section 4.1.2) $\mathbf{o}^m$ is an eigenvector of $\mathbf{Q}$, and hence for these problems $\mathbf{i}^{\rm opr} = \mathbf{0}$. For the Hamilton Path problem, $\mathbf{o}^m$ is not an eigenvector of $\mathbf{Q}$, and for the Graph Matching problem $\mathbf{Q}$ varies according to the problem instance. Thus for these two problems, whether $\mathbf{i}^{\rm opr} = \mathbf{0}$ will depend upon the particular problem instance.

A.3 Derivation of the constant α

The derivation of this constant is based on evaluating $\mathbf{J}$ at $\mathbf{v} = \mathbf{s}$, where $\mathbf{J} = \nabla_{\mathbf{u}}\mathbf{g}(\mathbf{u})$. For the Hopfield network and MFA network without normalization, $[\mathbf{g}(\mathbf{u})]_i$ is a function of u_i only, hence $\mathbf{J}$ will have nonzero elements only on its leading diagonal. Unfortunately, for the MFA network with normalization, $[\mathbf{g}(\mathbf{u})]_i$ is function of several elements of $\mathbf{u}$, resulting in $\mathbf{J}$ having nonzero off diagonal entries. Consequently, in the following analysis, these two cases will be treated separately.

(i) The Hopfield network and MFA network without normalization.

For these problems, from Table 1.1, it can be seen that:

$$\begin{aligned} v_l &= \frac{1}{1+\exp(-u_l)} \\ \Rightarrow J_{kl} &= \frac{\partial}{\partial u_k}\left(\frac{1}{1+\exp(-u_l)}\right) \\ \Rightarrow J_{kl} &= \begin{cases} v_k(1-v_k) & k = l \\ 0 & k \neq l \end{cases} \end{aligned}$$

For all the combinatorial optimization problems considered in this thesis (apart from chapter 6), $\mathbf{s} = \frac{1}{m}\mathbf{o}^{nm}$, hence at $\mathbf{v} = \mathbf{s}$,

$$\mathbf{J} = \frac{m-1}{m^2}\mathbf{I}^{nm}$$

Substituting this into (3.54), we get:

$$\Delta\mathbf{v} = \frac{m-1}{m^2}\mathbf{I}^{nm}\Delta\mathbf{u}$$

Multiplying both sides by $\mathbf{T}^{\mathrm{zs}}$ and using $\mathbf{T}^{\mathrm{zs}}\Delta\mathbf{v} = \Delta\mathbf{v}^{\mathrm{zs}}$ and $\mathbf{T}^{\mathrm{zs}}\Delta\mathbf{u} = \Delta\mathbf{u}^{\mathrm{zs}}$, it follows that:

$$\begin{aligned} \mathbf{T}^{\mathrm{zs}}\Delta\mathbf{v} &= \frac{m-1}{m^2}\mathbf{T}^{\mathrm{zs}}\Delta\mathbf{u} \\ \Rightarrow \Delta\mathbf{u}^{\mathrm{zs}} &= \frac{m^2}{m-1}\Delta\mathbf{v}^{\mathrm{zs}} \end{aligned}$$

But we are linearizing around $\mathbf{v} = \mathbf{s}$ with the assumption that $|\mathbf{v}^{\mathrm{zs}}| \ll |\mathbf{s}|$. Hence we can substitute $\mathbf{v}^{\mathrm{zs}}$ for $\Delta\mathbf{v}^{\mathrm{zs}}$ and consequently $\mathbf{u}^{\mathrm{zs}}$ for $\Delta\mathbf{u}^{\mathrm{zs}}$, giving:

$$\mathbf{u}^{\mathrm{zs}} = \frac{m^2}{m-1}\mathbf{v}^{\mathrm{zs}} \tag{A.1}$$

(ii) The MFA network with normalization.

In this case, from Table 1.1:

$$V_{xi} = \frac{\exp(U_{xi})}{\sum_i \exp(U_{xi})}$$

where $\mathbf{u} = \mathrm{vec}(\mathbf{U}^T)$ and $\mathbf{v} = \mathrm{vec}(\mathbf{V}^T)$.

