MONASH UNIVERSITY
THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
ON..................2 November 2001....................

.........................[redacted]..

For Sec. Research Graduate School Committee

NEURAL NETWORKS WITH NONLINEAR SYSTEM DYNAMICS FOR COMBINATORIAL OPTIMIZATION

Terence Kwok
B.Sc(Hons), MbusSyst

School of Business Systems,
Monash University.

June, 2001.

Submitted in total fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

For the past decade or so, neural networks have been developed as a new computing tool to solve combinatorial optimization problems (COP's). One of the main advantages of neural networks is that they promise a dramatic performance gain when implemented in parallel hardware. This factor makes neural networks an attractive option for solving COP's commonly encountered in business and engineering applications. However, for neural networks to be a robust and effective tool for practical use, further research into their optimization dynamics is needed to improve their optimization performance such as solution quality, efficiency, etc.

There are currently two major neural approaches to COP's: the Hopfield-type and the self-organizing approach, each with limitations of its own. The traditional Hopfield-type approaches are based on the minimization of an energy function via the steepest descent dynamics of the Hopfield network, which often result in local minimum solutions. Furthermore, solution feasibility is sensitively dependent on the choice of penalty parameters. For the self-organizing approaches, a number of models have been limited to solving Euclidean problems only, and most of them suffer from convergence problems that affect solution quality.

In this thesis, we study the theoretical foundations of using nonlinear system dynamics (NSD) to improve the optimization performance of the two neural

approaches. This includes investigations on the generation, characterization, and exploitation of NSD in the neural networks for improved optimization. For the Hopfield-type approach, we study the Hopfield-type chaotic neural networks (HCNN's) which use chaotic dynamics for performance enhancement. A theoretical framework unifying various existing HCNN models is proposed, which serves the purpose of classifying the chaotic models in a new dynamical perspective. We then experimentally investigate the optimization performance of the HCNN models by their implementation to solve the *N*-queen problem, which is an example of constraint satisfaction problems. Computer simulation results of the chaotic simulated annealing (CSA) models show that the associated chaotic dynamics are able to improve optimization performance by having good feasibility, efficiency, robustness and scalability. Similar measurements on a chaotic noise model also show a corresponding performance enhancement. By combining the parametric study of performance with observations of the dynamical behaviors during the solution process, we identify characteristic chaotic dynamics responsible for improved solution qualities, and provide a guide for choosing the corresponding parameters.

For the self-organizing approach, we study the self-organizing neural network with weight normalization (SONN-WN) for the role of NSD in improving optimization performance. The SONN-WN is implemented to solve the *N*-queen problem via computer simulations. Through a parametric study of the annealing schedules associated with the normalization process, we demonstrate the use of noise and small neighborhoods for improvements in feasibility and robustness. By means of a detailed computational investigation of the network's convergence dynamics, a special

normalization temperature range for ensuring effective convergence to good quality solutions is discovered. These results enable us to propose a noise-induced mechanism for oscillation control that leads to improved convergence, as well as providing a guide for choosing annealing schedules of the normalization process. A theoretical model describing the essential equilibrium dynamics of the SONN-WN is also derived. The model reveals the key role of bifurcation dynamics in characterizing the convergence behaviors observed in our experiments. Furthermore, a range of nonlinear phenomena is revealed by the equilibrium model, thereby opening up a dynamical perspective for understanding and improving the SONN-WN.

Acknowledgements

I am greatly indebted to my supervisor, Dr. Kate Smith, for providing me the opportunity and freedom to work on this research. This thesis would not have been possible without her guidance and support over the past few years. I would like to thank Prof. Lipo Wang for his advice on various aspects of this research. Members of the School of Business Systems at Monash University have also been very helpful throughout my candidature. Special thanks to the staff of the PPME facility at the School of Computer Science and Software Engineering (Monash University), for their technical support and provision of the fabulous computing cluster. I am grateful to both Monash University and the School of Business Systems for the MGS scholarship and the supplementary funding. I would also like to thank my family for their constant support. Finally, I would like to dedicate this thesis to my mother and grandmother, for their sufferings endured in the course of this research.

Declaration

This is to certify that, except where other authors have been cited, this thesis is the result of my own original work conducted between February 1998 and June 2001. The thesis is less than 100,000 words in length and contains no material which has been accepted for the award of any other degree or diploma in any university or other institution.

[redacted]

Contents

List of Figures

Notation

The following is a list of notations used in this thesis:[1]

Combinatorial Optimization

$\boldsymbol{x}$	vector solution of a 0-1 COP
$\mathbf{X}$	matrix solution of a 0-1 COP
$f(\boldsymbol{x})$	objective function to be minimized
$\mathbf{R}$	constraint coefficient matrix

Hopfield-type Neural Networks

y_i	internal state of neuron *i*
x_i	output state of neuron *i*
Y_{ij}	internal state of neuron (*i*, *j*)
X_{ij}	output state of neuron (*i*, *j*)
W_{ij}	weight of the connection between neuron *i* and *j*
V_{ij}	weight on the external input *j* applied to neuron *i*
θ_i	threshold of neuron *i*
I_i	external input of neuron *i*
E_{Hop}	energy function of the continuous Hopfield network

[1] The original notations from various sources have been altered where necessary for consistency.

E_d	energy function of the discrete Hopfield network
E	generalized energy function of Hopfield-type networks
R_i	trans-membrane resistance of neuron i
C_i	input capacitance of neuron i
$g(y_i)$	activation function of neuron i
ε	gain control parameter of the sigmoidal activation function
α	refractory scaling parameter ($\alpha > 0$)
β	damping factor ($0 \leq \beta \leq 1$)
k	refractory decay parameter ($0 \leq k \leq 1$)
I_0	positive parameter
τ	time constant parameter ($\tau > 0$)
$z_i(t)$	self-feedback connection weight (or refractory strength) ($z_i(t) \geq 0$)
$\eta_i(t)$	normalized time series of noise added to neuron i
$\xi(t)$	time series of noise with controlled auto-correlation
ϕ	auto-correlation control parameter for $\xi(t)$
A_p, A_η	noise amplitude factor
a	bifurcation parameter of the iterated logistic map
$H_{\mathrm{CA},\,\mathrm{WS},\,\eta}$	Hopfield energy modifier for HCNN models
$\lambda(t)$	time dependent positive parameter
γ	positive parameter ($0 < \gamma < 1$)

Self-Organizing Neural Networks

x	input vector

$\mathbf{W}$	weight matrix
W_{ij}	weight element connecting node *i* and input element *j*
W_i	reference vector of node *i*
i_0	index of the winning node
$\alpha\,(i_0, i, t)$	neighborhood function centered on the winning node i_0
$\beta(t)$	learning rate of Kohonen's SOFM update rule ($0 < \beta < 1$)
r_β	decay rate of $\beta(t)$ ($0 \le r_\beta \le 1$)
$T(t)$	normalization temperature in the weight normalization procedure
r	annealing rate of $T(t)$ ($0 \le r \le 1$)
r_i	position vector of node *i* in the array of nodes
η	neighborhood size in Kohonen's weight updating
$\sigma(t)$	factor controlling the width of the Gaussian function
V_{ij}	cost potential of assigning input element *j* to node *i*
$\xi_i(t)$	random noise added to node *i* ($\xi_i \in [0, 1]$)

The Equilibrium Model of the SONN-WN

w_j	the *j*-th weight of the winning node
T^*	symmetry-bifurcation temperature
$T_{b1,\,b2...}$	temperatures at successive period-doubling bifurcations

Acronyms and Abbreviations

COP	Combinatorial Optimization Problem
TSP	Travelling Salesman Problem
ILP	Integer Linear Programming
SA	Simulated Annealing
H-T	Hopfield-Tank
CNN	Chaotic Neural Network
HCNN	Hopfield-type Chaotic Neural Network
CSA	Chaotic Simulated Annealing
GCM	Globally Coupled Map
SOFM	Self-Organizing Feature Map
SONN	Self-Organizing Neural Network
SONN-WN	Self-Organizing Neural Network with Weight Normalization
NSD	Nonlinear System Dynamics
CAM	Content-Addressable Memory
GCM	Globally Coupled Map
TCNN	Transiently Chaotic Neural Network
QAP	Quadratic Assignment Problem
FDT	Flexible Decision Temperature

Chapter 1

Introduction

This thesis studies neural networks with features of nonlinear dynamical systems for solving combinatorial optimization problems. These features include chaos, bifurcation, intermittency, and other attractor properties of nonlinear systems. We investigate how these nonlinear phenomena arise in the networks, their properties, and how they can be exploited to make neural networks more practical and effective for combinatorial optimization. From a theoretical point of view, this can be seen as a generalization of traditional neural networks to cover a wider spectrum of possible dynamics and functionality. We begin this chapter with an introduction to combinatorial optimization, and describe various traditional neural approaches for tackling the problem. Theoretical and practical limitations of these approaches are then outlined, from which we draw the motivation of this thesis. A brief summary of basic concepts in chaos theory and related dynamics is also included. We then present the objectives and major contributions of this thesis, followed by an overview of its organization.

1.1 Combinatorial optimization

Combinatorial optimization problems (COP's) are concerned with finding optimal solutions to satisfy a given set of constraints. Many problems in engineering and business are found to be combinatorial in nature, therefore efficient and robust methods to obtain optimal solutions are highly sought after. A well-known example of a COP is the travelling salesman problem (TSP) [67]. Given a number of cities to visit, a salesman has to find the shortest route around them with each city visited exactly once. The constraint here is the requirement of visiting each city exactly once, and an optimal solution is the shortest route satisfying this constraint. In this example and also in general, there is usually more than one optimal solution to each COP. Because of the combinatorial nature of these problems, the solutions are discrete and usually expressed as matrices of 0 and 1 (0-1 solution). A general optimization problem with a quadratic objective function $f(x)$ with linear constraints can be written as

$$\text{Minimize} \quad f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\mathrm{Q}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{1.1}$$

$$\text{subject to} \quad \mathrm{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.2}$$

$$x_i \in \{0, 1\} \quad i = 1, \ldots, N \tag{1.3}$$

An optimal solution to the problem specified by (1.1), (1.2) and (1.3) is called a global minimum, and a local minimum is a sub-optimal solution. An infeasible solution is one that does not satisfy (1.2). Common problems of this class are: the TSP, transportation problem, facilities layout problem, vehicle scheduling problem, etc. In this thesis, the *N*-queen problem which also belongs to this class is used as a test case for our investigations.

For a class of problems called integer linear programming (ILP) where both the objective function and constraints are linear, exact methods are available for guaranteed optimal solutions. There are two broad approaches: enumeration techniques and *cutting planes* [40]. Examples of implicit enumeration techniques are *branch-and-bound* [66] and dynamic programming [17]. These methods are only useful for small-sized problems due to their complexity. As the problem size grows, they become too consuming in time and resources even with modern computers. Because of combinatorial explosion, many COP's belong to a class called *NP-hard* problems [60]. Algorithms whose computational time is a polynomial function of the problem size are unlikely to be found for these problems.

Since many COP's found in engineering and business are nonlinear, medium to large scale, and often NP-hard, more practical approaches are required due to limitations in time and resources. To this end, heuristics have been developed to deliver near-optimal solutions in reasonable computational time. An example is the local search method, where the optimal solution is obtained by searching through neighboring feasible vertex solutions. A steepest descent mechanism is often employed in this method, and while this converges quickly to a stable solution, it is often only a locally optimal solution. A more robust and widely used heuristic is simulated annealing

(SA) [61], which enables local minima to be escaped in the search for globally optimal solutions. It was inspired by the physics of statistical mechanics governing the annealing of a molten metal to its solid state. The transition of the atomic configuration from a disordered state to a more ordered one as the temperature decreases is described by the Boltzmann distribution, which is exploited by SA as the functional mechanism for optimization. Since uphill moves on the energy landscape are allowed, this method can avoid being trapped in local minima. One important feature of SA is its stochastic element and its role in the optimization process, which is also shared to a certain extent by some neural network methods described later. Many other heuristics have been developed for general or special purposes, e.g. tabu search, maximum spanning trees, etc. Since the treatment of heuristics lies outside the scope of this thesis, interested readers should refer to other sources, e.g. Reeves [82].

1.2 Neural approaches to combinatorial optimization

For the past decade or so, neural networks have become a new tool for solving COP's. There are currently two main types of neural networks for this purpose: the Hopfield neural network and self-organizing approaches. In this thesis, we base our investigations on these two types of neural networks. In order to draw the motivation behind this thesis, a brief outline of the neural approaches is given here, and more detailed literature surveys are presented in Chapter 2 and 5.

In 1985, Hopfield and Tank first demonstrated that the TSP could be solved using a Hopfield neural network [54]. This marks the beginning of using neural networks to solve NP-hard problems. The seminal paper of Hopfield and Tank introduced the mapping of a quadratic objective function of a COP to the Hopfield energy, where the optimization is achieved by the network's convergence to the nearest fixed point via steepest descent dynamics. This approach treats the Hopfield network as a dynamical system, and exploits the local behavior of fixed points to fulfill the goal of optimization. However, other nonlinear phenomena and global dynamics, together with their potential functionality, have been largely unexplored.

Despite the success of bridging the fields of neural networks and combinatorial optimization, many problems have been identified with the Hopfield-Tank (H-T) approach as a practical tool. There were initial difficulties in the determination of various parameters, as well as the problem of infeasible solutions and poor solution quality [111]. As a result, techniques have been devised to improve the feasibility problem by tuning the parameters [57, 65] and modifying the energy function [20, 103]. Although the problem of feasibility was subsequently solved [6, 39], solution quality remains to be improved. Not surprisingly, many modifications and variations of the Hopfield network have been proposed to improve the solution quality, and some of the more successful ones are stochastic in nature, e.g. Boltzmann machines [51], Cauchy machines [96] and Gaussian machines [8]. The improvement in solution quality in these methods is principally achieved by incorporating principles of SA to avoid local minima.

Recently, a class of Hopfield-type neural networks exploiting the global dynamics of nonlinear systems has been developed for improved optimization performance. They are generally called chaotic neural networks (CNN's), or more specifically Hopfield-type chaotic neural networks (HCNN's), which feature the nonlinear phenomenon of chaos in various ways. Broadly speaking, they can be divided into two major classes. One is the *internal* approach where chaos is generated within the network. Examples of this type include chaotic simulated annealing (CSA) approaches [24, 107] and the globally coupled map (GCM) model [75]. The other type is the *external* approach, where an externally generated chaotic signal is added to the network as chaotic noise [15, 47, 48]. Many of these approaches have been used to solve the TSP, with measured performance gains including improved feasibility, solution quality, robustness, ease of parameter choices, etc., when compared to the basic H-T approach.

The other major class of neural networks for solving COP's is based on Kohonen's self-organizing feature map (SOFM) [62]. Most of these efforts have been directed to solving the TSP with the *elastic net* approach [30]: an 'elastic band' or a ring of nodes representing a tour is stretched to seek the shortest path. This technique of combining SOFM with the elastic net method have been limited to solving problems defined in the Euclidean plane, where the inter-node distances on the 'elastic band' are measured in the same space [34, 37]. To extend self-organizing networks beyond elastic net based approaches and thereby solving a broader class of "0-1" optimization problems, a more general model called the self-organizing neural network (SONN) was later developed to operate within feasible permutation matrices rather than the Euclidean space of the elastic net [90, 91].

Common among the self-organizing approaches is the oscillation problem during convergence. This occurs when cheaper nodes are favored in the competition to be the winners, leading to infeasible or local minimum solutions. Many ways have been proposed to suppress the oscillations, e.g. by the creation/deletion of nodes [11], by punishing the 'greedy' nodes with a *conscience* mechanism [22, 29], or by the incorporation of explicit statistics into the learning [13]. For the SONN, a weight normalization procedure was introduced [45, 44] for the purpose of constraint satisfaction, as well as for reducing oscillations. The normalization procedure involves the lowering of a temperature parameter as the network learning proceeds, which is akin to the cooling process in SA. How this normalization process interacts with the self-organizing dynamics of the network is an important open question in terms of understanding and improving the optimization ability of the SONN. Also, solution quality depends sensitively on the cooling parameters, which are often difficult to choose.

Most of the research mentioned so far, for both the Hopfield-based and self-organizing approaches, has been implemented as computer simulations in software to solve the COP's. In fact, the parallel and interconnected structure inherent in the networks makes them perfect candidates for hardware implementations [53, 85, 106]. The result would be a drastic reduction in running time and cost, which makes neural networks a more powerful tool for solving COP's in engineering and business.

1.3 Chaos and related dynamics

In this section, we briefly outline the basic concepts of chaos and related dynamics relevant to this thesis. Since chaos theory belongs to the enormous field of nonlinear dynamics, a detailed treatment is beyond the scope of this thesis.

For a broad class of systems, their dynamical behaviors can be described by a set of first order differential equations in the form:

$$\begin{aligned} \frac{dx_1}{dt} &= F_1(x_1, x_2, \ldots, x_N) \\ \frac{dx_2}{dt} &= F_2(x_1, x_2, \ldots, x_N) \\ &\vdots \\ \frac{dx_N}{dt} &= F_N(x_1, x_2, \ldots, x_N) \qquad (1.4) \end{aligned}$$

where $\boldsymbol{F}$ is in general a nonlinear, continuous function of the time dependent variable $\boldsymbol{x}$. The space specifying the system state $\boldsymbol{x}$ is called the state space of the system. For dissipative systems, the long-term behavior is largely independent of its transients (although the final *state* can be affected). As the system evolves in time, the trajectory approaches an *attractor*, which may be a point, curve or an area. A *basin of attraction* is the set of initial points in the state space that gives rise to the attractor. To find the fixed point of (1.4), we set $\boldsymbol{F} = \boldsymbol{0}$ and then solve the resultant system of algebraic equations. The local behavior around the fixed point can be determined by a *stability analysis*. For a stable (unstable) fixed point, nearby points in the state space follow convergent (divergent) trajectories as time evolves. For one-dimensional systems ($N = 1$), the only possible dynamics are convergent or divergent behaviors relative to the fixed points. For two-dimensional systems ($N = 2$), cyclic or periodic behaviors can also exist. Because of the fundamental property that trajectories in the state space never cross, chaos cannot exist for $N < 3$, i.e. only in three or higher dimensions can chaos be found (note: this condition refers to continuous *differential* equations, not to discrete *difference* equations).

Although a single definition of chaos is yet to be agreed upon among scientists, some necessary but not sufficient characteristics of chaotic dynamics have been documented [35]:

1 *Deterministic*: chaotic dynamics are completely deterministic. The future of a chaotic system is completely determined by its past history (c.f. random processes where a future state is uncorrelated with the present);

2 *Sensitive dependence on initial conditions*: errors in the initial states grow exponentially with time, rendering accurate long-term predictions impossible;

3 *Ergodic*: a chaotic trajectory always returns to the local region of a previous point along the trajectory (but never repeating itself);

4 *Embedded*: there are an infinite number of unstable periodic orbits embedded in the chaotic attractors.

Since chaos requires at least a three-dimensional state space to describe its dynamics, a low dimensional section of the state space called the Poincaré section is commonly used for analysis. With this approach, the continuous dynamics can be described by a set of *difference* equations of lower dimensions. A well-known example of a difference equation is the logistic map:

$$x(n+1) = ax(n)(1-x(n)) \tag{1.5}$$

where n denotes the iteration, and a is the bifurcation parameter. The logistic map (1.5) captures most of the important nonlinear phenomena observed in continuous systems of three-dimensions or higher. Starting from an initial point $x(0)$, (1.5) is to be computed repeatedly to represent a discrete time evolution of the dynamical system. It can be seen that a strong feedback effect is at work in (1.5) where the present value $x(n)$ nonlinearly generates $x(n + 1)$. The feedback strength is controlled by the parameter a, which determines the characteristics of the whole system. Figure 1.1(a) shows the long-term state values of x with varying values of a (the early transients are not shown). Such a diagram is called the bifurcation diagram of the system (1.5).

From Figure 1.1(a), the period-doubling bifurcation route to chaos can be observed. For $0 \leq a \leq 3$, the system converges to a fixed point. For $a > 3$, the fixed point becomes unstable and bifurcates into a two-cycle, where the long-term behavior is an oscillation between two x values. Successive bifurcations can be observed from Figure 1.1(a), and such a scenario is called a *cascade* of bifurcations. For $a > 3.57$ (approx.), the system exhibits deterministic chaos interspersed with windows of periodic behaviors. A quantitative measure known as the Lyapunov exponent (λ_{Lya}) is used to describe the stability of attractors. The Lyapunov exponent of the logistic map is plotted in Figure 1.1(b). Negative values ($\lambda_{Lya} < 0$) indicate periodic or quasi-periodic motion, while positive values ($\lambda_{Lya} > 0$) indicate chaos.

Another bifurcation event relevant to this thesis is a phenomenon known as *crisis*. A crisis occurs when a chaotic attractor vanishes or suddenly increases in size [42]. It is caused by the "collision" of an unstable fixed point or unstable limit cycle with the chaotic attractor as some system parameter is changed. Common to systems with multi-stable states is another bifurcation phenomenon known as symmetry-breaking bifurcation. It occurs when a stable state is destabilized by a change of system parameters, and the new stable state is "decided" among many others by very small fluctuations. In other words, the symmetry between the stable states is broken by the outcome of the event.

Since (1.5) is only a one-dimensional example of an iterative map, there are others in two or higher dimensions. A famous two-dimensional example is the Hénon map, whose chaotic attractor, or *strange attractor*, can be observed in a two-dimensional space. For a more detailed treatment of iterative maps and chaos in general, there are some other excellent sources available e.g. Peitgen *et al* [78] and Hilborn [50].

1.4 Motivation

This thesis is motivated by the current need to fine-tune neural network algorithms in order to ensure convergence to optimal solutions. In light of the promising perspective of using nonlinear system dynamics (NSD) in neural networks for improved optimization performance, we study in this thesis the theoretical foundations of this new approach. For the Hopfield-type chaotic neural networks (HCNN's), there are important similarities and differences among different models both in construction and performance. An investigation is thus required to experimentally compare the optimization performance of various HCNN models when solving the same COP, and then theoretically examine their characteristics to explain their behavior. For the self-organizing approach, there is currently a lack of self-organizing models focusing on NSD for COP's, which are potential candidates to address the issues of convergence and solution quality. Therefore, just like the Hopfield approach, a dynamical system perspective is needed to study the possible role of NSD in enhancing the optimization performance of the self-organizing approach.

1.5 Objectives and major contributions

The main theme of this thesis is the effective exploitation of nonlinear system dynamics (NSD) in neural networks for enhanced performance in solving combinatorial optimization problems. The theoretical basis of how NSD can be effectively utilized for improved optimization is investigated for the two main types of neural networks for combinatorial optimization. For the Hopfield-type approach, we propose a unifying framework in which various chaotic models can be placed and compared, thus providing a means to classify existing HCNN models and study their dynamics in a new perspective. The HCNN models are implemented (via computer simulation) to solve the constraint satisfaction problem known as N-queen, such that their optimization performance can be compared in different parameter spaces measuring feasibility, efficiency, robustness and scalability. Further computational results on the network dynamics are investigated and examined in the light of the unifying framework. This allows an identification of characteristic chaotic dynamics crucial to effective optimization, as well as providing a guide to choosing model parameters.

For the self-organizing approach, we study the SONN with weight normalization (SONN-WN) for solving general 0-1 COP's in the perspective of NSD. We show how existing limitations in convergence and solution quality can be overcome by bifurcation dynamics due to the normalization process, especially with the presence of noise. The SONN-WN is studied both theoretically and experimentally by using the *N*-queen problem as an example to demonstrate the dependence of optimization performance on annealing schedules and other system parameters. In order to explain such dependence, an equilibrium model of the SONN-WN is derived, which unfolds the phenomenon of bifurcation and its crucial role in the convergence process. The dynamical systems view of the SONN-WN also reveals various nonlinear phenomena such as cascades of period-doubling bifurcations to chaos and a strange attractor. Combined with the action of added neuronal noise, we also demonstrate how the improved solution quality is achieved by reducing unwanted oscillatory behaviors of the network.

Since the focus of this thesis is on the theoretical foundation of the generation, characterization, and exploitation of NSD in neural networks for combinatorial optimization, in-depth performance evaluations e.g. comparisons with heuristics, intensive scalability study with large problem sizes, application to other COP's, etc., are beyond our scope. This research contributes to the literature an alternative view of these neural networks. It shows the role of existing NSD and presents opportunities for including richer dynamics for optimization. From the study presented here, it is indeed surprising to find various aspects of NSD relevant to *both* Hopfield and self-organizing type of neural networks for combinatorial optimization, despite their many differences.

In summary, the main contributions of the thesis are:

- proposal of a theoretical framework for unifying HCNN approaches to COP's, thereby suggesting a systematic classification of HCNN models according to their formulations and chaotic properties;
- optimization performance comparisons of various HCNN models in terms of feasibility, efficiency, robustness and scalability on representative parameter spaces (which also serves as a general guide for choosing parameters);
- implementation of various HCNN models to solve the *N*-queen problem, and characterization of their time-dependent dynamics via computer simulation;
- identification of promising new HCNN models with theoretical foundations informed by the unifying framework;
- implementation of a SONN with weight normalization (SONN-WN) to solve the *N*-queen problem, with feasibilities measured on the normalization parameter space;

- discovery of improved feasibility by 1). adding external noise to the nodes of the SONN-WN and 2). using a small neighborhood for nodes updating, with an explanation based on symmetry-breaking in node competitions;
- discovery of high feasibility bands in the normalization parameter space, revealing the sensitivity of solution quality on annealing schedules for the SONN-WN;
- derivation of an equilibrium model of the SONN-WN, revealing bifurcation dynamics hidden in the weight updating-normalization system;
- explanation of high feasibility bands in terms of the bifurcation dynamics of the SONN-WN, and providing a guide for choosing various system parameters;
- discovery of various nonlinear phenomena exhibited by the equilibrium model e.g. multi-dimensional cascades of period-doubling bifurcations to chaos, crisis, strange attractor, etc.

An outline of the organization of this thesis is as follows:

Chapter 2 presents an overview of the Hopfield neural network and the Hopfield-Tank method in combinatorial optimization, together with various approaches to overcome their limitations. As one of these approaches, a new breed of Hopfield-type neural networks incorporating chaotic dynamics are introduced.

Chapter 3 proposes a unified framework for Hopfield-type chaotic neural networks (HCNN's), and discusses its role as a theoretical foundation on which various HCNN models can be placed and compared. This framework has been published in *IEEE Transactions on Neural Networks*.

Chapter 4 demonstrates the implementation of HCNN models to solve the *N*-queen problem, followed by computational results measuring and comparing their optimization performance. Computational illustrations of chaotic dynamics arising from these models are also presented. We then discuss the significance of these results in the light of the proposed unified framework. This investigation has been published in *Neural Networks.*

The second half of this thesis is devoted to the self-organizing approach of neural networks for COP's, in particular self-organizing neural networks (SONN's) with nonlinear system dynamics (NSD):

Chapter 5 reviews various self-organizing approaches of neural networks for COP's, and their corresponding limitations. The self-organizing neural network with weight normalization (SONN-WN) is introduced as the main model for our NSD investigation.

Chapter 6 demonstrates the implementation of SONN-WN to solve the *N*-queen problem, followed by computational results measuring its optimization performance with respect to normalization annealing schemes. The dynamics of the network are illustrated computationally, and the effects of added neuronal noise and small neighborhood sizes are presented and discussed. The dependence of solution quality on annealing parameters is demonstrated. Part of this investigation has been published in the *Proceedings of the ICSC Symposia on Intelligent Systems and Applications (ISA2000).*

Chapter 7 presents a dynamical systems view of the SONN-WN via the derivation of an equilibrium model. Numerical investigations of the model are presented to illustrate the bifurcation properties of the weight updating-normalization process, together with other nonlinear phenomena. The significance of these results is then discussed. Also, the experimental observations of Chapter 6 are explained by these theoretical insights. A paper containing part of this study has been accepted for publication in the *Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks (IWANN2001).*

Chapter 8 closes the thesis with our main conclusions, and suggests directions for future research.

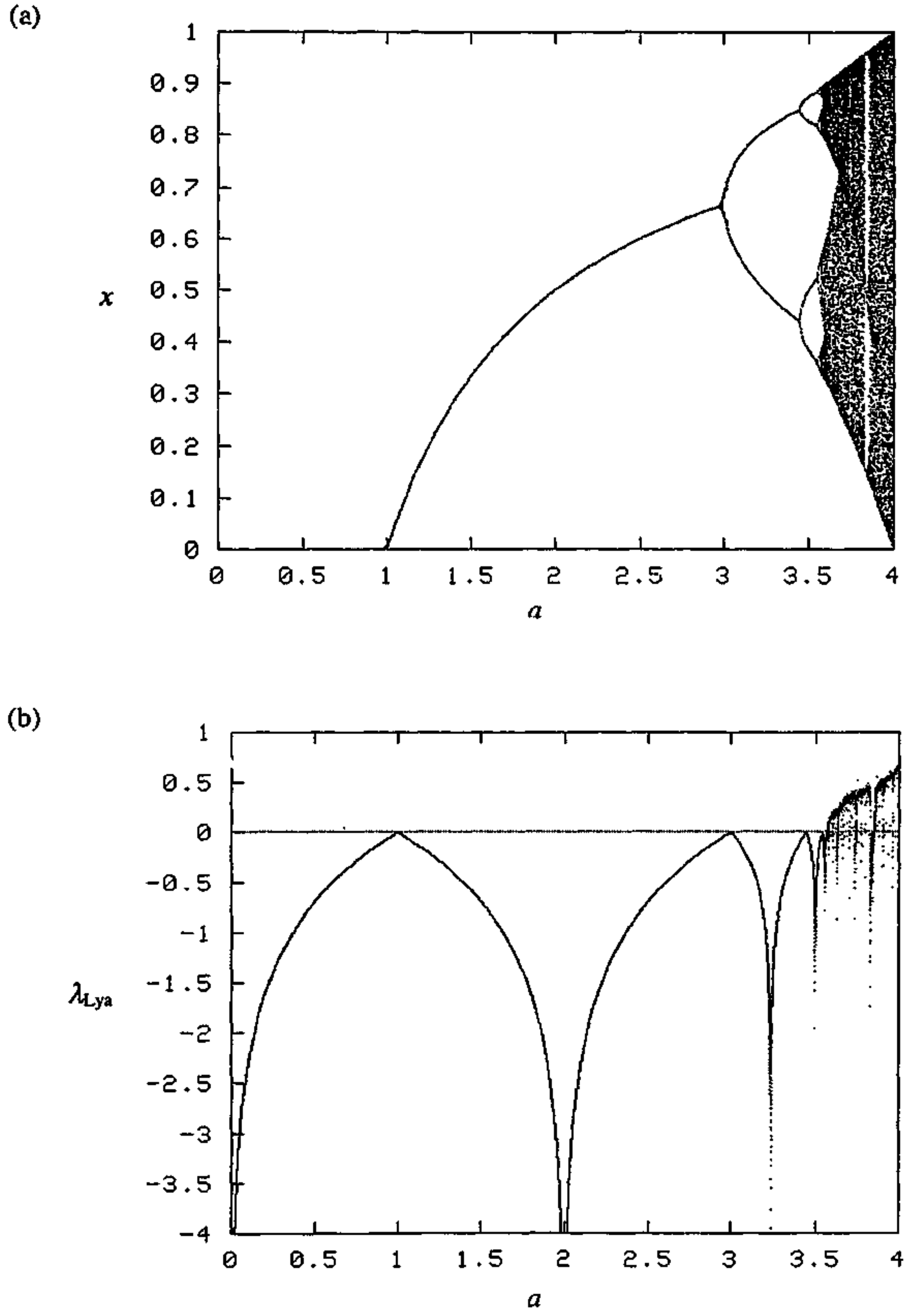


Figure 1.1: (a) Bifurcation diagram of the logistic map and (b) the corresponding Lyapunov exponent.

Chapter 2

Hopfield-type Neural Networks for Combinatorial Optimization

2.1 Introduction

In this chapter we discuss the Hopfield-type neural network approach for combinatorial optimization. Among the many variations and modifications of the original Hopfield network [52, 53], we survey the recent developments of Hopfield-type chaotic neural networks (HCNN's), which utilize chaotic dynamics for improved optimization performance. The nonlinear dynamics in a HCNN can be seen as a generalization to the local convergence dynamics in the original Hopfield neural network. In Section 2.2, we introduce the development of the Hopfield networks and the Hopfield-Tank (H-T) approach to combinatorial optimization, which represents a new era of solving difficult COP's with neural networks.

Realizing the collective computational power of the Hopfield network, Hopfield and Tank [54] developed the method of using the continuous Hopfield network to solve difficult COP's. The inherent energy minimization property of the network is exploited for 0-1 optimization from a continuous decision space. The method was applied to solve the TSP by computer simulation of the Hopfield model. Their mixed results show that the Hopfield network can indeed be used to solve COP's, but there are problems of poor solution quality and feasibility, especially when the problem size is increased. They encountered difficulties in choosing penalty parameters of the energy function that can lead to satisfactory solutions. An investigation by Wilson and Pawley [111] to apply the H-T approach to larger instances of the TSP further revealed serious weaknesses of the method. No parameter combination could be found in an attempt to solve a 64-city TSP, and even Hopfield and Tank's result on a 10-city TSP could not be repeated. Serious doubts were cast on the H-T approach as a practical means to solve COP's. In Section 2.3, the H-T approach is introduced, followed by a review of Wilson and Pawley's investigation in Section 2.3.1.

To overcome the limitations of the H-T approach, many improvements and modifications have been proposed. Since the solution feasibility is sensitively dependent on the penalty parameter combinations of the Hopfield energy function, one of the approaches extensively studied in the literature is on the treatment of these parameters for improved solution feasibility [49, 57, 28, 65, 23, 12, 68, 70]. To improve the solution quality, modifications to the energy function have also been suggested [20, 74, 103, 6, 39, 2]. Inspired by simulated annealing (SA) [61], a number of stochastic variations of the Hopfield network have also been proposed for further improvements in solution quality [1, 8, 96], together with related techniques in adjusting activation levels [54, 9, 101, 104, 105, 79]. All these improvements are outlined in Section 2.3.2.

Since the main limitation of the H-T approach is due to the existence of many local minima in the energy function and the simple steepest-descent dynamics of the Hopfield network, a novel class of neural networks with more general network dynamics has been proposed. A HCNN utilizes chaos to avoid being trapped in local minima, and also to facilitate a kind of chaotic searching process. In Section 2.4, we introduce various HCNN approaches for solving COP's.

By using chaotic neurons of refractory properties, Aihara *et al* [5, 4] constructed a HCNN based on the Hopfield network which exhibits chaotic dynamics. It was implemented to solve the TSP with interesting transitions of network states during the chaotic solution process [114]. Nozawa [75, 76] proposed another approach to HCNN by starting from the Euler discretization of the Hopfield model, and then adding a negative self-feedback term to generate chaos in the system. Experiments in solving the TSP suggested a relationship between a possible 'chaotic search' process and the chaotic itinerancy phenomenon in nonlinear dynamics. The dynamical basis of such a relationship was later clarified by Tokuda *et al* [99]. Details of these are given in Section 2.4.1.

Although the chaotic dynamics of a HCNN allow the network to escape from local minima, its chaotic nature means the lack of a converged solution state. This problem of combining chaotic search with solution convergence was solved by a method called chaotic simulated annealing (CSA) [24]. There are two approaches to CSA: the decaying self-feedback method [24] and the decaying time-step method [107]. Both methods involve an initial chaotic search phase, followed by convergence to a fixed point as the converged solution. Models employing CSA are discussed in Section 2.4.2.