Let $k = m(x-1)+i$, $l = m(y-1)+j$ and $J_{kl} = J_{xi,yj}$. Thus,

$$\begin{aligned} J_{kl} &= \frac{\partial}{\partial u_k}[\mathbf{g}(\mathbf{u})]_l \\ \Rightarrow J_{xi,yj} &= \frac{\partial}{\partial U_{xi}}\left(\frac{\exp(U_{yj})}{\sum_j \exp(U_{yj})}\right) \\ &= \begin{cases} V_{xi}(1-V_{xi}) & x = y \ \&\ i = j \\ -V_{xi}V_{xj} & x = y \ \&\ i \neq j \\ 0 & x \neq y \end{cases} \end{aligned}$$

If this is evaluated at $\mathbf{v} = \mathbf{s} = \frac{1}{m}\mathbf{o}^{nm}$, we obtain:

$$\begin{aligned} J_{xi,yj} &= \begin{cases} \frac{1}{m} - (\frac{1}{m})^2 & x = y \ \&\ i = j \\ -(\frac{1}{m})^2 & x = y \ \&\ i \neq j \\ 0 & x \neq y \end{cases} \\ \Rightarrow J_{xi,yj} &= \tfrac{1}{m}\delta_{xy}\delta_{ij} - (\tfrac{1}{m})^2\delta_{xy} \end{aligned}$$

Using (2.6), this can be written in terms of Kronecker products:

$$\mathbf{J} = \tfrac{1}{m}(\mathbf{I}^n \otimes \mathbf{I}^m) - (\tfrac{1}{m})^2(\mathbf{I}^n \otimes \mathbf{O}^m)$$

But from (2.19) $\mathbf{R}^m = \mathbf{I}^m - \frac{1}{m}\mathbf{O}^m$, hence,

$$\mathbf{J} = \tfrac{1}{m}(\mathbf{I}^n \otimes \mathbf{R}^m)$$

Substituting this into (3.54), and multiplying both sides by $\mathbf{T}^{\mathrm{zs}}$, we obtain:

$$\begin{aligned} \Delta\mathbf{v} &= \tfrac{1}{m}(\mathbf{I}^n \otimes \mathbf{R}^m)\Delta\mathbf{u} \\ \Rightarrow \Delta\mathbf{v}^{\mathrm{zs}} &= \tfrac{1}{m}\mathbf{T}^{\mathrm{zs}}(\mathbf{I}^n \otimes \mathbf{R}^m)\Delta\mathbf{u} \end{aligned}$$

But,

$$\begin{aligned} \mathbf{T}^{\mathrm{zs}}(\mathbf{I}^n \otimes \mathbf{R}^m) &= (\mathbf{R}^n \otimes \mathbf{R}^m)(\mathbf{I}^n \otimes \mathbf{R}^m) \\ &= (\mathbf{R}^n \otimes \mathbf{R}^m) \end{aligned}$$

Hence

$$\begin{aligned} \Delta\mathbf{v}^{\mathrm{zs}} &= \tfrac{1}{m}(\mathbf{R}^n \otimes \mathbf{R}^m)\Delta\mathbf{u} \\ \Rightarrow \Delta\mathbf{u}^{\mathrm{zs}} &= m\Delta\mathbf{v}^{\mathrm{zs}} \end{aligned}$$

As argued in case (i), if we are linearizing around $\mathbf{v} = \mathbf{s}$ with the assumption that $|\mathbf{v}^{\mathrm{zs}}| \ll |\mathbf{s}|$ then we can assume $\mathbf{v}^{\mathrm{zs}} = \Delta\mathbf{v}^{\mathrm{zs}}$, and $\mathbf{u}^{\mathrm{zs}} = \Delta\mathbf{u}^{\mathrm{zs}}$, hence

$$\mathbf{u}^{\mathrm{zs}} = m\mathbf{v}^{\mathrm{zs}} \tag{A.2}$$

Putting (A.1) and (A.2) together, we obtain the final result:

$$\mathbf{u}^{\mathrm{zs}} = \alpha\mathbf{v}^{\mathrm{zs}}$$

where,

$$\alpha = \begin{cases} \dfrac{m^2}{m-1} & \text{for the continuous Hopfield network} \\ & \text{and the MFA network without normalization} \\ m & \text{for the MFA network with normalization} \end{cases}$$

A.4 Derivation of the form of A and Z that minimizes E^{opr}

The basis of these derivations is the proof of the following theorem

Theorem A.1 *If* $\mathbf{D}$ *is a* $n \times n$ *matrix which satisfies:*

$$\sum_{k=1}^{n} D_{kl} = 1 \qquad \sum_{l=1}^{n} D_{kl} = 1 \tag{A.3}$$

$$D_{kl} \geq 0 \tag{A.4}$$

and $\mathbf{x}$ *and* $\mathbf{y}$ *are* n*-element vectors which satisfy:*

$$x_1 > x_2 > \cdots > x_n \quad \textit{and} \quad y_1 > y_2 > \cdots > y_n \tag{A.5}$$

then the function

$$F = \mathbf{x}^T\mathbf{D}\mathbf{y} = \sum_{k=1}^{n}\sum_{l=1}^{n} x_k y_l D_{kl} \tag{A.6}$$

is maximized when

$$\mathbf{D} = \mathbf{I}^n \tag{A.7}$$

Proof

Firstly, F is a linear cost function, thus the maximization of F subject to the above equality and inequality constraints is a linear programming problem over the D_{kl}'s. Hence the optimum solution must lie at a vertex of the solution region, where the solution region is the region of D_{kl} space which satisfies the above equality and inequality constraints. From the analysis in section 2.2, it can be seen that the equality conditions correspond to vec($\mathbf{D}^T$) lying in the TSP valid subspace. The only vertices that lie in the valid subspace for which $D_{kl} > 0 \ \forall\ k, l$ are the valid solution hypercube corners. Hence the form of $\mathbf{D}$ that maximizes F must be a permutation matrix: $\mathbf{D} = \mathbf{V}(\mathbf{p})$.

From (1.5),

$$\begin{aligned} F_{\mathbf{D}=\mathbf{V}(\mathbf{p})} &= \mathbf{x}^T\mathbf{V}(\mathbf{p})\mathbf{y} \\ &= \sum_{k=1}^{n}\sum_{l=1}^{n} x_k y_l \delta_{p_k l} \\ &= \sum_{k=1}^{n} x_k y_{p_k} \end{aligned}$$

Let $\mathbf{p}'$ be formed from $\mathbf{p}$ by swapping the last element of $\mathbf{p}$ with the element that has a value of n. Thus,

$$p'_k = \begin{cases} n & k = n \\ p_n & k = q_n \\ p_k & \text{otherwise} \end{cases} \tag{A.8}$$

where $q_n = k$ if $p_k = n$

Hence $$F_{\mathbf{D}=\mathbf{V}(\mathbf{p}')} = \sum_{k=1}^{n} x_k y_{p'_k} \tag{A.9}$$

Thus by substituting for p'_k it can be seen that,

$$\begin{aligned} F_{\mathbf{D}=\mathbf{V}(\mathbf{p}')} - F_{\mathbf{D}=\mathbf{V}(\mathbf{p})} &= x_n y_n + x_{qn} y_{pn} - x_n y_{pn} - x_{qn} y_n \\ &= x_n(y_n - y_{pn}) - x_{qn}(y_n - y_{pn}) \\ &= (x_{qn} - x_n)(y_{pn} - y_n) \end{aligned}$$

From (A.5), $x_{qn} \geq x_n$ and $y_{pn} \geq y_n$. Hence $F' \geq F$, which means that by swapping two elements of $\mathbf{p}$ so that $p_n = n$, F will be increased. This process can be repeated so that $p_{n-1} = n-1$, and using a similar analysis to the above, it can be shown that F will increase again. Thus by induction, F will be maximized when,

$$p_k = k \quad k = 1 \cdots n$$

This corresponds to $\mathbf{V}(\mathbf{p}) = \mathbf{I}^n$, which proves theorem A.1.

A.4.1 Form of A

For $k, l \in \{1, 2, \ldots, n-1\}$ let $\mathbf{D}$ be a $(n-1) \times (n-1)$ matrix defined by,

$$D_{kl} = A^2_{k+1,l+1} \tag{A.10}$$

and let $\mathbf{x}$ and $\mathbf{y}$ be $(n-1)$-element vectors given by:

$$x_k = \lambda_{k+1} \qquad y_l = \gamma_{l+1}$$

Thus (3.30) becomes (assuming $B_{kl} = 0$):

$$\begin{aligned} E^{\text{opr}} &= -\tfrac{1}{2}\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\gamma_l A_{kl}^2 \\ &= -\tfrac{1}{2}\sum_{k=1}^{n-1}\sum_{l=1}^{n-1} x_k y_l D_{kl} \\ &= -\tfrac{1}{2}F \end{aligned}$$

From (3.41), (3.42) and (A.10), it can be seen that $\mathbf{D}$ satisfies the equality and inequality conditions of theorem A.1, while $\mathbf{x}$ and $\mathbf{y}$ satisfy (A.5). Hence, by theorem A.1, to minimize E^{opr} it follows that $\mathbf{D} = \mathbf{I}^{n-1}$. From (3.26) if $\mathbf{v}$ lies in the valid subspace where $\mathbf{s} = \frac{1}{n}\mathbf{o}^{nn}$ then $A_{11} = 1$ and $A_{1k} = A_{l1} = 0$ for $k, l > 2$. Thus, to minimize E^{opr},

$$A_{kl}^2 = \delta_{kl}$$

A.4.2 Form of Z

From (4.11) and (4.12), the complex matrix $\mathbf{Z}$ is related to $\mathbf{A}$ by:

$$|Z_{kl}|^2 = \begin{cases} A_{kl}^2 & l = 1 \text{ or } l = \frac{n+2}{2} \\ A_{kl}^2 + A_{k\tilde{l}}^2 & 1 < l < \frac{n+2}{2} \end{cases} \qquad \text{where } \tilde{l} = n - l + 2$$