All the HCNN's described above generate chaos by the inherent instability of the networks, but there are models where the chaos is not generated within the network itself. In these models, externally generated chaotic noise is added to the network for the purpose of escaping from local minima. Various chaotic noises have been used and their optimization performance was often compared to stochastic noises [47, 48, 15]. These models are further described in Section 2.4.3.

The purpose of this chapter is to review relevant techniques in the literature with Hopfield-type neural networks, before proposing a unifying framework of HCNN's in Chapter 3.

2.2 Hopfield neural networks

The Hopfield neural network was first proposed in 1982 by Hopfield [52], and has been considered to be a significant neural network architecture ever since. It was inspired by biological and physical systems to capture the dynamics of a system having a large number of simple and equivalent components. The Hopfield network consists of interconnected neurons, forming an architecture with strong backward coupling (or feedback) that performs asynchronous parallel processing. At the system level, the network exhibits collective computational abilities of a content-addressable memory (CAM), allowing a memory item to be retrieved by presenting sufficient partial information. Because of the network's general nature, many potential applications were anticipated, e.g. generalization, familiarity recognition, categorization, error correction, and time sequence retention.

The original model of the Hopfield network was discrete in nature, comprising of two-state threshold neurons [52]. For neuron *i*, the output x_i can attain values of either 0 or 1, which can be interpreted in the biological sense as not firing or firing respectively. The total input to a neuron *i* is given by

$$\text{Total input to neuron } i = y_i = \sum_{j=1, j \neq i}^{N} W_{ij} x_j + I_i \tag{2.1}$$

where W_{ij} is the strength or weight of the connection between neuron *i* and *j*, with the summation representing inputs from other neurons (*N* neurons in total), and I_i the external input. The output of a neuron is then given by

$$x_i = \begin{cases} 0 \\ 1 \end{cases} \quad if \quad \sum_{j=1, j \neq i}^{N} W_{ij} x_j + I_i \quad \begin{cases} \leq \theta_i \\ > \theta_i \end{cases} \tag{2.2}$$

where $\theta_i = 0$ is the threshold. The emergent computational behaviors of the network are described by the motion in a state space where (2.2) is used for updating. A *stochastic evolution* scheme is employed where each neuron is updated at random times. The use of such an asynchronous updating scheme is justified by its biological resemblance.

From a dynamical system perspective, the Hopfield network behaves as a content-addressable memory (CAM), where memory items are mapped to the fixed points in the state space. The retrieval of a memory item from incomplete information then corresponds to a state space flow where an initial point is attracted to the local fixed point. Such an abstract computational property is a result of the neurons' collective dynamics, and is the main feature of the Hopfield network that sets it apart from others like the McCulloch-Pitts model [72] and the discrete Perceptron [73].

The Hopfield network with continuous variables and responses was developed in 1984 [53]. The modification allows a closer resemblance to biological neural systems as well as analogue electronic systems, and still retains all the significant behaviors of the original discrete Hopfield network. Given a total net input (or internal state) y_i to neuron i, the output is given by $x_i = g(y_i)$, which is a continuous, monotonically increasing function bounded beiow by 0 and above by 1. A sigmoidal activation function is often used:

$$x_i = g_i(y_i) = \frac{1}{2}\left[1 + \tanh\left(\frac{y_i}{\varepsilon}\right)\right],$$

where ε is the gain control parameter.

To describe the evolution of the network in the context of a biological neural model, some specific parameters are required. C_i is the input capacitance of cell membranes for neuron i; R_i is the trans-membrane resistance; W_{ij}^{-1} is the finite impedance between output x_j and cell body of cell i; and I_i represents other fixed input current to neuron i. The resistance-capacitance (RC) charging equation governing the evolution can then be expressed as

$$C_i\left(\frac{dy_i}{dt}\right)=\sum_{j=1}^{N}W_{ij}x_j-\frac{y_i}{R_i}+I_i$$

$$y_i=g_i^{-1}(x_i)\ . \tag{2.3}$$

The same set of equations also describes a corresponding electronic implementation of the network, with appropriate re-definitions of various parameters.

The convergence and stability of the continuous Hopfield network can be examined by constructing the energy function

$$E=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}W_{ij}x_ix_j+\sum_{i=1}^{N}\frac{1}{R_i}\int_0^{x_i}g_i^{-1}(x)dx-\sum_{i=1}^{N}I_ix_i \tag{2.4}$$

For a symmetric matrix W, the time derivative of E is given by

$$\frac{dE}{dt}=-\sum_{i=1}^{N}\frac{dx_i}{dt}\left(\sum_{j=1}^{N}W_{ij}x_j-\frac{y_i}{R_i}+I_i\right)$$

$$=-\sum_{i=1}^{N}C_i\left(\frac{dx_i}{dt}\right)\left(\frac{dy_i}{dt}\right)$$

$$=-\sum_{i=1}^{N}C_ig_i^{-1'}(x_i)\left(\frac{dx_i}{dt}\right)^2.$$

For a monotonically increasing function $g_i^{-1}(x_i)$ and positive C_i, we have

$$\frac{dE}{dt} \leq 0, \qquad \text{and} \qquad \frac{dE}{dt} = 0 \Rightarrow \frac{dx_i}{dt} = 0 \quad \forall i. \tag{2.5}$$

Given E is bounded, (2.5) represents a steepest descent dynamics, where the state space motion is a convergence toward a local minimum of E. Thus (2.5) shows that the energy function E in (2.4) is a Lyapunov function and guarantees convergence of the continuous Hopfield network. This analysis can be extended to examine the stability and convergence of the discrete Hopfield network. For a high-gain limit of $g(y_i)$, i.e. $\varepsilon \rightarrow 0$, the activation function approximates the threshold function of $\theta_i = 0$ in (2.2), and the integral term in (2.4) becomes zero. It follows that the Lyapunov function for the discrete network is

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} x_i x_j - \sum_{i=1}^{N} I_i x_i \tag{2.6}$$

For the state space of the discrete case, the local minima of E lie at the corners of the unit hypercube. When we consider the high-gain limit of the continuous case, as long as W is symmetric and $g'(y_i)$ exists, then there is guaranteed convergence to corners of the hypercube and E is also minimized.

2.3 The Hopfield-Tank approach

In 1985, Hopfield and Tank pioneered an approach to use the Hopfield network for combinatorial optimization [54], and since then it has become the main approach to using neural networks to solve COP's. Inspired by the effectiveness of the biological neural network in solving difficult perceptual problems, the method makes use of the collective computational power of the Hopfield network to solve difficult COP's. It is based on the *continuous* Hopfield network because of its resemblance to the biological counterpart and the resultant speed in an analogue electronic circuit. With this approach, *discrete* problems are solved in a *continuous* decision space.

In the Hopfield and Tank (H-T) approach, the energy minimization property of the Hopfield network is exploited for general combinatorial optimization. The weights and external inputs are chosen such that the energy function E represents the function to be minimized. The constraints to be satisfied are also included in the energy formulation. As an example, consider a typical COP as follows [88]:

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) \\ \text{With constraints} & [\mathbf{R}]_1 \boldsymbol{x} = b_1 \\ & [\mathbf{R}]_2 \boldsymbol{x} = b_2 \\ & \ldots \\ & [\mathbf{R}]_{\mathrm{m}} \boldsymbol{x} = b_m \end{array}$$

where $[\mathbf{R}]_{\mathrm{i}}$ represents the ith row of a constraint matrix $\mathbf{R}$, and there are m constraints. Both $[\mathbf{R}]_{\mathrm{i}}$ and $\boldsymbol{x}$ are vectors of dimension n. According to the H-T approach, the energy formulation would be

$$E = A\,f(\boldsymbol{x}) + B_1\,([\mathbf{R}]_1 \boldsymbol{x} - b_1)^2 + B_2\,([\mathbf{R}]_2 \boldsymbol{x} - b_2)^2 + \ldots + B_m\,([\mathbf{R}]_{\mathrm{m}} \boldsymbol{x} - b_m)^2 .$$

A, $B_1, \ldots, B_m$ are penalty parameters representing the relative importance of various terms. For this COP, the minimization of $f(\boldsymbol{x})$ with satisfied constraints (all the constraint terms equal to zero) corresponds to the minimization of E. It is this correspondence that allows COP's to be mapped onto the Hopfield network for the minimization process. For the actual implementation of this approach to solve a specific COP, the weight matrix **W** and the external input *I* must be fixed according to the problem mapping. This is done by expanding the objective function $f(\mathbf{x})$, and then re-grouping various terms (quadratic and linear) in the same form as (2.6) for extracting various coefficients.

For the general solution process, it starts with an initial state of the Hopfield network. For an unbiased choice, the initial state values can be set to around 0.5 (the center of the unit hypercube) with perturbed variations among neurons. With asynchronous updating, the network evolves from the interior of the hypercube toward a stable fixed point sufficiently close to the vertex. The gain of the activation function should be high enough such that the converged state corresponds to an unambiguous 0-1 solution. The choice and balance of various penalty parameters in the energy function is a critical issue in the H-T approach, since it greatly affects the feasibility and quality of the solution. There is a trade-off problem in the choice of these parameters. If the constraint terms are emphasized (with greater values of B_i than A), then feasible solutions are often produced at the expense of optimality of the objective function. On the other hand if the objective function term is emphasized (by a larger A value), then non-zero constraint terms may result, which represents an infeasible solution.

To illustrate the computational power of the Hopfield network, the H-T approach was used by Hopfield and Tank [54] to solve the TSP that is well known to be NP-hard. The 10 and 30-city problems were solved by computer simulation of the Hopfield network. For the case of 10 cities, good results were achieved: 16 legitimate tours were obtained out of 20 random starting states, and about 50% of all the trials attained one of the 2 shortest paths. For the bigger problem size of 30 cities, the results were less satisfactory and fragmentary due to the demanding $O(n^3)$ simulation of differential equations involving 900 neurons. Also, Hopfield and Tank reported that it was difficult to choose parameter combinations that yielded good solutions for such a problem size.

2.3.1 An investigation by Wilson and Pawley

An investigation into the performance of the H-T approach for large problem sizes was published in 1988 by Wilson and Pawley [111]. The TSP was again used as the benchmark problem for their study. They began by applying the H-T approach to a 64-city problem, but the results were unsatisfactory. Primarily, they could not find any parameter combination that would yield a valid tour (or feasible solution). Surprisingly, in an attempt to reproduce Hopfield and Tank's results by solving the 10-city problem, they obtained different outcomes. An Euler approximation of a differential equation similar to (2.3) was used, and with the same parameters as Hopfield and Tank's. But out of 100 random starts, only 15 valid tours were obtained, with 45 local minima (invalid tours), and 40 didn't converge within 1000 iterations. Moreover, for 10 cities randomly generated, only 8% were valid. Even when valid tours were obtained, the solution quality was poor. The overall results were much worse than those reported by Hopfield and Tank despite solving the identical problem. There are some reasons that may explain these inconsistencies in part, e.g. discrepancies in simulation procedure, halting conditions, etc.

Modifications were attempted in Wilson and Pawley's paper toward solving the difficulties they encountered. These included variations in parameter selection, measures to prevent re-visiting the same city in succession, etc., but none significantly improved the percentage of valid tours obtained, or solution quality. Their investigation could not find any parameter combination that yielded consistent results as the problem size grows. Their paper concluded that the difficulty is inherent in the H-T approach, which makes it unreliable and not amenable to improvements.

2.3.2 Traditional approaches to overcome limitations

In general there are two approaches in the literature [88] to overcome the limitations of the H-T approach: by selecting optimal penalty parameters that yield good results, or by modifying the energy function. An experimental study on the choice of penalty parameters was carried out in 1988 by Hegde *et al* [49]. A parameter space was used to show parameter combinations resulting in feasible solutions to the TSP. It was found that as the problem size grows, parameter regions of feasible solutions become narrower, thus indicating poor scalability. They concluded that the result is due to the H-T formulation of the TSP, and suggested that the penalty parameters interact with each other. Also on the TSP, a theoretical study on penalty parameters selection by Kamgar-Parsi *et al* [57] examines the dynamic stability of valid tours. Eigenvalues are calculated for the Jacobian matrix of the H-T energy function, from which the penalty parameters corresponding to stable valid tours are determined. Agreements to these outcomes are also found from an extensive analysis by Davis [28], where only limited ranges of parameters that correspond to stable valid solutions are obtained.

Despite all the difficulties, the search for optimal penalty parameters continued. Lai and Coghill [65] used genetic algorithms to breed the penalty parameters with the goal of satisfying the validity requirements of the networks, while Cavalieri and Russo [23] applied a fuzzy logic approach for automatic parameter tuning. An exact method was proposed by Aourid *et al* [12], where the penalty parameters are treated as Lagrange multipliers in order to find the optimal parameter selection. The algorithm of augmented Lagrange multipliers from optimization theory has also been applied to the H-T approach [68, 70].

An alternative way to refine the H-T approach is to modify the energy function. A variety of suggestions could be found in the literature, e.g. using alternative forms of constraints expression, reducing the number of terms and parameters in the energy function, etc. [20, 74, 103]. Notably, the valid subspace approach of Aiyer *et al* [6] and the associated polytope concept [39] have succeeded in reducing the representation of constraints to a single term in the energy function, thus drastically reducing the number of penalty parameters. A technique of suppressing spurious states of the H-T approach has also been proposed [2].

Apart from efforts in enhancing the H-T approach, many variations of the Hopfield network have been proposed to improve the solution quality. Of particular relevance to this thesis are the stochastic methods, which supplement the deterministic Hopfield network with introduced randomness for escaping from local minima. The Boltzmann machine [1, 51] is based on the discrete Hopfield network, but with a stochastic activation function. The Boltzmann probability distribution is employed within the activation function to affect the binary output of the neurons. This allows uphill movements in the H-T energy landscape, thus enabling the escape from local minima. One disadvantage of this method is its large computational time, which renders it unsuitable for many real world applications. The Gaussian machine [7, 8] is based on the continuous Hopfield network, but with a Gaussian distributed noise added to the net input of each neuron. Both the Hopfield network and the Boltzmann machine can be considered as special cases of this model. By solving various COP's, the Gaussian machine was found to perform better than the Hopfield network and the Boltzmann machine [8]. Another variation on stochastic Hopfield networks is the Cauchy machine [93, 96]. It incorporates a parallel version of the fast simulated annealing algorithm proposed by Szu [94] that uses Cauchy noise to generate new search states. Compared to the Gaussian machine, the Cauchy noise used in this model allows *global* as well as local jumps for escaping from local minima. Furthermore, hybrid neurons having binary outputs from analog inputs are used, which allow the Cauchy machine to be reduced to the discrete Hopfield network in the high-gain limit. Related techniques in gain sharpening of the Hopfield networks have also been proposed [54, 9, 101]. The mean field approximation annealing approach is another widely adopted method for improving solution quality [79, 104, 105]. On the other hand, a way of escaping local minima by means of a *deterministic* "rock and roll" perturbation approach to the energy landscape has also been devised [69].

2.4 Hopfield-type chaotic neural network approaches

In this section, we review Hopfield-type chaotic neural networks (HCNN's) and their applications to solve COP's. These networks are based on the Hopfield network, but instead of the simple steepest descent dynamics toward a local fixed point, they exhibit more general nonlinear dynamics associated with chaos. There are various existing HCNN models for combinatorial optimization, and we examine their respective ways of incorporating chaos as part of their dynamics. Despite their differences, all these models have the common aim of utilizing chaos or associated dynamics to improve optimization performance for wide ranging COP's.

2.4.1 Chaotic neural networks and the 'chaotic search'

In 1990, Aihara *et al* [5, 4] proposed a neuron model capable of chaotic dynamics and includes some other neuron models as its special cases. It captures significant properties of biological neurons by having graded responses, relative refractoriness (self-feedback), and spatio-temporal summation of inputs. As an element of a neural network, the discrete-time chaotic neuron model is governed by the equations

$$\begin{aligned} y_i(t+1) = &\sum_{j=1}^{M} W_{ij} \sum_{r=0}^{t} k^r x_j(t-r) + \sum_{j=1}^{N} V_{ij} \sum_{r=0}^{t} k^r I_j(t-r) \\ &- \alpha \sum_{r=0}^{t} k^r x_i(t-r) - \theta_i \end{aligned} \tag{2.7}$$

$$x_i(t) = \frac{1}{1+e^{-y_i(t)/\varepsilon}} \tag{2.8}$$

where $y_i(t + 1)$ is the internal state of neuron i at time $t + 1$, $x_i(t + 1)$ is the output of neuron i, M is the number of neurons in the network, W_{ij} is the connection weight from neuron i to neuron j, V_{ij} is the weight on external input I_j of neuron i, N is the number of externally applied inputs, θ_i is the threshold of neuron i, k is the refractory decay parameter, and α is the refractory scaling parameter. The first three terms in (2.7) represent mutual interactions between neurons, the external force, and the neuron refractoriness respectively. The response characteristics of this neuron model were experimentally demonstrated to reveal cascades of period-doubling bifurcations, with positive Lyapunov exponent in chaotic regions. Moreover, the model is designed to be general, and the resulting chaotic neural network includes other conventional networks such as the McCulloch-Pitts networks [72] and the Hopfield network [53] as special cases.

The application of the above chaotic neural network in combinatorial optimization was demonstrated by Yamada *et al* [114] in 1993. The TSP is solved by using a simplified version of (2.7) [5, 4]:

$$y_i(t+1) = ky_i(t) - \alpha x_i(t) + \sum_{j=1}^{M} W_{ij} x_j(t) - \theta_i (1-k) . \tag{2.9}$$

Both the continuous activation function given by (2.8) and the following discrete version are used

$$x_i(t) = \begin{cases} 1 & \text{if} \quad y_i(t) \geq 0 \\ 0 & \quad y_i(t) < 0. \end{cases}$$

The 10-city problem is solved with different values of the refractory scaling parameter α. The results show that global minimum can be obtained for both continuous and discrete activation functions, and that the solving ability (best routes obtained out of 1000 initial states) varies with relative refractoriness α. Time series depicting the chaotic behaviors of the network are also presented.