It is used in analysis of the Travelling Salesman problem, to handle the two-fold degeneracy of the eigenvalues of $\mathbf{T}^{\text{opr}}$ (if $1 < l < \frac{n+2}{2}$ then $\gamma_l = \gamma_{\tilde{l}}$). Let $\mathbf{p}$ be a permutation which re-orders the γ_l's so that:

$$\text{if } k < l \text{ then } \gamma_{p_k} > \gamma_{p_l}$$

From (4.9) γ_l is given by $\gamma_l = 2\cos\left(\frac{2\pi}{n}(l-1)\right)$, hence it can be seen that a possible definition of $\mathbf{p}$ is:

$$p_k = \begin{cases} \frac{k+1}{2} & k \text{ is odd} \\ n + 1 - \frac{k}{2} & k \text{ is even} \end{cases} \tag{A.11}$$

Let $\hat{\gamma}_k = \gamma_{p_k}$ and $\hat{A}_{kl} = A_{kp_l}$. Hence, assuming $B_{kl} = 0$, from (3.30) we can see that:

$$\begin{aligned} E^{\text{opr}} &= -\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\gamma_l A_{kl}^2 \\ &= -\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\gamma_{p_l} A_{kp_l}^2 \\ &= -\sum_{k=2}^{n}\sum_{l=2}^{m}\lambda_k\hat{\gamma}_l \hat{A}_{kl}^2 \end{aligned}$$

If $a < b$ then $\lambda_a > \lambda_b$ and $\hat{\gamma}_a \geq \hat{\gamma}_b$ hence from the analysis in the previous subsection, it can be seen that E^{opr} is minimized when,

$$\hat{A}^2_{kl} = \delta_{kl}$$

However there is a slight complication caused by the fact that $\hat{\gamma}_{2l} = \hat{\gamma}_{2l+1}$. Hence columns $2l$ and $2l+1$ of $\hat{\mathbf{A}}$ can be swapped without changing E^{opr}. Thus the actual set of $\mathbf{A}$'s that minimize E^{opr} is given by all $\mathbf{A}$'s that can be formed by performing any number of swaps of columns $2l$ and $2l+1$ of $\mathbf{I}^n$, where $1 \leq l < \frac{n}{2}$.

From (A.11), $l = p_{2l-1}$, and $\tilde{l} = p_{2l-2}$. Hence Z^2_{kl} can be expressed as follows:

$$|Z_{kl}|^2 = \begin{cases} \hat{A}^2_{k1} & l = 1 \\ \hat{A}^2_{k,2l-2} + \hat{A}^2_{k,2l-1} & 1 < l < \frac{n+2}{2} \\ \hat{A}^2_{kn} & l = \frac{n+2}{2} \end{cases}$$

We can see that swapping columns $2l-2$ and $2l-1$ of $\hat{\mathbf{A}}$ leaves Z^2_{kl} unchanged, since $Z^2_{kl} = A^2_{k,2l-2} + A^2_{k,2l-1}$. Hence we can assume that $\hat{\mathbf{A}} = \mathbf{I}^n$, from which it follows that:

$$|Z_{kl}|^2 = \begin{cases} 1 & k = \text{integer part of } \frac{l+2}{2} \\ 0 & \text{otherwise} \end{cases}$$

A.5 Confinement of v by Subspace Projection Network

This Appendix is concerned with the proof of the assumption that the repeated application of operations (1) and (2) in the subspace projection network will confine $\mathbf{v}$ both to the valid subspace and the unit hypercube.

First assume that $\mathbf{v}$ at time t satisfies both these constraints. The value of $\mathbf{v}$ at time $t + \Delta t$ is given by $\mathbf{v} + \Delta\mathbf{v}$ where $\Delta\mathbf{v}$ is given by the dynamic equation (5.15),

$$\Delta\mathbf{v} = \Delta t\dot{\mathbf{v}} = \Delta t[\mathbf{T}^{\text{op}}\mathbf{v} + \mathbf{i}^{\text{op}}]$$

If $\mathbf{v} + \Delta\mathbf{v}$ is to remain both within the valid subspace and the unit hypercube, then $\Delta\mathbf{v}$ must satisfy the following conditions:

(c1) $\Delta\mathbf{v}$ is such that $\mathbf{v} + \Delta\mathbf{v}$ remains within the unit hypercube.

(c2) $\Delta\mathbf{v}$ lies wholly in the zerosum subspace.

Unfortunately it is very unlikely that $\Delta\mathbf{v}$ will satisfy both (c1) and (c2). However there will always be a component of $\Delta\mathbf{v}$, say $\Delta\hat{\mathbf{v}}$, which does satisfy both these conditions.