Instead of building a chaotic neural network from chaotic neurons, Nozawa [75, 76] proposed a HCNN by using the Euler's discretization of Hopfield's continuous model (2.3) with a negative self-feedback connection $W_{ii} < 0$:

$$y_i(n+1) = (1 - \frac{\Delta t}{\tau}) y_i(n) + \Delta t W_{ii} g(y_i(n)) + \Delta t \sum_{j \neq i}^{M} W_{ij} g(y_i(n)) + \Delta t I_i . \quad (2.10)$$

$y_i(n)$ is the input of neuron i at the discrete time n, Δt is the time-step, τ is the time constant parameter ($\tau > 0$), g is the sigmoidal activation function, and M is the number of neurons in the network. It is well known that nonlinear self-feedback mechanisms can often generate chaos in one-dimensional dynamical systems (see Section 1.3). Indeed, the derived model is equivalent to the simplest version of chaotic neural network proposed by Aihara *et al* [5]. The HCNN model is then cast into the form of a globally coupled map (GCM) [58, 59] to demonstrate how various nonlinear phenomena such as chaos and periodicity arise as a result of different nonlinear maps. This GCM formulation of the HCNN is then applied to solve the TSP with 10 cities. Within 1000 iterative steps, the model is able to achieve the shortest route from 94% of the random starts. It was observed that during the solution process, the firing pattern of the network chaotically *itinerates* between both global and local minimum solutions, and that it stays mostly in the global minimum solutions. This *chaotic itinerancy* was suggested to be reflecting a kind of 'chaotic search' whose detailed dynamics was unknown. A concrete explanation was not given as to why the observed nonlinear system dynamics are capable of improving the solution quality. However, it was suggested that clusters are formed in the internal state of the GCM, and it is the interaction among these clusters that contributes to the improvement.

In 1997, Isao Tokuda and his colleagues adopted Nozawa's chaotic neural networks [75, 76] to study the dynamical basis of chaotic itinerancy as a searching mechanism for global/local minimum solutions, with an emphasis on global bifurcation structures [99]. The chaotic neural network was then implemented to solve TSP's with 5 and 10 cities for the purpose of investigating the solution process. In order to trace the solution states of the network, a coding of attractors was introduced to map every attractor of the dynamical system to a possible TSP solution. By decreasing the bifurcation parameter $r = 1-(\Delta t/\tau)$, local minimum solutions were observed to undergo period-doubling bifurcation routes to chaos, followed by a size increase in the localized chaotic attractors, which then merge into a global attractor via crises. Specifically, the dynamical phenomenon of crisis-induced intermittent switching [41, 42] was found to be the dynamical basis of the previously observed chaotic itinerancy among minima. Global/local minimum solution states identified as *ruins of localized chaotic attractors* were observed to switch from one to another as the network is in the solution process. This occurs when attractors are merged together via crises, and represents the 'chaotic search'.

The result of solving the 5-city TSP with the above dynamics reveals that the probability of transition from a local minimum to a global minimum solution is higher than the reverse, thus demonstrating the efficiency of the 'chaotic search'. It was remarked that the efficiency of the 'chaotic search' depends on the complex linkage structure of previous localized chaotic attractors. Thus an efficient search only occurs when there are a large number of dynamical paths leading to a global minimum. The study of the bifurcation procedure above also serves as a guide for choosing the bifurcation parameter r such that a 'chaotic search' occurs.

2.4.2 Models with chaotic simulated annealing

It is clear from the HCNN models [5, 75, 76, 114] described so far that when a COP is being solved, global/local minima are accessible by chaotic dynamics of the network [75, 76, 99]. The problem of the solution state being trapped at a local minimum, which is the main drawback of the H-T approach [54], is essentially solved by the itinerant nature of the HCNN's dynamical searching process. However, the issue of arriving at a *converged* solution with optimal/near-optimal quality from the chaotic searching process had not been dealt with. This was addressed by the proposal of chaotic simulated annealing (CSA) by Chen and Aihara in 1995 [24]. This scheme is analogous to Kirkpatrick *et al*'s simulated annealing (SA) [61], but the solution searching process is deterministically chaotic rather than stochastic. The main feature of CSA is that it allows a neural network to be *transiently chaotic*, meaning the chaotic searching phase is temporary, and the algorithm terminates at a converged stable state.

In their paper [24], a transiently chaotic neural network (TCNN) model based on Aihara *et al*'s chaotic neural network [5] was proposed:

$$y_i(t+1) = ky_i(t) + \alpha\left(\sum_{j=1,j\neq i}^{N} W_{ij}x_j(t) + I_i\right) - z_i(t)\left(x_i(t) - I_0\right) \qquad (2.11)$$

$$z_i(t+1) = (1-\beta)z_i(t), \qquad (2.12)$$

where $i = 1,\ldots, N$, $y_i(t)$ is the internal state of neuron i, $x_i(t) = g(y_i(t))$ is the output of neuron i, g is the sigmoidal function in (2.8), $\alpha > 0$ is a scaling parameter, β is the damping factor of $z_i(t)$ with $0 \leq \beta \leq 1$, k is the nerve membrane damping factor with $0 \leq k \leq 1$, $z_i(t) \geq 0$ is the self-feedback connection weight (replacing the role of W_{ii}), I_0 is a positive parameter, and W_{ij} is the connection weight matrix element defined by

$$\sum_{j=1,j\neq i}^{N} W_{ij}x_j + I_i = -\frac{\partial E}{\partial x_i}$$

with $W_{ij} = W_{ji}$ and $W_{ii} = 0$, and E is the Hopfield energy function in (2.6). The last term in (2.11) represents the self-feedback of the network with $z_i(t)$ controlling its strength. For simplicity, the same $z(t)$ is used for all neurons. Analogous to the temperature parameter in SA, $z(t)$ acts as the annealing parameter of the system. Together with (2.12), the self-feedback strength of the system is exponentially decayed, and the network passes through a reversed period-doubling bifurcation from chaos to a stable fixed point. As $z(t) \to 0$ for increasing time, the term $-z(x_i - I_0)$ also tends to zero, and the TCNN asymptotically approaches a Hopfield-like network. The parameters that require tuning in this model are: ε, α, β, k, $z(0)$ and I_0.

By applying the TCNN with CSA to solve TSP's with 4, 10, 48 cities, general nonlinear dynamics were observed. It was shown that there are two phases of the solution search: a chaotic searching phase for large $z(t)$, followed by a stable convergent phase for small $z(t)$. The parameter α was also found to carry the function of balancing the nonlinear refractory dynamics and the steepest descent dynamics of the network, e.g. transient chaos could be suppressed by using a large α. The best result obtained with the TCNN on the 10-city problem yields optimal solutions to 100% of all 5000 random starts, which is significantly higher than that of Hopfield and Tank [54]. For comparison, a 92% rate was obtained by using traditional SA. The high efficiency of the TCNN was suggested in the paper as due to the solution search in a possibly fractal subspace, while SA searches all possible states by varying probability distributions. Their experimental results also suggested a possible relationship between annealing speed (controlled by β) and solution quality, with slower annealing (smaller β's) leading to more optimal solutions out of the random starts. However, a theoretical study of the *hierarchical merging of chaotic attractors* shows that an infinitely slow annealing does not necessarily lead to the global minimum [98], and some adaptive and learning schemes have been proposed for fast convergence to good solutions [98, 100]

Issues of asymptotical stability and the existence of chaos in both the TCNN and general discrete-time recurrent neural networks were later studied by Chen and Aihara [25]. They proved that the networks only have one bounded fixed point when the absolute value of the self-feedback connection weight (z_i in (2.11)) is sufficiently large, and derived sufficient conditions on asymptotical stability toward the fixed point for the symmetrical versions of the networks. For the formulation in (2.11) with sigmoidal activation function (2.8), $W_{ij} = W_{ji}$, and an asynchronous updating, asymptotic convergence to a fixed point is guaranteed if one of the following is satisfied:

1. $1/3 \geq k \geq 0,\ 4(1-k)\varepsilon > -\min_{i=1}^{N}\{-z_i(t)\}$, or
2. $1 \geq k \geq 1/3,\ 8k\varepsilon > -\min_{i=1}^{N}\{-z_i(t)\}$, or
3. $k > 1,\ 8\varepsilon > -\min_{i=1}^{N}\{-z_i(t)\}$. (2.13)

This stability condition determines how small the refractory parameter $z_i(t)$ should be for convergence. For synchronous updating, the corresponding stability condition is:

1. $1/3 \geq k \geq 0,\ 4(1-k)\varepsilon > -\lambda_{\min}$, or
2. $1 \geq k \geq 1/3,\ 8k\varepsilon > -\lambda_{\min}$, or
3. $k > 1,\ 8\varepsilon > -\lambda_{\min}$, (2.14)

where $\lambda_{\min}$ is the smallest eigenvalue of the matrix αW.

In 1998, Wang and Smith [107] proposed an alternate approach to CSA with fewer system parameters, and hence less effort for parameter tuning. The governing equation of the network is given by

$$y_i(t+\Delta t) = (1-\frac{\Delta t}{\tau})y_i(t) + \Delta t\left(\sum_{j=1}^{N} W_{ij}x_j(t) + I_i\right) \tag{2.15}$$

$$x_i(t) = g_i(y_i(t)) \quad \in [0,1], \tag{2.16}$$

where $x_i(t)$ is the output of neuron i, $y_i(t)$ is the corresponding internal state, and τ is a positive parameter. The activation function g in (2.16) can be any monotonically increasing function, and the sigmoidal function (2.8) is often used. In fact, (2.15) is the Euler's discretization of the continuous Hopfield model (2.3) with the discretized time-step Δt. As $\Delta t \to 0$, (2.15) is equivalent to the Hopfield model. Moreover, (2.15) can be transformed to Chen and Aihara's TCNN [24, 25] if $\Delta t = 1$. This relationship suggests that Tokuda *et al*'s [98, 99, 100] global attractor scenario of 'chaotic search' may also be applicable to this CSA scheme.

The chaotic simulated annealing in this Wang-Smith model is achieved by a reversed bifurcation process controlled by Δt. Initially Δt is chosen to be sufficiently large to induce chaos in the network. As Δt is gradually reduced, for example by an exponential decaying rule similar to (2.12), a transition from chaos to a fixed point takes places. The convergence to a fixed point occurs when Δt becomes sufficiently small, and for asynchronous updating the following stability condition applies:

1. $0 < \Delta t \le \tau$,
2. $\Delta t < \dfrac{2\tau}{1-\mu_{\max} W_{ii}}$, and
3. $W_{ii} \le 0$ (2.17)

where **W** is a symmetric matrix and $\mu_{\max}$ is the maximum slope of the activation function. For synchronous updating, w_{ii} is replaced by the minimum eigenvalue of **W**. When compared to the stability conditions of the TCNN in (2.13) and (2.14) where the sigmoidal activation function is used, this model only requires an arbitrarily increasing function for the activation function. This would be an advantage for hardware implementation. The parameters that require tuning in this model are: ε, τ, $\Delta t(0)$ and β. Hence fewer than the Chen-Aihara model.

2.4.3 Models with chaotic noise

In the previous sections, we have presented HCNN models with chaotic dynamics embedded within the network by the use of a particular term or a system parameter. Here we outline another approach to HCNN for combinatorial optimization: adding externally generated chaotic noise or chaotic time series into the Hopfield model. The objective of this approach is to improve solution quality by using chaotic noise to escape from local minima. The main difference between chaotic noise and other random noises is that it possesses short-term autocorrelation only, and the long-term behavior is deterministically stochastic. The study of this approach often prompts interesting comparisons with the previously mentioned HCNN models with internally induced chaos, in the aspects of optimization performance, functional role of chaos, theoretical relationships, etc.

Based on phenomena of the interactions between correlated noise and various physical and chemical systems, Hayakawa *et al* [48] proposed a chaotic neural network by introducing chaotic noise to the Hopfield model. An Euler discretized model (2.15) with $\tau = 1$ and $\Delta t = 1$ is used, with the usual sigmoidal activation function. The noise is added to the internal state of the neurons:

$$y_i(t) \leftarrow y_i(t) + A_p \eta_i(t) \tag{2.18}$$

where A_p controls the amplitude of the noise and $\eta_i(t)$ is the normalized noise for neuron i. Three kinds of noises are used: random noise, logistic map time series, and shuffled logistic map time series. The logistic map time series is generated from the logistic map

$$\eta_i(t+1) = a\eta_i(t)(1-\eta_i(t)) \tag{2.19}$$

where a is the bifurcation parameter. The shuffled logistic map is obtained by randomly shuffling the time series generated from (2.19).

The model was applied to the TSP with 10 cities to measure for how many iterations the optimal state is attained during the second half of each run. The best result was obtained with the logistic map noise where 100% of the 1000 steps are at the global minimum, while the corresponding result for random noise is 49%. This performance varies with a, suggesting a relationship between noise structures and the search for global minima. Interestingly, the use of the shuffled logistic map noise dramatically worsens the result for various values of a. It was unclear what aspects of the chaotic noise are responsible for the enhanced performance, but it was suggested that the characteristic autocorrelation in the chaotic noise plays an important role.

In order to study the effects of noise autocorrelation on the network's ability to attain global/local minima, Asai *et al* [15] constructed an autoregressive time series of the form

$$\xi(t) = \phi\xi(t-1) + \eta(t)$$

where $\xi(t)$ is the time series added to the internal state of the network after normalization, ϕ controls the degree of autocorrelation, and $\eta(t)$ is any normalized noise. In their experiments, four kinds of noise were used: random noise, logistic map noise (2.19), and two time series from the chaotic Chua's circuit [71]. By solving the 6-city TSP, the frequency of feasible solutions obtained during iterations was measured. It was found that a larger ϕ representing a stronger autocorrelation generally gives better results in attaining the global/local minima. However, it is unclear from the study whether chaotic noise gives better results than random noise.

A more intensive experimental study was carried out by Hasegawa *et al* [47] to compare the optimization performance due to the logistic map noise and various stochastic processes. The externally generated noise with adjustable amplitude is added to the internal state of the neurons. TSP's with 10 and 20 cities were solved, and the optimization performance was measured as the percentage of runs, out of 100 random starts, that hit the optimum solution at least once within 1000 iterations. Note that this measure is different from some of the other approaches mentioned above. For both the 10 and 20-city TSP, the logistic map noise (2.19) with $a = 3.95$ (i.e. chaotic, see Section 1.3) gives high solving abilities of 100% and 94% respectively. Both the uniformly distributed random noise and Gaussian distributed random noise score similarly, yielding ~ 88% for the 10-city problem and a low 1% for 20-cities. This shows the superior scalability of the chaotic noise in delivering high solving ability when compared to stochastic noise. The use of $1/f^{\alpha}$ stochastic noise with $0 \leq \alpha \leq 2.5$ on the 10-city problem also reveals that the logistic map yields consistently higher performance. Surrogate time series [97] preserving different statistics of the logistic map noise such as power spectrum, empirical histogram, etc. were used to seek out the statistical properties responsible for high solving abilities. Among the surrogate noises used to solve the TSP's, those preserving autocorrelation exactly or approximately perform similarly to the original logistic map noise. It was concluded that temporal structures in the form of autocorrelation within a chaotic noise are the crucial elements responsible for high solving abilities.

2.5 Concluding remarks

In this chapter we have reviewed the developments of Hopfield-type neural networks for solving COP's. From a dynamical system perspective, it is the local convergence property of the Hopfield network that gives rise to its energy minimization behavior, and consequently underlies the H-T approach as a new tool for combinatorial optimization. However, its poor performance in solving nontrivial problems as revealed by Wilson and Pawley [111] has rendered the H-T approach an impractical tool for real-world applications. This has prompted various improvements and modifications to both the H-T approach and the Hopfield network, targeting the problem of penalty parameter selection and convergence toward local minima solutions. A new direction has emerged recently to overcome these limitations by generalizing the dynamics of the Hopfield network. The development of Hopfield-type neural networks with chaotic dynamics alleviates the problem of strict convergence to the nearest local minima, and provides a global search dynamics that can escape from local minima.

Various HCNN's have been reviewed in this chapter, and all of them use chaos as a dynamical enhancement for the optimization process. Three approaches to chaotic neural networks have been discussed: using chaotic neurons with strong self-feedback

(Chen-Aihara model); adding a negative self-feedback term to the Euler-discretized Hopfield model (Nozawa model); and using a sufficiently large time-step to the Euler-discretized Hopfield model (Wang-Smith model). The HCNN models obtained from these approaches are all inter-related, and it is expected that the same global attractor scenario underlies the search mechanism of these models. The development of chaotic simulated annealing schemes has allowed the networks to combine chaotic search with convergence property, thus making HCNN a more practical tool for optimization applications. The introduction of externally generated chaotic noise into the Hopfield network has also yielded improvements in solution quality, with autocorrelation in the chaotic noise playing an important part.

From this chapter, we can see there are currently two major classes of HCNN's: one is the *internal* approach, where chaos is generated within the network controlled by some bifurcation parameters; the other is the *external* approach, where an externally generated chaotic signal is added to the network as perturbation. In the next chapter, we propose a theoretical framework that unifies these two approaches, thereby providing a more systematic and insightful treatment of HCNN's.

Chapter 3

A Unified Framework for Hopfield-type Chaotic Neural Networks

3.1 Introduction

In this chapter, we provide an organized way to study the chaotic structures and their effects in solving COP's with HCNN's. A unifying framework is proposed to serve as a basis where the existing HCNN models can be placed and compared. From the last chapter, it can be seen that most researchers are concerned with investigating how chaos can benefit neural network performance, and how the performance is affected by problem size and complexity. The purpose of this chapter is to introduce a well-defined framework as a theoretical foundation for comparing different chaotic approaches. Only when we establish a common framework can the chaotic structures and their role in the optimization of each approach be understood.

The key of the proposed framework is the introduction of an extra energy term into the computational energy of the Hopfield model, which takes on different forms for different HCNN models, and modifies the original Hopfield energy landscape in various manners. Three HCNN models, namely the Chen-Aihara model with self-feedback chaotic simulated annealing (CSA) [24], the Wang-Smith model with time-step CSA [107], and the chaotic noise model [47, 15, 48], are chosen as examples to show how they can be classified and compared within the proposed framework.

3.2 The Hopfield network and the modified energy

In this section, a unified framework for HCNN's is described. Since all HCNN models for solving COP's being studied so far are related to the Hopfield network, the Hopfield computational energy formulation [54] thus serves as a relevant starting point for constructing a framework for these models. To illustrate the methodology of the framework, we first outline how the Hopfield network's equation of motion can be obtained from the energy landscape. From (2.4), the Hopfield computational energy E_{Hop} associated with the network state $\boldsymbol{x}$ is restated here as

$$E_{Hop}(t) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}x_i(t)x_j(t) - \sum_{i=1}^{N} I_i x_i(t) + \frac{1}{\tau}\sum_{i=1}^{N}\int_0^{x_i(t)} g^{-1}(\zeta)d\zeta \ . \quad (3.1)$$

To simplify notations, $x_i = x_i(t)$ and $y_i = y_i(t)$, with $x_i = g(y_i)$, where g is the sigmoidal function (2.8). The ability of the Hopfield network to solve COP's relies on its steepest descent dynamics and guaranteed convergence to local minima of the energy landscape. If **W** is symmetric and $g^{-1}(x_i)$ is monotonously increasing, then the relationship that guarantees $\mathrm{d}E/\mathrm{d}t \le 0$, and $\mathrm{d}E/\mathrm{d}t = 0 \Leftrightarrow \mathrm{d}y_i/\mathrm{d}t = 0$, $\forall i$ is given by [54]

$$dy_i / dt = -\partial E / \partial x_i . \tag{3.2}$$

The equation of motion of the Hopfield model can then be obtained by applying (3.2) to (3.1):

$$\frac{dy_i}{dt} = -\frac{y_i}{\tau} + \sum_{j=1}^{N} W_{ij} x_j + I_i . \tag{3.3}$$

In order to view existing HCNN models under a unified framework, the Hopfield energy landscape is now modified to include nonlinear dynamics in the resulting equation of motion, and its general form is given by

$$\begin{aligned} E &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} x_i x_j - \sum_{i=1}^{N} I_i x_i + \frac{1}{\tau}\sum_{i=1}^{N}\int_0^{x_i} g^{-1}(\zeta) d\zeta + H(x_i, W_{ij}, I_i) \\ &= E_{Hop} + H \end{aligned} \tag{3.4}$$

where the function H is an additional term that modifies the energy landscape. Clearly $H = 0$ for the Hopfield network. To obtain the equation of motion for any particular HCNN, we can apply the steepest descent relationship (3.2) onto (3.4). Both types of HCNN's, with internal or external chaos, can be reproduced from this formulation by choosing a different H. In the following sections, three HCNN models are described and derived from this framework, namely, Chen & Aihara's decaying self-coupling CSA, Wang & Smith's decaying time-step CSA, and the chaotic noise model.

3.3 CSA with decaying self-coupling

In this section, we show how the Chen-Aihara model of CSA with decaying self-coupling can be obtained from the unified framework approach described above. Suppose $H = H_{\mathrm{CA}}$ is chosen as the energy modifier in (3.4), where

$$H_{CA} = \frac{\lambda(t)}{2}\sum_{i=1}^{N} x_i(x_i - 1). \tag{3.5}$$

The positive parameter $\lambda(t)$ decays monotonically in time. By (3.2), (3.4) & (3.5), we have

$$\frac{dy_i}{dt} = -\frac{y_i}{\tau} + \sum_{j=1}^{N} W_{ij}x_j + I_i - \lambda(x_i - \frac{1}{2}). \tag{3.6}$$

Assume **W** is symmetric, and by using Euler discretization,

$$y_i(t+1) = (1 - \frac{\Delta t}{\tau})y_i(t) + \Delta t\left(\sum_{j=1}^{N} W_{ij}x_j(t) + I_i\right) - \Delta t\lambda(t)\left(x_i(t) - \frac{1}{2}\right) \tag{3.7}$$

If we assume the following relationships,

$$1 - \frac{\Delta t}{\tau} = k,\ \ \Delta t = \alpha,\ \ \Delta t\lambda(t) = z(t),\ \ I_0 = \frac{1}{2} \tag{3.8}$$

then it follows that

$$y_i(t+1) = ky_i(t) + \alpha\left(\sum_{j=1}^{N} W_{ij}x_j(t) + I_i\right) - z(t)\big(x_i(t) - I_0\big). \tag{3.9}$$

Moreover, if the following decaying scheme is chosen,

$$z(t+1)=(1-\beta)z(t) \tag{3.10}$$

then (3.9) and (3.10) (or other decaying schemes) together with $x_i = g(y_i)$ form the Chen-Aihara model of CSA with decaying self-coupling [24], where $z(t) \geq 0$ is the self-feedback connection weight (or self-coupling), β ($0 \leq \beta \leq 1$) is the damping factor, and I_0 is a positive parameter, as defined in Chapter 2. Comparing (3.9) to (2.11), it should be noted that the corresponding stability conditions of the Chen-Aihara model in the form of (3.9) are:

1. $1/3 \geq k \geq 0, 4(1-k)\varepsilon > -\min_{i=1}^{N}\{\alpha W_{ii} - z_i(t)\}$, or

2. $1 \geq k \geq 1/3, 8k\varepsilon > -\min_{i=1}^{N}\{\alpha W_{ii} - z_i(t)\}$, or

3. $k > 1, 8\varepsilon > -\min_{i=1}^{N}\{\alpha W_{ii} - z_i(t)\}$ (3.11)

with asynchronous updating. (c.f. (2.13)).

Although the particular expression of H_{CA} chosen here is by no means unique, its required form solely depends on the given HCNN model. In later sections, we show that different means of chaos generation require different forms of H. For the Chen-Aihara CSA model, it takes on a quadratic form as expressed in (3.5). H_{CA} is convex, and hence drives x towards the interior of the hypercube. This driving force is diminished as $H_{\mathrm{CA}} \to 0$ when $\lambda(t) \to 0$. This process is in fact the hysteretic annealing studied by Eberhardt *et al* [31] and Gee [39], which is aimed at improving 0-1 solution qualities. Here, H_{CA} modifies the original Hopfield energy landscape to accommodate transient chaos via $\lambda(t)$, which is proportional to $z(t)$ from (3.8). This correspondence between $\lambda(t)$ and $z(t)$ implies that the size of the energy modifier determines the dynamical stability of the Chen-Aihara model, since $\lambda(t)$ determines the size of H_{CA}, and $z(t)$ is the main parameter for chaos as discussed in Chapter 2. Also, it can be seen from (3.5) that H_{CA} represents the addition of many logistic maps to the Hopfield energy function, suggesting chaotic properties of the maps as outlined in Section 1.3. Thus the chaotic behaviors of the Chen-Aihara CSA model can be understood in terms of an energy landscape deformation with specific form and sizes.

3.4 CSA with decaying time-step

In this section, we turn to the Wang-Smith model of CSA with decaying time-step [107]. Instead of the self-feedback strength $z(t)$ that induces chaos in the Chen-Aihara model, the time-step Δt is used as the bifurcation parameter here. Such a time-step is a result of the Euler discretization applied to the continuous Hopfield network. We again use the unified framework to obtain the model, followed by insights obtained from such an approach.

Assuming the form of the energy modifier $H = H_{\mathrm{WS}}$ is chosen as follows:

$$H_{WS} = (1-\gamma^t)\left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}x_i x_j + \sum_{i=1}^{N} I_i x_i - \frac{1}{\tau}\sum_{i=1}^{N}\int_0^{x_i} g^{-1}(\zeta)d\zeta\right) \qquad (3.12)$$

where $0 < \gamma < 1$, and γ^t is γ to the power of t. By (3.2), (3.4) and (3.12),

$$\frac{dy_i}{dt} = \sum_{j=1}^{N} W_{ij}x_j(t) + I_i - \frac{y_i(t)}{\tau} + (\gamma^t - 1)\left(\sum_{j=1}^{N} W_{ij}x_j(t) + I_i - \frac{y_i(t)}{\tau}\right). \qquad (3.13)$$

Using Euler discretization,

$$\frac{y_i(t+1)-y_i(t)}{\Delta t} = -\frac{\gamma' y_i(t)}{\tau} + \gamma'\left(\sum_{j=1}^{N} W_{ij}x_j(t)+I_i\right) \tag{3.14}$$

or

$$y_i(t+1) = \left(1-\frac{\gamma'\Delta t}{\tau}\right)y_i(t) + \gamma'\Delta t\left(\sum_{j=1}^{N} W_{ij}x_j(t)+I_i\right) \tag{3.15}$$

Let $\gamma = 1 - \beta$ and assume a monotonically decreasing time-step, we put

$$\Delta t(t+1) = (1-\beta)\Delta t(t). \tag{3.16}$$

It follows that

$$y_i(t+1) = \left(1-\frac{\Delta t(t)}{\tau}\right)y_i(t) + \Delta t(t)\left(\sum_{j=1}^{N} W_{ij}x_j(t)+I_i\right). \tag{3.17}$$

The equations (3.16) and (3.17) together with $x_i = g(y_i)$ form the Wang-Smith model of CSA with decaying time-step, which is originally derived from the Euler discretized continuous Hopfield network [107] as discussed in Chapter 2. According to the original formulation, the network becomes chaotic when a large initial time-step $\Delta t(0)$ is chosen, which starts the chaotic searching phase. When Δt gradually becomes smaller, a reverse bifurcation process occurs, which is followed by convergence toward a stable fixed point.

In fact, the energy modifier as expressed in (3.12) can be written as

$$H_{WS} = (\beta' - 1)E_{Hop} \tag{3.18}$$

and

$$E = E_{Hop} + H_{WS}$$

or
$$E = \beta' E_{Hop} \tag{3.19}$$

The net effect of (3.19) is a time-dependent flattening of the energy landscape. As $t \to \infty$, the energy landscape becomes almost flat and the steepest descent updating rule (3.2) suggests the network state vector would remain almost at the same position. This agrees with the difference equation (3.15) when Δt becomes very small. This is different from the Chen-Aihara CSA model where the energy modifier H_{CA} is the only decaying element of the annealing process. In comparison, H_{WS} has a scaling effect to E_{Hop} without changing its shape, while H_{CA} contributes to E_{Hop} and causes an additive deformation to the energy landscape. However, the specification of the combinatorial problem is unchanged, as the asymptotic shape of the energy landscape is the same as E_{Hop}.

3.5 Models with chaotic noise

In this section we consider HCNN models with chaotic noise, where chaotic properties are generated externally and then added into the network. By using the unified framework, we can derive the difference equation of the network dynamics. The energy modifier $H = H_\eta$ is chosen for this model as

$$H_\eta = -A_\eta \sum_{i=1}^{N} \eta_i(t) x_i(t) \tag{3.20}$$

where $\eta_i(t)$ is a normalized chaotic time series added to neuron i, with the amplitude factor A_η. For example, the iterative logistic map with bifurcation parameter a as in (2.19) can be used:

$$\eta_i(t+1) = a\eta_i(t)(1-\eta_i(t)) \tag{3.21}$$

which can be normalized by setting

$$\eta_i(t) \leftarrow \left(\frac{\eta_i(t) - \langle \eta_i \rangle}{\sigma_\eta} \right) \tag{3.22}$$

where $<\eta_i>$ is the mean of $\eta_i(t)$ over all t in the data set, σ_η is the corresponding standard deviation. Other kinds of chaotic time series have been used in a similar manner [15, 47, 48]. By using (3.2), (3.4) and (3.20) with Euler discretization, we obtain the difference equation for the noise model:

$$y_i(t+1) = \left(1 - \frac{\Delta t}{\tau}\right) y_i(t) + \Delta t \left(\sum_{j=1}^{N} W_{ij} x_j(t) + I_i + A_\eta \eta_i(t) \right) \tag{3.23}$$

where $x_i = g(y_i)$. This is basically the original Hopfield model but with the external noise term $A_\eta \eta_i(t)$ added.

From (3.20), H_η represents a chaotic, time-varying perturbation to the Hopfield energy landscape E_{Hop}. Compared to the two *internal* models described previously which have quadratic forms of H, namely H_{CA} and H_{WS}, this *external* model has a linear form of H. Thus the form of the modifier can determine whether chaos is generated within or outside the networks (or no chaos when $H = 0$). This represents a new way for classifying HCNN models with respect to their *dynamics* by inspecting the *form* of the energy modifiers. Furthermore, this relationship between form and dynamics can be used to identify new models of HCNN. For example, by constructing an energy modifier with both quadratic and linear terms, a new type of neural networks encompassing *both* internal and external chaotic dynamics can be devised.

Indeed, a new type of neural network combining both internal chaos and external noise has been recently proposed [108, 109].

In terms of improving optimization performance, the usefulness of chaotic perturbations could be due to the fact that a chaotic time series has the peculiar property of never repeating itself exactly, thus H_η acts as a dynamically rich perturbation function that invokes chaotic undulations to the energy landscape. As long as the perturbation is small compared to E_{Hop}, we can make sure the problem specification is unchanged, and at the same time providing a means to avoid the trappings of local minima.

3.6 Concluding remarks

There are many existing approaches to incorporating chaos into Hopfield networks. While some studies have compared the different approaches as seen in Chapter 2, a true comparison is difficult when the underlying chaotic mechanism is different. In this chapter we have proposed a unified framework in which these chaotic models can be placed and compared. An energy modifier is added to the original Hopfield energy to form a new energy landscape that admits various chaotic dynamics. The significance of this is the provision of a means to classify existing HCNN models and study the network dynamics in a new perspective. Two models with internal approaches to chaos are used as examples. One employs self-feedback connections to facilitate CSA, with a hysteretic annealing approach to the energy modifier. The other one uses a variable time-step for CSA, with an energy modifier that performs a time dependent scaling to the Hopfield energy landscape. Both energy modifiers are found to be quadratic, which can be a common feature for all internal approaches to chaos. The CSA process is described as a temporary deformation to the energy landscape but asymptotically preserving its shape. A model with external chaotic noise is also studied and its energy modifier is shown to cause chaotic energy perturbations that allow the network dynamics to avoid local minima. The energy modifier for this noise approach is linear in form, which we suggest may be the reason for its lesser optimization ability when compared to models with internal approaches. Using the proposed framework, different HCNN models are all studied with a global perspective with the introduction of the energy modifier to the original Hopfield energy. Already this framework has allowed observations that the Chen-Aihara method is equivalent to adding logistic maps to the Hopfield energy function, and operates like hysteretic annealing. It is anticipated that this framework approach can serve as a potential starting point for identifying new models of HCNN, as well as helping to provide additional insights into the existing models.

Chapter 4

Computational Performance of Hopfield-type Chaotic Neural Networks

4.1 Introduction

In this chapter, we present the computational results of applying Hopfield-type chaotic neural networks (HCNN's) to solve a COP. The three main types of HCNN models as described by the unified framework in the last chapter are included in this investigation: the Chen-Aihara model of chaotic simulated annealing (CSA) with decaying self-coupling, the Wang-Smith model of CSA with decaying time-step, and the model with chaotic noise. These models are implemented to solve the *N*-queen problem via computer simulation, and their optimization performance is then measured and compared.

The N-queen problem is chosen to be the test problem for our computational investigations in this thesis. It belongs to the class of NP-hard constraint satisfaction problems and captures many characteristics of other similar problems found in business and engineering applications. Its simplicity in mathematical representation makes theoretical and computational investigations more effective in our study of HCNN's. The N-queen problem is introduced in Section 4.2.

In order to study computationally the optimization performance and dynamics of the three HCNN models discussed previously, they are implemented to solve the N-queen problem by using the Hopfield-Tank approach [54]. An energy function is constructed such that it acts as a penalty function of the COP. Since the three chaotic neural network models are Hopfield based and share the same basic Hopfield energy function (E_{Hop}) discussed in the last chapter, they have the same N-queen implementation, but different dynamics. Details of the N-queen implementation are given in Section 4.3.

In our computational investigation, we focus on theoretical aspects of the HCNN models as to how their dynamics lead to improved optimization performance. For each model, the N-queen problem of various sizes are solved computationally, and their optimization performance measured on respective parameter spaces. The chaotic dynamics of the models are also investigated by examining the time evolution of their neuron states and energy functions. All these analyses allow us to discover and discuss the relationships between chaotic dynamics and optimization performance of HCNN's. The computer simulations and results are presented in Section 4.4, with specific results for the Chen-Aihara model, the Wang-Smith model, and the chaotic noise model in Section 4.4.1, 4.4.2 and 4.4.3 respectively. The discussions of the results are concluded in Section 4.5.

4.2 The *N*-queen problem

The *N*-queen problem is concerned with placing *N* queens on an $N \times N$ chessboard in such a way that no queen is under attack. It belongs to the class of NP-hard constraint satisfaction problems, and the constraints are:

- only one queen in each row;
- only one queen in each column;
- only one queen in each diagonal (there are more than two diagonals); and
- exactly *N* queens on the chessboard.