A.5.1 The projection Matrix $\hat{\mathbf{T}}$

We now show that there exists a projection matrix $\hat{\mathbf{T}}$ such that,

$$\Delta\hat{\mathbf{v}} = \hat{\mathbf{T}}\Delta\mathbf{v} \tag{A.12}$$

and that repeated application of operations (1) and (2) on $\mathbf{v} + \Delta\mathbf{v}$, is equivalent to the multiplication of $\Delta\mathbf{v}$ by $\hat{\mathbf{T}}$. i.e,

$$\mathbf{v} + \Delta\mathbf{v} \overset{(1)(2)\cdots(1)(2)}{\rightarrow} \mathbf{v} + \hat{\mathbf{T}}\Delta\mathbf{v} = \mathbf{v} + \Delta\hat{\mathbf{v}}$$

Let $\mathcal{A}^0$ denote the set of elements of $\mathbf{v}$ for which the inequality constraint $0 \leq v_q$ is active i.e. $v_q = 0$ and $\Delta v_q < 0$. The set $\mathcal{A}^0$ is defined formally by,

$$\mathcal{A}^0 = \{q : v_q = 0 \text{ and } \Delta v_q < 0\} \quad q \in \{1, \ldots, nm\} \tag{A.13}$$

Let $\mathcal{A}^1$ denote the set of elements of $\mathbf{v}$ for which the inequality constraint $1 \geq v_q$ is active i.e. $v_q = 1$ and $\Delta v_q > 0$. The set $\mathcal{A}^1$ is defined formally by,

$$\mathcal{A}^1 = \{q : v_q = 1 \text{ and } \Delta v_q > 0\} \quad q \in \{1, \ldots, nm\} \tag{A.14}$$

Let $\mathcal{A}$ denote the set given by the union of $\mathcal{A}^0$ and $\mathcal{A}^1$, i.e,

$$\mathcal{A} = \mathcal{A}^0 \cup \mathcal{A}^1 \tag{A.15}$$

It can be seen that if,

$$\Delta\hat{v}_q = 0 \;\; \forall \; q \in \mathcal{A} \tag{A.16}$$

and $|\Delta\hat{\mathbf{v}}|$ is small enough to ensure that,

$$0 \leq v_p + \Delta\hat{v}_p \leq 1 \;\; \text{for all } p \notin \mathcal{A} \tag{A.17}$$

then $\Delta\hat{\mathbf{v}}$ satisfies (c1).

Let $\mathbf{T}^{\mathrm{ia}}$ be a $NM \times NM$ matrix where,

$$[\mathbf{T}^{\mathrm{ia}}]_{pq} = \begin{cases} 0 & \text{if } p \neq q \\ 0 & \text{if } p = q \quad \text{and } p \in \mathcal{A} \\ 1 & \text{if } p = q \quad \text{and } p \notin \mathcal{A} \end{cases} \tag{A.18}$$

It can be seen $\mathbf{T}^{\mathrm{ia}}$ is a projection matrix since,

$$\mathbf{T}^{\mathrm{ia}}\mathbf{T}^{\mathrm{ia}} = \mathbf{T}^{\mathrm{ia}}$$

Further if $\Delta\hat{\mathbf{v}}$ satisfies (A.16) then,

$$\Delta\hat{\mathbf{v}} = \mathbf{T}^{\mathrm{ia}}\Delta\hat{\mathbf{v}} \tag{A.19}$$

Similarly if $\Delta\hat{\mathbf{v}}$ satisfies (c2) (i.e. $\Delta\hat{\mathbf{v}}$ lies in the zerosum subspace) then,

$$\Delta\hat{\mathbf{v}} = \mathbf{T}^{\mathrm{zs}}\Delta\hat{\mathbf{v}} \tag{A.20}$$

Let the subspace spanned by all vectors $\mathbf{u}$ which satisfy $\mathbf{u} = \mathbf{T}^{\mathrm{ia}}\mathbf{u}$ be termed the inactive subspace. It can be seen that $\hat{\mathbf{T}}$ is simply the matrix which projects $\mathbf{v}$ into the subspace which is the intersection of the zerosum subspace and the inactive subspace. Hence $\hat{\mathbf{T}}$ satisfies,

$$\begin{aligned} \hat{\mathbf{T}} &= \mathbf{T}^{\mathrm{zs}}\hat{\mathbf{T}} \\ \hat{\mathbf{T}} &= \mathbf{T}^{\mathrm{ia}}\hat{\mathbf{T}} \end{aligned} \tag{A.21}$$

Now assuming $\Delta\mathbf{v}^{(0)} = \Delta\mathbf{v}$, let,

$$\Delta\mathbf{v}^{(n)} = \Delta\hat{\mathbf{v}} + \mathbf{u}^{(n)}$$

where,

$$\begin{aligned} \Delta\mathbf{v}^{(n)} &= \begin{cases} \mathbf{T}^{\mathrm{zs}}\Delta\mathbf{v}^{(n-1)} & \text{if } n \text{ is odd} \\ \mathbf{T}^{\mathrm{ia}}\Delta\mathbf{v}^{(n-1)} & \text{if } n \text{ is even} \end{cases} \\ &= \begin{cases} \Delta\hat{\mathbf{v}} + \mathbf{T}^{\mathrm{zs}}\mathbf{u}^{(n-1)} & \text{if } n \text{ is odd} \\ \Delta\hat{\mathbf{v}} + \mathbf{T}^{\mathrm{ia}}\mathbf{u}^{(n-1)} & \text{if } n \text{ is even} \end{cases} \\ \Rightarrow \mathbf{u}^{(n)} &= \begin{cases} \mathbf{T}^{\mathrm{zs}}\mathbf{u}^{(n-1)} & \text{if } n \text{ is odd} \\ \mathbf{T}^{\mathrm{ia}}\mathbf{u}^{(n-1)} & \text{if } n \text{ is even} \end{cases} \end{aligned} \tag{A.22}$$

Since $\mathbf{T}^{\mathrm{zs}}$ and $\mathbf{T}^{\mathrm{ia}}$ are projection matrices,

$$\begin{aligned} |\mathbf{T}^{\mathrm{zs}}\mathbf{u}^{(n-1)}| &\le |\mathbf{u}^{(n-1)}| \\ \text{and } |\mathbf{T}^{\mathrm{ia}}\mathbf{u}^{(n-1)}| &\le |\mathbf{u}^{(n-1)}| \end{aligned}$$

Hence

$$|\mathbf{u}^{(n)}| \le |\mathbf{u}^{(n-1)}| \tag{A.23}$$

Multiplying by $\mathbf{T}^{\mathrm{zs}}$ corresponds to operation (1) while multiplying by $\mathbf{T}^{\mathrm{ia}}$ corresponds to operation (2). From equation (A.23) it is clear that as $n \to \infty$, $\Delta\mathbf{v}^{(n)} \to \Delta\hat{\mathbf{v}}$. Hence the repeated application of operations (1) and (2) is equivalent to the projection performed on $\Delta\mathbf{v}$ by multiplication with $\hat{\mathbf{T}}$. (**N.B.** This proof assumes that (A.22) does not affect the set of active inequality constraints, $\mathcal{A}$, and hence $\mathbf{T}^{\mathrm{ia}}$ remains the same for all n. Assuming (A.17) holds, this can be proved, although for conciseness the proof is not given here.)

Since $\mathbf{v} + \Delta\hat{\mathbf{v}}$ satisfies (c1) and (c2) it follows that repeated application of operations (1) and (2) does indeed confine $\mathbf{v}$ to both the valid subspace and the unit hypercube, regardless of the change in $\mathbf{v}$ effected by $\Delta\mathbf{v}$.

Appendix B

Implementation of Subspace Projection Network

Essentially the Subspace Projection Network was implemented exactly as shown in Fig 5.3. The specific details of the implementation were:

- For each traversal of the bottom loop in Fig 5.3 it was necessary to make several traversals of the top loop in order to ensure that operations (1) and (2) were applied enough times to enforce confinement within the valid subspace and unit hypercube. Rigid enforcement of these confinement conditions would in theory have required an infinite number of loops, however in practise it was found sufficient to go round the top loop enough times to ensure that,

$$\left| \sum_{p=1}^{N^2} v_q - N \right| < 0.1N \tag{B.1}$$

- The time step size Δt was dynamically computed so that

$$\frac{|\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{v}}|}{|\mathbf{v}^{zs}|}\Delta t = 0.06$$

  An exact calculation of $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{v}}$ is very difficult, hence an estimate was obtained by applying operations (1) and (2) to $\mathbf{v} + \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}$ and subtracting $\mathbf{v}$ from the result.

- Since all the matrix multiplications in Fig 5.3 were of the form

$$\mathbf{u} = (\mathbf{P} \otimes \mathbf{Q})\mathbf{v}$$

  the Kronecker product identity (2.14) was applied to re-express the matrix-vector multiplications as matrix-matrix multiplications:

$$\begin{aligned} \mathbf{u} &= (\mathbf{P} \otimes \mathbf{Q})\mathbf{v} \\ \Rightarrow \mathrm{vec}(\mathbf{U}^T) &= \mathrm{vec}(\mathbf{Q}^T\mathbf{V}^T\mathbf{P}) \\ \Rightarrow \mathbf{U} &= \mathbf{P}^T\mathbf{V}\mathbf{Q} \end{aligned}$$

Hence operation (1) became,

$$\mathbf{V} = \mathbf{S} + \mathbf{R}^n \mathbf{V} \mathbf{R}^n \quad \text{where } \mathbf{s} = \text{vec}(\mathbf{S}^T)$$

and without MGNC, operation (3) became,

$$\Delta \mathbf{V} = \Delta t \mathbf{P}^T \mathbf{V} \mathbf{Q}$$

If MGNC version 2 was being used, then operation (3) was computed by:

$$\Delta \mathbf{V} = \Delta t \mathbf{P}^T \mathbf{V}(\mathbf{Q} + \kappa \mathbf{I}^n)$$