To solve this COP, we formulate it as a 0-1 programming problem. A neuron output matrix **X** of dimension $N \times N$ is used to represent the chessboard, where $X_{ij} = 1$ corresponds to a queen placed in row *i* and column *j*, and $X_{ij} = 0$ means no queen is placed. Figure 4.1 shows a feasible 5-queen solution using both the chessboard and matrix representations. Mathematically, the *N*-queen constraints are expressed as follows:

Integrality: $X_{ij} \in \{0,1\}$

Row constraint: $\sum_{j=1}^{N} X_{ij} = 1$ for $i = 1, 2, \ldots, N$

$$\text{Column constraint:} \qquad \sum_{i=1}^{N} X_{ij} = 1 \qquad \text{for } j = 1, 2, \ldots, N$$

$$\text{Diagonal constraints:} \qquad \left.\begin{array}{l} \displaystyle\sum_{\substack{1\le i-p\le N\\ 1\le j-p\le N}} X_{i-p,j-p} = 1 \\[2ex] \displaystyle\sum_{\substack{1\le i-p\le N\\ 1\le j+p\le N}} X_{i-p,j+p} = 1 \end{array}\right. \qquad \text{for } i, j = 1, 2, \ldots, N$$

$$\text{Totality constraint:} \qquad \sum_{i,j=1}^{N} X_{ij} = N \qquad \text{for } i, j = 1, 2, \ldots, N \tag{4.1}$$

The 8-queen problem was first proposed in 1848, and has been traditionally investigated by mathematicians, including C. F. Gauss in 1850 [112] and more recently by Yaglom and Yaglom [113]. The *N*-queen problem has become a research subject for various fields like algorithmic design [77, 56], artificial intelligence [46, 92], and parallel/distributed computing [3, 55]. It has also been examined under the phase transition theory in search problems [86]. The *N*-queen problem has been solved with a variety of artificial neural networks previously, e.g. Hopfield network [95], Cauchy machines [96], Gaussian machines [8] and the hysteretic Hopfield network [18].

Apart from being a challenging problem in its own right, *N*-queen has implications for many practical applications in business and industry, such as air traffic control, data routing and load balancing in computer clusters, scheduling, etc. All these problems share the common goal of satisfying a set of *global constraints.* Feasible solutions to these problems mean satisfying all the specified constraints, and this makes the solution process a difficult one. This is best illustrated by the *N*-queen example, where the placing of any queen affects the "following moves" and the final solution. In neural network implementations of solving the N-queen problem, this interdependence of queen positions is translated into the interaction among neurons and their associated attractors in the state space. Thus in order to study the optimization performance of HCNN models and their chaotic dynamics, we choose N-queen as our test problem for its simplicity and representative characteristics. One of the reasons for choosing the N-queen problem over the TSP as our test problem is that almost all existing studies of HCNN models in the literature use the TSP [114, 76, 99, 24, 48, 15, 47, 97], therefore it would be a contribution to the field by solving a different, but equally hard COP such as the N-queen. One may argue that constraint satisfaction problems are easier to solve due to the lack of cost terms in the penalty function, and the difficult trade-off issue between solution quality and constraint satisfaction is non-existent. But in fact this does not make the N-queen problem any easier to solve, as the trade-off issue still exists among the various terms in the penalty function, only that they all represent constraints.

4.3 Formulation

In this section, we implement the HCNN models by using the Hopfield-Tank method of constructing a cost from the COP, which is then mapped to the Hopfield energy [54]. Since all three HCNN models examined in previous chapters share the same N-queen implementation, they have the same weight matrix $\mathbf{W}$ and external input $\mathbf{I}$. In order to satisfy the constraints in (4.1), a cost function (or penalty function) is constructed such that its value increases if a constraint is violated:

$$f = \frac{A}{2}\sum_{i,j=1}^{N}\sum_{l=1,l\neq j}^{N} X_{ij}X_{il} + \frac{B}{2}\sum_{i,j=1}^{N}\sum_{k=1,k\neq i}^{N} X_{ij}X_{kj} + \frac{C}{2}\left(\sum_{i,j=1}^{N} X_{ij} - N\right)^{2}$$
$$+\frac{D}{2}\left(\sum_{i,j=1}^{N}\sum_{\substack{p\neq 0\\ 1\leq i-p\leq N\\ 1\leq j-p\leq N}} X_{ij}X_{i-p,j-p} + \sum_{i,j=1}^{N}\sum_{\substack{p\neq 0\\ 1\leq i-p\leq N\\ 1\leq j+p\leq N}} X_{ij}X_{i-p,j+p}\right) \tag{4.2}$$

where A, B, C and D are positive parameters. Each term in (4.2) is associated with a constraint described above: the first and second terms corresponds to the constraints of having only one queen in each row and each column respectively; the third term corresponds to having exactly N queens; and the last term corresponds to the diagonal constraint. Expanding (4.2), neglecting the constant term $CN^2/2$ and comparing (4.2) to the energy function of Hopfield's discrete model [54]:

$$E_d = -\frac{1}{2}\sum_{i,j,k,l=1}^{N} W_{ijkl}X_{ij}X_{kl} - \sum_{i,j=1}^{N} I_{ij}X_{ij} \tag{4.3}$$

gives

$$f = \frac{A}{2}\sum_{i,j,k,l=1}^{N}\delta_{ik}(1-\delta_{jl})X_{ij}X_{kl} + \frac{B}{2}\sum_{i,j,k,l=1}^{N}\delta_{jl}(1-\delta_{ik})X_{ij}X_{kl}$$
$$+\frac{C}{2}\left(\sum_{i,j,k,l=1}^{N} X_{ij}X_{kl} - 2N\sum_{i,j=1}^{N} X_{ij}\right)$$
$$+\frac{D}{2}\left(\sum_{i,j,k,l=1}^{N} X_{ij}X_{kl}\left(\delta_{i+j,k+l} + \delta_{i-j,k-l}\right)\right)(1-\delta_{ik}) \tag{4.4}$$

where $\delta_{ik} = 1$ (0) if $i = k$ ($i \neq k$). Moreover

$$\begin{aligned} W_{ijkl} &= -A\delta_{ik}(1-\delta_{jl}) - B\delta_{jl}(1-\delta_{ik}) - C \\ &\quad - D(1-\delta_{ik})(\delta_{i+j,k+l} + \delta_{i-j,k-l}) \end{aligned} \tag{4.5}$$

which are the coefficients of the quadratic terms in (4.3). For the remaining linear term,

$$I_{ij} = CN\,. \tag{4.6}$$

Through (4.5) and (4.6), the N-queen problem is fully specified for our HCNN models, and the difference equations (3.9), (3.17) and (3.23) of the three corresponding models all carry the same values of W_{ijkl} and I_{ij} given here.

4.4 Computer simulations and results

In this section, computational experiments for investigating the dynamics and properties of the HCNN models are described. The N-queen problem is solved by computer simulations of the three HCNN models described in this paper, focusing on the aspects of optimization dynamics, performance, and comparisons among models. In contrast to the original Hopfield network with steepest descent dynamics only, HCNN's can give high quality solution without any adjustment of the penalty parameters A, B, C, D in (4.5) and (4.6). Hence, $A = B = C = D = 1$ is applied for all experiments. In these experiments, only asynchronous updating of neuron states is considered.

4.4.1 CSA with decaying self-coupling

In this section, the Chen-Aihara CSA model is used to solve the N-queen problem via computer simulation, with $N = 5$, 10 and 20. The updating equation (3.9) is used with the sigmoidal activation function (2.8) and the annealing schedule (3.10). A round consists of 100 runs of the algorithm, each with a different set of initial states with X_{ij} values randomly centered around 0.5. The system is said to be convergent and hence terminated if the criterion $\max\{|X_{ij}(t+1)-X_{ij}(t)|; i,j=1,2,\dots,N\}<5\times10^{-5}$ is met. Note that this is a strict implementation of the theoretical convergence criterion, and in general a more efficient alternative for heuristicaliy checking the convergence is to inspect the discrete neuron states [75, 24]. First, we investigate the optimization performance of the model in terms of feasibility, efficiency, robustness and scalability in a two-parameter space. Feasibility is measured by the proportion of optimal solutions obtained in a round; efficiency is the average number of iterations for convergence; robustness is concerned with the distribution of high feasibility regions; and scalability is about the dependency of these measures on problem size.

The parameters used are: ε = 0.004, β = 0.001 and I_0 = 0.65, as used by Chen and Aihara [24]. $k = 1 - \alpha$ is used according to (3.8). The parameter space is defined by α and $z(0)$, which are the main parameters concerned with the balance of steepest descent and nonlinear feedback terms. Figure 4.2(a)-(c) shows the feasibility obtained for N = 5, 10 and 20 respectively. A grey-scale is used where a darker region means a higher feasibility. It can be observed that a high feasibility of over 90% is obtained for the best cases, and the high feasibility region spans a large portion of the parameter space, showing its good robustness in terms of choosing parameters giving high quality solutions. Moreover, the high feasibility region does not shrink significantly when the problem size increases, thus suggesting good scalability. Figure 4.3(a)-(c) shows the average number of iterations for convergence, with the grey-scale measuring efficiency. Lighter shade means fewer iterations to reach convergence, or higher efficiency. By inspecting both Figure 4.2 and 4.3, we can observe that regions with higher feasibility generally correspond to fewer iterations for convergence, which is generally not true for other optimization algorithms e.g. simulated annealing.

This relationship is particularly observable as the problem size increases. Another feature found in the feasibility diagrams is the gradual transition in feasibility from the dark region to the light region, which suggests a gradual change of optimization dynamics.

In order to look more closely into the neurodynamics corresponding to different feasibility regions, the time evolution of neuron states and the energy E_d of a typical run with $N = 10$, $\alpha = 0.4$ and $z(0) = 1.0$ is shown in Figure 4.4. Figure 4.4(a) shows the instability of a typical internal state with the chosen parameters. Before $t = 550$, its value is mostly negative, but afterwards it becomes mostly positive and also less scattered. This change of dynamics can be more easily observed in the neuron output state shown in Figure 4.4(b), where the output value of 1 is much more frequently attained after $t = 550$. We can have a glimpse of the overall neurodynamics by looking at Figure 4.4(c), which shows the time evolution of the computational energy E_d. The instability of the overall system is illustrated by the chaotic motion in E_d. In spite of the instability, there is a gradual lowering of the value of E_d as time increases, and it appears that a two-cycle is reached when $t > 600$. A global minimum of $E_d = -50$ is achieved in this run and due to our convergence criterion, the system is halted before it can settle to a stable fixed point. Strictly speaking, the whole process of CSA is not completed in this run, but the main purpose of optimization is fulfilled. Since the choice of $\alpha = 0.4$ and $z(0) = 1.0$ in this particular experiment corresponds to a point on the right side of the two-parameter space in Figure 4.2(b) and 4.3(b), the observed instabilities are caused by the strong initial self-coupling. Experiments show that similar dynamics as shown in Figure 4.4(c) are obtained for many initial conditions

X(0). In other words, the optimization neurodynamics in this parameter regime is insensitive to initial conditions, and this property is a major contribution to the high feasibility shown in Figure 4.2(b).

One feature of nonlinear systems is their diverse structural dynamics obtainable by changes in parameter values. In Figure 4.5, we show how the structure of the neurodynamics varies when we move across the two-parameter space in Figure 4.2(b) or 4.3(b), which enables us to extract nonlinear features that are relevant to improved optimization performance. Figure 4.5 shows various time evolutions of E_d with $\alpha = 0.4$ and decreasing values of $z(0)$. The same set of initial neuron states is used for all cases unless otherwise stated. Figure 4.5(a) shows similar unstable neurodynamics as in Figure 4.4(c) but the number of iterations to obtain the optimal solution is halved. According to Figure 4.2(b) and 4.3(b), this parameter set offers both high feasibility and high efficiency. Because of the annealing schedule used, this high efficiency is a result of the choice of $z(0)$ just large enough to sustain an effective but brief period of instability before convergence. The dynamics here are also insensitive to initial neuron states, thus contributing to the high feasibility observed.

If a smaller value of $z(0) = 0.4$ is used as shown in Figure 4.5(b) and (c), the network dynamics becomes less chaotic but more sensitive to initial conditions. In Figure 4.5(b), the systems start with a two-cycle, which appears to bifurcate and becomes chaotic around $t = 500$, and then back to a two-cycle. The optimal solution is not reached in this case, but it shows that transient chaos due to bifurcation exists even when $z(t)$ is monotonically decreasing, where only *reverse* bifurcation is expected. An entirely different behaviour is observed if another set of initial neuron states is used as shown in Figure 4.5(c). The system starts with a simple descent and then locks into a two-cycle, which becomes stable at a global minimum (fixed point) around $t = 60$. These two kinds of dynamics are typical for this parameter regime when other initial states are used, with the optimal solution mostly obtained from the dynamics as in Figure 4.5(c). Note that the window of chaotic behaviour in Figure 4.5(b) does not in general improve optimization. The lack of instability in the early stage of neurodynamics and its dependency on initial conditions accounts for the low feasibility and low efficiency for this parameter regime.

Figure 4.5(d) corresponds to the case with $z(0) = 0.05$, which means the initial self-coupling is very weak. The dynamics observed are far from unstable, but a small uphill movement is still present around $t = 8$. However, the system is trapped at a local minimum, which is typical for almost all initial conditions. Chaos is nonexistent in this parameter regime, and as shown in Figure 4.2(b) and 4.3(b), low feasibility is coupled with fast convergence. In other words, this parameter regime represents fast convergence to a poor solution.

By relating optimization performance in the two-parameter space and the corresponding neurodynamics given above, we have identified adequate instability to be the functional feature in the decaying self-coupling CSA scheme, which serves as a computational element to the HCNN in solving the N-queen problems with both high feasibility and efficiency.

4.4.2 CSA with decaying time-step

In this section, the Wang-Smith CSA model is implemented as a computer simulation to solve the N-queen problem with $N = 5$, 10 and 20. The difference equation (3.17) is used with the same sigmoidal activation function (2.8) as in the last section, together with the annealing schedule (3.16). The same 100 sets of initial neuron states and convergence tolerance are used, as for the previous experiments with the decaying self-coupling model. The four measures of optimization performance, namely feasibility, efficiency, robustness and scalability are likewise investigated in a two-parameter space.

The constant parameters used are: $\varepsilon = 0.01$ and $\beta = 0.001$. The parameter space is defined by τ and $\Delta t(0)$, which are the main parameters controlling the stability of the neurodynamics. Figure 4.6(a)-(c) shows the feasibility obtained for $N = 5$, 10 and 20 respectively, and Figure 4.7(a)-(c) shows the corresponding efficiency plots.

From Figure 4.6(a)-(c), there exists a wedge-shaped dark region where high feasibility can be obtained, with the best cases ranging from 80 ~ 90% depending on N. The scalability of this model is marginally worse than the previous CSA model because of this slight deterioration in feasibility as problem size increases. However, the drop in feasibility inside the dark region is less than the drop in the light region on the top-left half of the parameter space, suggesting a difference in scalability for the neurodynamics in the two regions. Robustness is good for this model in the sense that the high feasibility region is distinctly well defined in the parameter space, in contrast to the diffuse boundary between high and low feasibility regions found in Figure 4.2(a)-(c). For optimization efficiency, Figure 4.7(a)-(c) shows the average number of iterations for convergence for the corresponding problem sizes. High efficiency is obtained within around 300 iterations to achieve high feasibility inside the wedge-shaped region. This represents a region ideal for optimization where both high feasibility and high efficiency co-exist, which is also found in the previous CSA model. As observed in Figure 4.6, there are in general three regions where feasibility differs. The lower right half of the parameter space corresponds to divergent network dynamics which are irrelevant for optimization. The dark wedge-shaped region is on the right half of the parameter space, which corresponds to network instability due to the violation of the first stability condition in (2.17).

We again look more closely into the neurodynamics for the other parts of the parameter space by observing the time evolution of neuron states and the energy E_d. Figure 4.8 shows the network dynamics with parameters $\tau = 0.5$ and $\Delta t(0) = 1.0$, which corresponds to a point on the right edge of the high feasibility region. Figure 4.8(a) shows the time evolution of the internal state of a typical neuron in the chosen parameter regime. Instability dominates the early part of the dynamics, with mostly negative values. After around $t = 600$, a two-cycle appears and later converges to a stable fixed point. This reverse bifurcation process appears as a switching scenario as shown in Figure 4.8(b), which shows the neuron output switching intermittently between 0 and 1 before $t = 600$, and then converges to the final output of 1. Figure 4.8(c) shows the time evolution of E_d, which summarizes the overall dynamics. The instability in the neuron states causes the instability in E_d, but a trend of gradually decreasing energy can be observed before the network stabilizes at the global minimum. This phenomenon of optimization by reverse bifurcation has been observed in Figure 4.4 for the decaying self-coupling model, and is a characteristic feature for effective optimization for both models. Also, the neurodynamics shown here (Figure 4.8) are insensitive to initial conditions, as in the previous model. To move across the two-parameter space as shown in Figure 4.6(b) or 4.7(b), we fix the value of $\tau = 0.5$ and use successively smaller values of $\Delta t(0)$ to investigate different neurodynamics in various parameter regimes. Figure 4.9 shows various time evolutions of E_d with different initial time-step $\Delta t(0)$. Exactly the same set of initial neuron states as used previously for the decaying self-coupling model is used here, unless otherwise stated. In Figure 4.9(a), the use of a smaller $\Delta t(0) = 0.7$ than in Figure 4.8(c) means the period for unstable dynamics and the resultant CSA process is shortened. However, the initial instability is still strong enough for solution searching and consequently arriving at a global minimum, with much fewer iterations than in Figure 4.8(c). The characteristic neurodynamics and fast convergence to the global minimum shown here are also insensitive to initial states of the network. With high feasibility and efficiency, together with low dependency on initial network states, the model is at its optimal configuration. To look at the neurodynamics of the model with parameters lying on the boundary of low and high feasibility regions of Figure 4.6(b), a smaller initial time-step of $\Delta t(0) = 0.5$ is used. Figure 4.9(b) shows the dynamics of such a case. Weak instability is observed at the beginning, which soon becomes a two-cycle that leads to a fixed point. This is an example of inadequate chaotic search and hence poor optimization dynamics that leads to local minima. If an even smaller value of $\Delta t(0) = 0.35$ is used, as shown in Figure 4.9(c), CSA is nonexistent and only weak uphill movement is possible. Optimization performance in this parameter regime is very low (see Figure 4.6(b) and 4.7(b)), and the network is often trapped in local minima.