- When version 2 MGNC was being used, it was implemented by starting κ at $-\gamma_3$ (assuming the γ_l's are ordered according to (3.37)) and incrementing it by 0.1 (up to a maximum of $-\gamma_n$) each time that,

$$\frac{\Delta \hat{\mathbf{v}}^T \mathbf{v}}{|\mathbf{v}^{zs}|} < \epsilon$$

where the threshold ϵ was set to 0.0001 for Hopfield and Tank's 30-city TSP [25]. Since a large time step was being used, even when $|\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{v}}|$ was very small, in practice it turned out that $\Delta \mathbf{v}$ was large due to oscillations caused by the large time step. Under these conditions the above criterion for changing κ failed to change it at the correct time. In order to iron out these effects, $\Delta \hat{\mathbf{v}}$ was computed using

$$\Delta \hat{\mathbf{v}} = \arg \min_{p=1 \cdots 5} \Big(\mathbf{v}(t) - \mathbf{v}(t - p\Delta t) \Big)$$

Bibliography

[1] Abe, S. *Theories on the Hopfield Neural Networks*, Proc. IJCNN 1989.

[2] Aiyer, S. V. B., Niranjan, M. & Fallside, F. *A Theoretical Investigation into the Performance of the Hopfield Model*, IEEE Transactions on Neural Networks, Vol. 1, No. 2, June 1990.

[3] Aiyer, S. V. B., Niranjan, M. & Fallside, F. *On the Performance of the Hopfield Model*, Proc. INNS Paris, July 1990.

[4] Aiyer, S. V. B. & Fallside, F. *A Subspace Approach to Solving Combinatorial Optimization Problems with Hopfield Networks*, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR 55, December 1990.

[5] Aiyer, S. V. B. & Fallside, F. *A Hopfield Network Implementation of the Viterbi Algorithm for Hidden Markov Models*, Proc. IJCNN-91 Seattle, July 1991.

[6] Amit, D. J., Gutfreund, H. & Sompolinsky, H. *Spin-Glass models of neural networks*, Phys. Rev. A, 32, p1007, 1985.

[7] Anderson, J. R. *A Mean Field Computational Model for PDP*, Proceedings of the 1988 Connectionist Models Summer School, Carnegie Mellon University.

[8] Angéniol, B., de la Croix Vaubois, G. & le Texier, J. *Self-Organizing Feature Maps and the Travelling Salesman Problem*, Neural Networks, Vol. 1, pp. 289-293, 1988.

[9] Bienenstock, E. & Doursat, R. *Elastic Matching and Pattern Recognition in Neural Networks*, Proc. nEuro Paris, 1988.

[10] Bilbro, G., Miller III, M. T., Snyder, W., Van den Bout, D. E., White, M. *Mean field annealing and neural networks* Advances in Neural Information Processing Systems, Morgan-Kaufmann, p91-98, 1989.

[11] Blake, A. & Zisserman, A. *Visual Reconstruction* The MIT Press series in artificial intelligence, London **1987**.

[12] Chiu,C., Maa, C-Y. & Shanblatt,M.A. *An artificial neural network for dynamic programming* International Journal of Neural Systems, Vol. 1, No. 3, 1989.

[13] Chiu,C., Maa, C-Y. & Shanblatt,M.A. *Energy Function Analysis of Dynamic Programming Neural Networks* IEEE Transactions on Neural Networks, Vol. 2, No. 4, July 1991.

[14] Davidor, Y. *An Intuitive Introduction to Genetic Algorithms as Adaptive Optimizing Procedures*, Technical Report CS90-07, Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, April 1990.

[15] Durbin, R. & Willshaw, D. *An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method*, Nature 326 (6114), pp. 689-691, April 1987.

[16] Eberhardt, S. P., Daud, T., Kerns, D. A., Brown, T. X. & Thakoor, A. P. *Competitive Neural Architecture for Hardware Solution to the Assignment Problem*, Neural Networks, Vol. 4, pp. 431-442, 1991.

[17] Fritzke, B. & Wilke, P. *FLEXMAP - A Neural Network For The Traveling Salesman Problem With Linear Time and Space Complexity*, Submitted to Proc. IJCNN-91 Singapore, 1991.

[18] Fu, Y. & Anderson, P. W. *Application of statistical mechanics to NP-complete problems in combinatorial optimisation* J. Physics A, vol. 19, p1605-1620, 1986.

[19] Garey, M. R. & Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness* Freeman, San Francisco, 1979.

[20] Gee, A. H., Aiyer, S. V. B. & Prager, R. W. *Neural Networks and Combinatorial Optimization Problems - the Key to a Successful Mapping*, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR 77, 1991.

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

[22] Hegde, S. U., Sweet J. L. & Levy W. B. *Determination of parameters in a Hopfield/Tank computational network* Proc. ICNN 1988.

[23] Hopfield, J. J. *Neural networks and physical systems with emergent collective computational abilities*, Proc. Natl. Acad. USA, Vol. 79, pp. 2554-2558, April 1982.

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

[25] Hopfield, J. J. & Tank, D. W. *"Neural" Computation of Decisions in Optimization Problems*, Biological Cybernetics 52, 141-152, 1985.

[26] Kamgar-Parsi, B. & Kamgar-Parsi, B. *On Problem Solving with Hopfield Neural Networks*, Biological Cybernetics, Vol. 62, pp. 415-423, 1990.

[27] Kanter, I. & Sompolinsky, H. *Graph optimization problems and the Potts glass*, Journal of Physics A, Vol. 20, L673-L679, 1987.

[28] Kernighan, B. W. & Lin, S. *An Efficient Heuristic Procedure for Partitioning Graphs*, The Bell System Technical Journal, February 1970.

[29] Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. *Optimization by Simulated Annealing*, Science, Vol. 220, No. 4598, May 1983.

[30] Kohonen, T. *Self-Organization and Associative Memory*, Springer Series on Information Sciences, Springer-Verlag, 1989.

[31] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. & Shmoys, P. B. *The Traveling Salesman Problem*, J.Wiley & Sons **1985**.

[32] Lee. B. W. & Sheu, B. J. *Modified Hopfield Neural Networks for Retrieving the Optimal Solution*, IEEE Transactions on Neural Networks, Vol. 2, No. 1, January 1991.

[33] Lee, Kai-Fu *Automatic speech recognition: the development of the SPHINX system* Kluwer Academic Publishers 1989.

[34] Lee, Kai-Fu & Hon, H.-W. *Speaker-Independent Phone Recognition Using Hidden Markov Models* IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 37, No. 11, November 1989.

[35] Lamel, L. F., Kasel, R. H. & Seneff S. *Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus*, Proceedings of the DARPA Speech Recognition Workshop, p26-32, March 1987.

[36] Levinson, S. *Structural Methods in Automatic Speech Recognition* Proc. IEEE, Vol. 73, No. 11, Nov. **1985**.

[37] Li, W. & Nasrabadi, N. M. *Object Recognition based on Graph Matching Implemented by a Hopfield-Style Neural Network*, Proc. IJCNN Washington DC, 1989.

[38] Lin, S. & Kernighan, B. W. *An effective heuristic algorithm for the traveling salesman problem* Operational Research, 11, p972-989, 1973.

[39] Murray, A. F., Corso, D. Del & Tarassenko, L. *Pulse-Stream VLSI Neoral Networks Mixing Analog and Digital Techniques* IEEE Transactions on Neural Networks, Vol. 2, No. 2, March 1991.

[40] McCulloch, W.S. & Pitts, W. *A logical calculus of ideas immanent in nervous actvity*, Bulletin of Mathematical Biophysics, No. 5, 1943.

[41] Neugebauer, C. F. & Yariv, A. *A Parallel Analog CCD/CMOS Neural Network IC*, Proc. IJCNN Seattle, 1991.

[42] Peterson, C. & Anderson, J. R. *Neural Networks and NP-complete Optimization Problems; A Performance Study on the Graph Bisection Problem*, Complex Systems 2:1, 1988.

[43] Peterson, C. & Söderberg, B. *A new method for mapping optimization problems onto neural networks*, International Journal of Neural Systems, Vol. 1, No. 1, 1989.

[44] Simic, P. D. *Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations*, Network - Computation in Neural Systems, Vol. 1,No. 1, January 1990.

[45] Van den Bout, D. E. & Miller III, T. K. *Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing*, Biological Cybernetics 62, pp. 129-139, 1989.

[46] Van den Bout, D. E. & Miller III, T. K. *Graph Partitioning using Annealed Neural Networks*, IEEE Transactions on Neural Networks, Vol. 1, No. 2, June 1990.

[47] Von der Malsberg, C. *Pattern Recognition by Labeled Graph Matching*, Neural Networks, Vol. 1, pp. 141-148, 1988.

[48] Wilson, V. & Pawley, G. S. *On the Stability of the TSP Problem Algorithm of Hopfield and Tank*, Biological Cybernetics 58, 63-70, 1988.

[49] Winston, P. H. *Artificial Intelligence*, Addison-Wesley, Reading MA, 1984.

[50] Wu, F. Y. *The Potts model*, Review of Modern Physics No. 54, 1983.

[51] Yanai, H. & Sawada, Y. *Associative memory network composed of neurons with hysteretic property*, Neural Networks, Vol. 3, pp. 223-228, 1991.