By associating regions of various optimization performance in the two-parameter space with the spectrum of neurodynamics given above, the decaying time-step CSA model can be seen to share the same characteristics with the decaying self-coupling model of having network instability as a functional element for effective optimization. Parameter regimes that give rise to adequate instability, thus high optimization performance, are also found for this model. Although the two CSA models generate internal chaos in different ways, the neurodynamics as perceived by the evolution of the computational energy (c.f. Figure 4.4 and 4.8) show that they behave similarly and hence have comparable optimization performance.

4.4.3 HCNN with chaotic noise

In this section, chaotic time series, or chaotic noise, is fed into the Hopfield network. Since the chaotic feature is not generated from within the neural network, the resulting neurodynamics are significantly different from the two HCNN models previously described. To illustrate the optimization ability of this model, the *N*-queen problem with N = 5, 10 and 20 is solved computationally using the difference equation (3.23), the sigmoidal activation function (2.8), and the logistic map time series given by (3.21) and (3.22).

Because of the added noise, the network does not converge to a stable fixed point, thus we set $t = 2000$ as the maximum number of iterations allov ed. For each network configuration, 100 sets of initial neuron states are used for 100 runs, and the feasibility measure is now the proportion of runs which encounter the optimal solution at least once. For example, say in a particular run the optimal solution is attained once or more times at any $t \leq 2000$, we count this run as successful in producing an optimal solution, no matter how many times the optimal solution is encountered in the non-convergent process (c.f. Section 4.4.1 and 4.4.2 in which the models are convergent). The parameters used are: $\varepsilon = 0.1$, $\Delta t = 0.1$, $\tau = 1$ and $a = 4$. Figure 4.10 shows a typical logistic map time series with $a = 4$ and $A_\eta = 1$. Each neuron has its unique logistic map time series generated with a different initial condition. To investigate the effect of noise amplitude on optimization performance and neurodynamics, the noise amplitude factor A_η is varied from 0.0 to 0.68, which corresponds to the actual noise amplitude of 0.0 to 1.02. The feasibility obtained for each case is shown in Figure 4.11. For the case of $N = 5$ (Figure 4.11(a)), feasibility increases with chaotic noise amplitude for small amplitudes. 100% feasibility is obtained as the noise amplitude reaches around 0.1, which is maintained until the amplitude becomes too large (~1.5) to be beneficial. Therefore, chaotic noise of medium amplitude yields ideal feasibility for this problem size. Figure 4.11(b) shows the case of $N = 10$. It can also be observed that medium noise amplitude gives high feasibility, although the range of appropriate noise amplitude is narrower. The best performance is achieved with a noise amplitude of slightly less than 0.4, giving a feasibility of over 90%, which is comparable to the two CSA models discussed. Note that when the noise amplitude is 0, i.e. no chaotic noise added, feasibility is only about 10%. For an even larger problem size of $N = 20$ (Figure 4.11(c)), an increase in feasibility due to chaotic noise is also observed for small amplitudes of around 0.15, which peaks at around 40%, but deteriorates as the amplitude further increases, i.e. an even narrower band of usable noise amplitudes. This lesser performance is most likely due to the dramatic increase of local minima as the problem size increases, and the clear inability of external noise to assist the dynamics to escape local minima. In general, high feasibility which peaks at some amplitude values can be achieved with this HCNN, with good robustness for small problem sizes. Scalability is poor for its rapid shrinkage of desirable range of noise amplitudes as problem size increases.

To look more closely into the effect of chaotic noise amplitude on the neurodynamics, we show in Figure 4.12 the time evolution of E_d and a typical internal state $Y_{3,3}$ for noise amplitudes of 0.36 and 0.15 (i.e. $A_\eta = 0.24$ and 0.1 respectively), with $N = 10$. Figure 4.12(a) and (b) correspond to the energy E_d and internal state $Y_{3,3}$ respectively when the optimal noise amplitude is used (see Figure 4.11(b)). Figure 4.12(a) shows a noisy neurodynamics with both downhill and uphill trends, with global minima obtained at around $t = 400$ and 1950. This noisy behaviour is caused by the chaotic noise injected into the internal states. Figure 4.12(b) shows the internal state dynamics of a typical neuron. If a smaller noise amplitude is used, we obtain a different picture of the neurodynamics as shown in Figure 4.12(c) and (d). From Figure 4.12(c), only a downhill trend can be observed, and the noise reflected in E_d is much smaller compared to Figure 4.12(a). This can be explained by the inability of small perturbations in **Y** to "kick" the network state to another local minima, as shown in Figure 4.12(d) where a less noisy dynamics in $Y_{3,3}$ is observed when compared to Figure 4.12(b). Note that a change in noise amplitude can change not merely the noisiness in the internal state dynamics, but the overall trend as well (compare Figure 4.12(b) and (d)). Hence, the existence of an optimal noise amplitude for effective optimization maybe a consequence of the specific chaotic perturbations of the internal state resonating with the specific energy landscape of the problem. As such, the detailed interaction between the perturbations and the energy landscape should be further investigated as a stochastic resonance phenomenon [110].

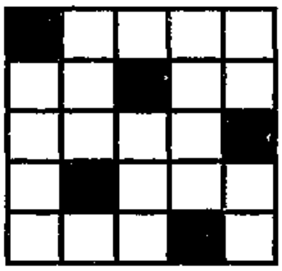
4.5 Concluding remarks

In this chapter, we have investigated the dynamical behaviors and optimization performance of the three HCNN models described by the unified framework in Chapter 3. The Chen-Aihara and Wang-Smith CSA models, together with the chaotic noise model, are included. Evidence of improved optimization as a result of chaos is obtained by solving the *N*-queen problem as an example. By means of computer simulation, the two CSA models, which belong to the same class of having internal chaos, are found to be capable of strong optimization performance by having high feasibility, efficiency, robustness and scalability. The spectrum of neurodynamics obtained from computations illustrates the flexibility of HCNN models to behave as different attractors in the dynamical sense, where the steepest descent dynamics of the Hopfield network and the chaotic dynamics of the HCNN occupy opposite ends of the spectrum. By measuring the optimization performance in a two-parameter space, we have experimentally illustrated the link between this spectrum of attractors and optimization performance: chaotic neurodynamics are beneficial to optimization. Our results show that an adequately strong chaotic dynamics at the early stage is crucial for effective optimization, which also serves as a guide to choosing model parameters.

On the other hand, the chaotic noise model belongs to the class with external chaotic dynamics, and the logistic map time series is used as chaotic noise in our experiments. The essential difference between external and internal chaotic sources in HCNN models can be highlighted by contrasting the ways in which logistic maps are involved in the energy modifiers of the Hopfield network with chaotic noise model and the decaying self-coupling model. For the former, the logistic map time series is independent of neuron states; but for the latter, each "iteration" of the logistic map involves the collective response of all neurons. Although the chaotic noise structure is entirely independent of the network states in the chaotic noise model, its ability to improve feasibilities comparable to CSA models is demonstrated via computer simulations. It is found that the feasibility achieved by this model is related to the amplitude of the chaotic noise used, with peak values obtainable for certain amplitudes depending on problem size. However, robustness deteriorates rapidly with problem size, and results in poor scalability.

Since both the form of the energy modifier and the neurodynamics obtained for the chaotic noise model are different to the two CSA models, it is likely that there is a different mechanism to explain the enhanced optimization performance due to chaotic noise. From the energy framework perspective, one possible candidate is the phenomenon of stochastic resonance [110] commonly found in nonlinear systems with multi-stable states, where added noise is beneficial in certain aspects.

Possible future research should involve computational investigations of new HCNN models suggested by the unified energy framework, and the application of HCNN's to other COP's. These would further our theoretical understanding of HCNN's as well as their implementation as an effective optimization tool.



$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 4.1: Chessboard and matrix representations of a 5-queen feasible solution

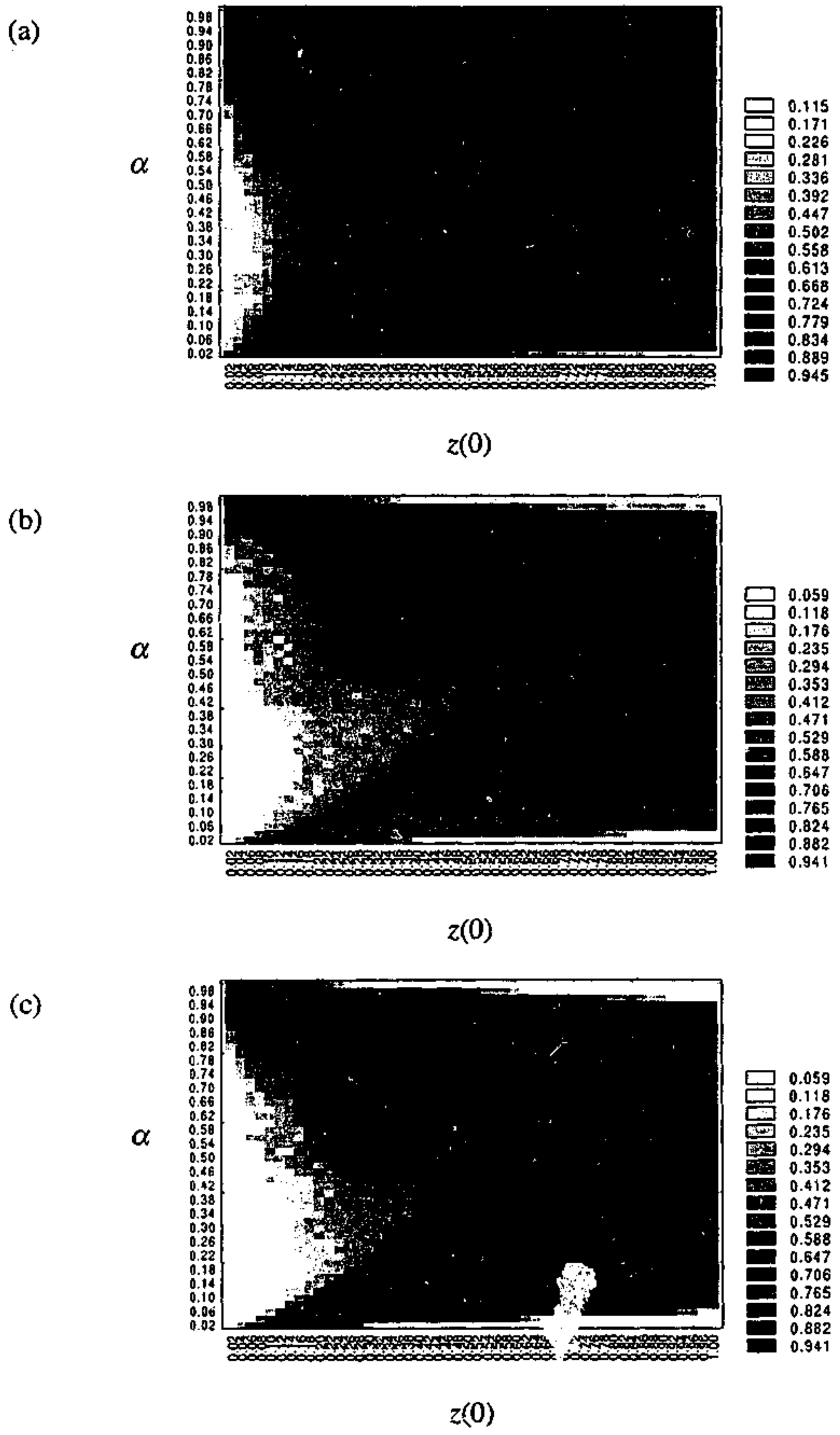


Figure 4.2: Feasibility graphs of CSA with decaying self-coupling solving the N-queen problem with (a) $N = 5$; (b) $N = 10$; and (c) $N = 20$.

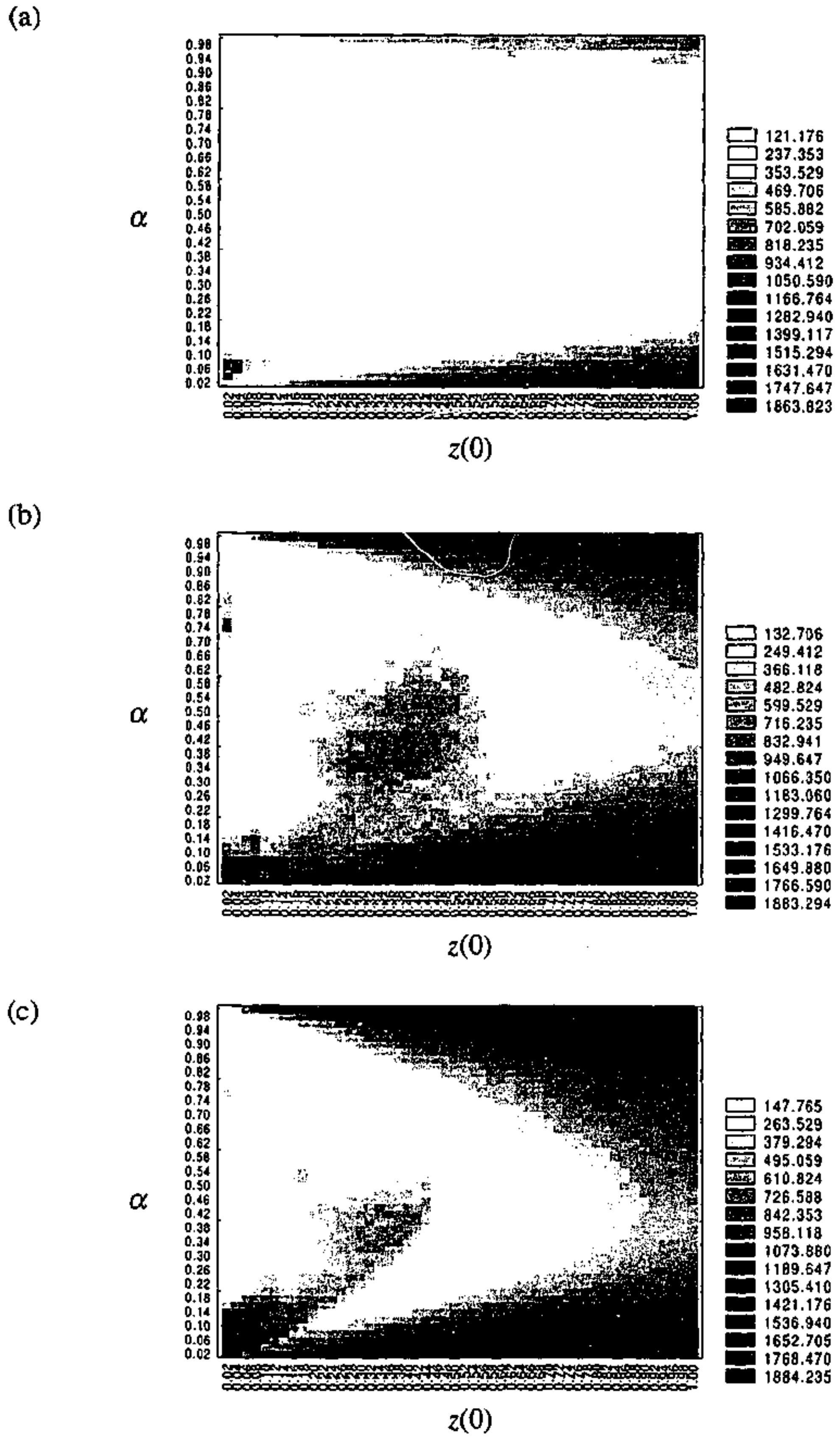


Figure 4.3: Efficiency graphs (average number of iterations) of CSA with decaying self-coupling solving the N-queen problem with (a) $N = 5$; (b) $N = 10$; and (c) $N = 20$.

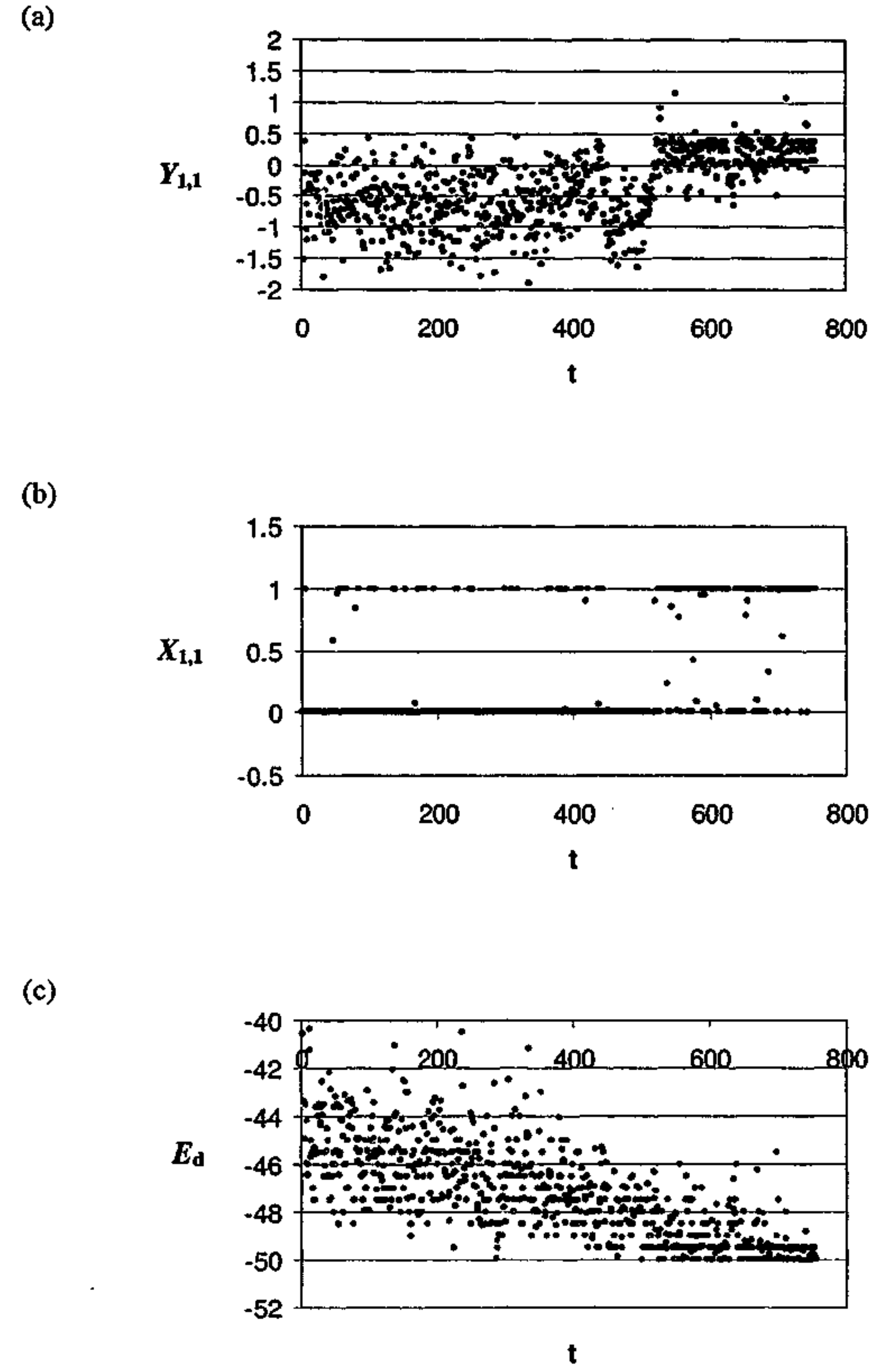


Figure 4.4: Time evolution of the neurodynamics of CSA with decaying self-coupling solving the 10-queen problem using $\alpha = 0.4$ and $z(0) = 1.0$. (a) The internal state $Y_{1,1}$. (b) The corresponding neuron output $X_{1,1}$. (c) The computational energy E_d.

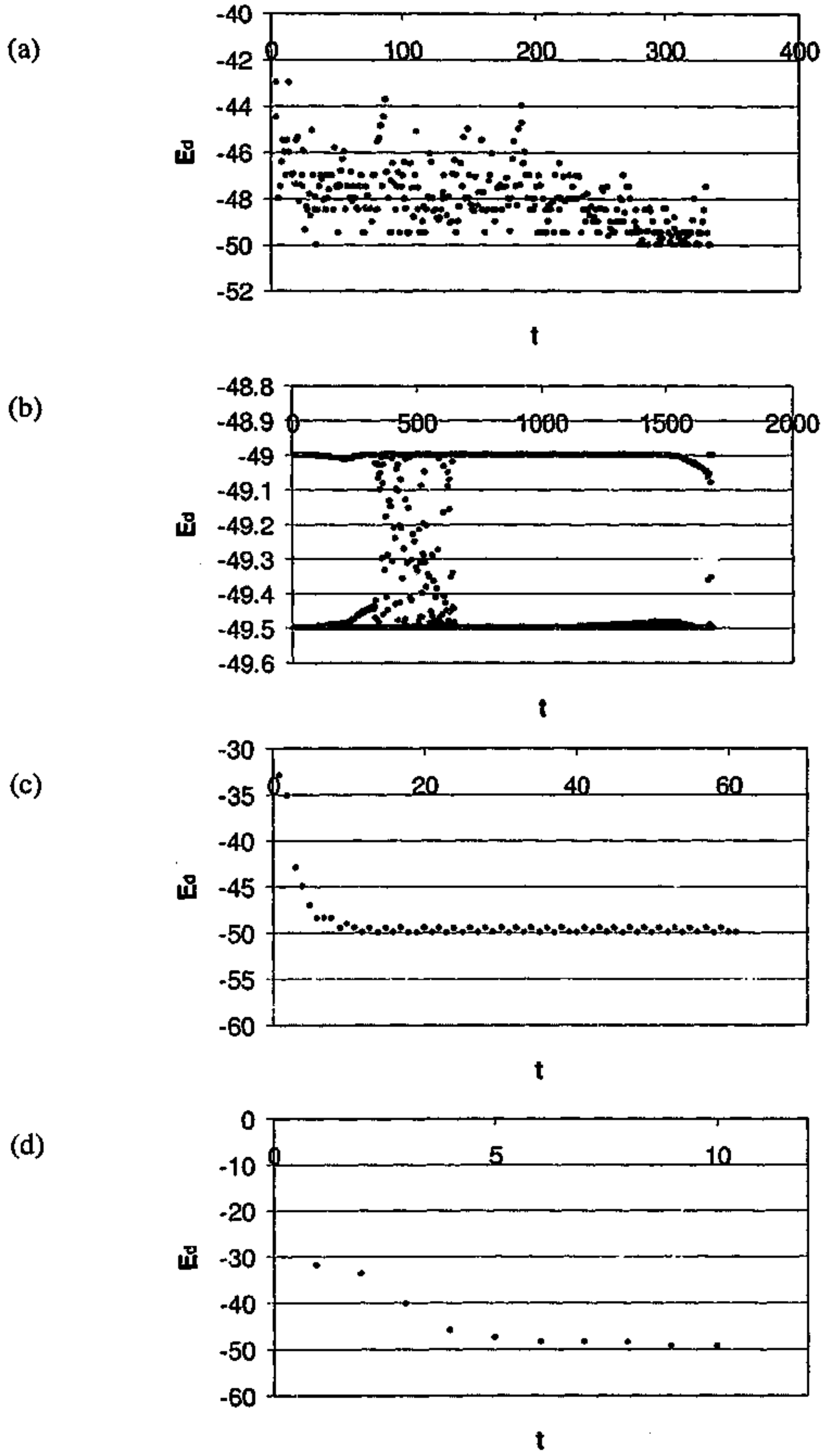


Figure 4.5: Time evolution of E_d with α = 0.4 and different values of $z(0)$. (a) z(0) = 0.7. (b) z(0) = 0.4. (c) z(0) = 0.4 with another **X**(0). (d) z(0) = 0.05.

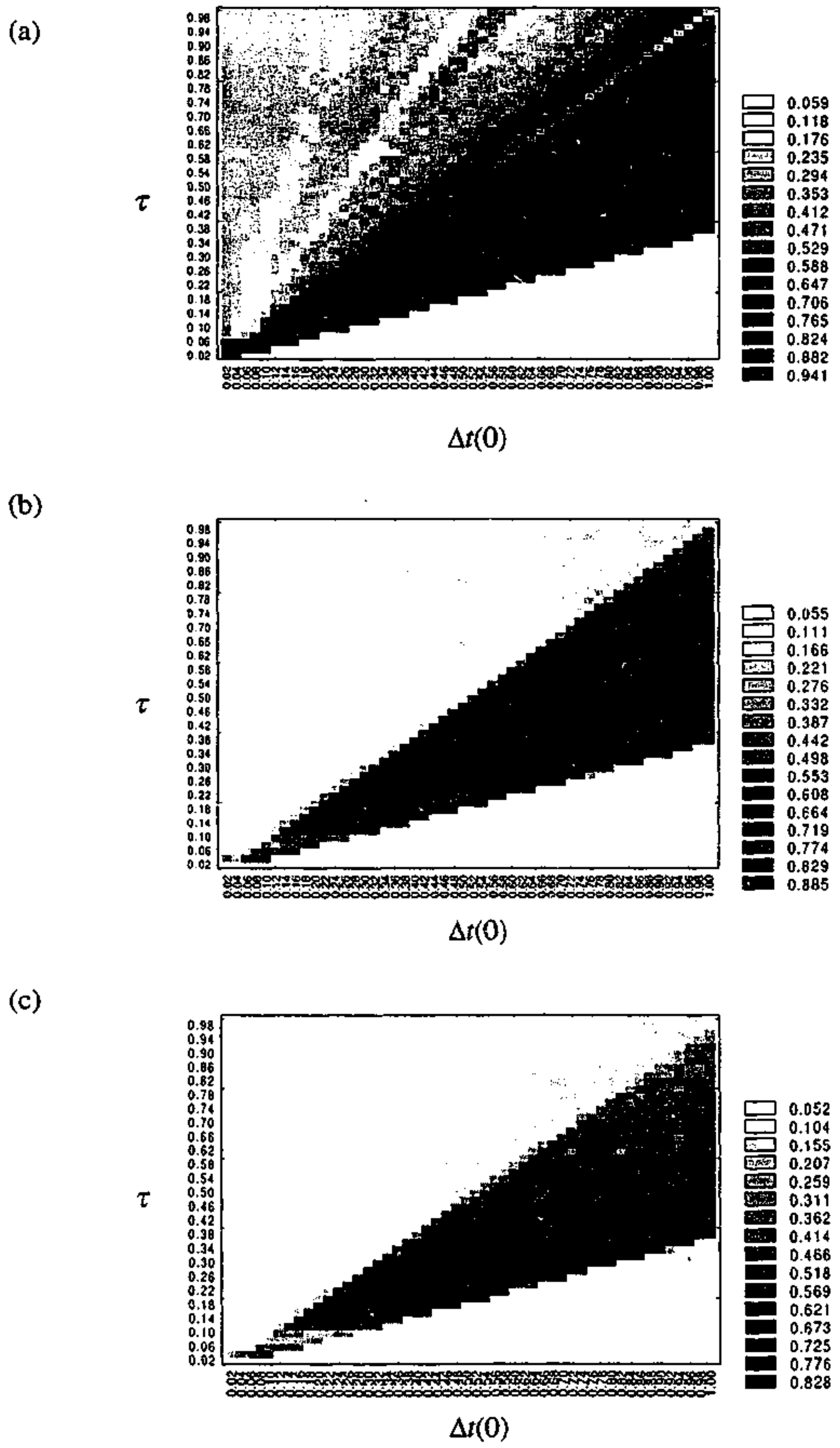


Figure 4.6: Feasibility graphs of CSA with decaying time-step solving the N-queen problem with (a) $N = 5$; (b) $N = 10$; and (c) $N = 20$.

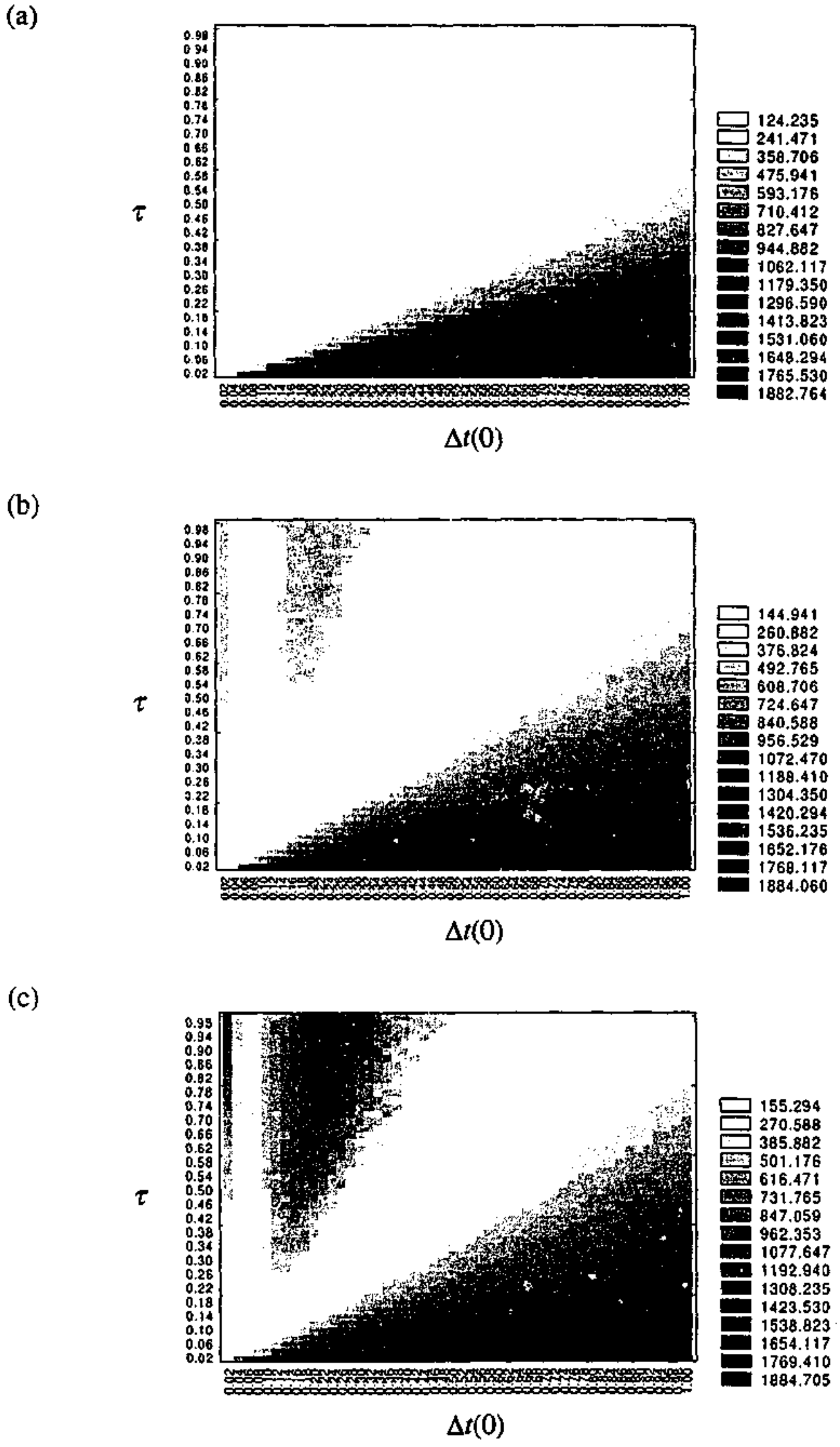


Figure 4.7: Efficiency graphs (average number of iterations) of CSA with decaying time-step solving the N-queen problem with (a) $N = 5$; (b) $N = 10$; and (c) $N = 20$.

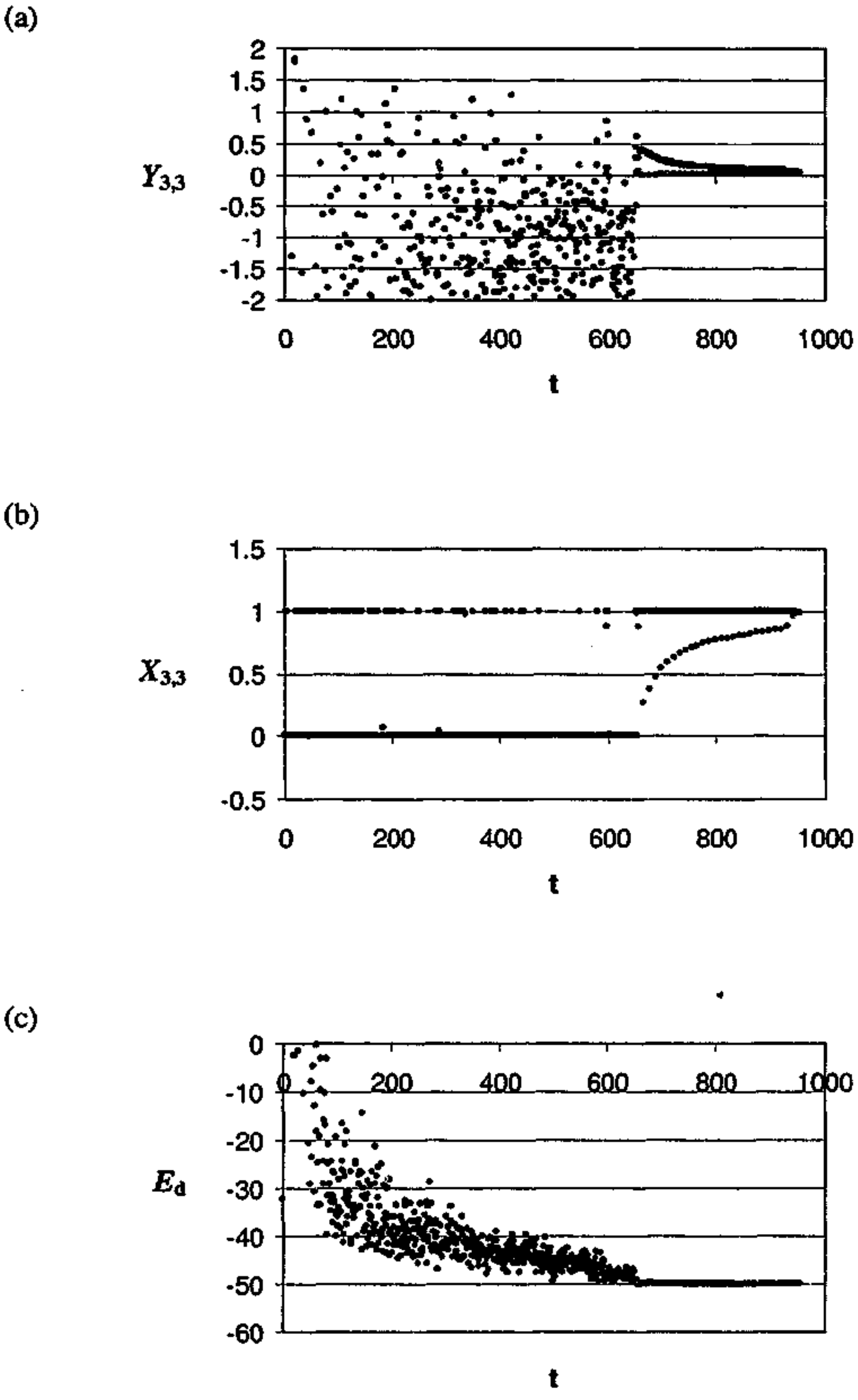


Figure 4.8: Time evolution of the neurodynamics of CSA with decaying time-step solving the 10-queen problem using $\tau = 0.5$ and $\Delta t(0) = 1.0$. (a) The internal state $Y_{3,3}$. (b) The corresponding neuron output $X_{3,3}$. (c) The computational energy E_d.

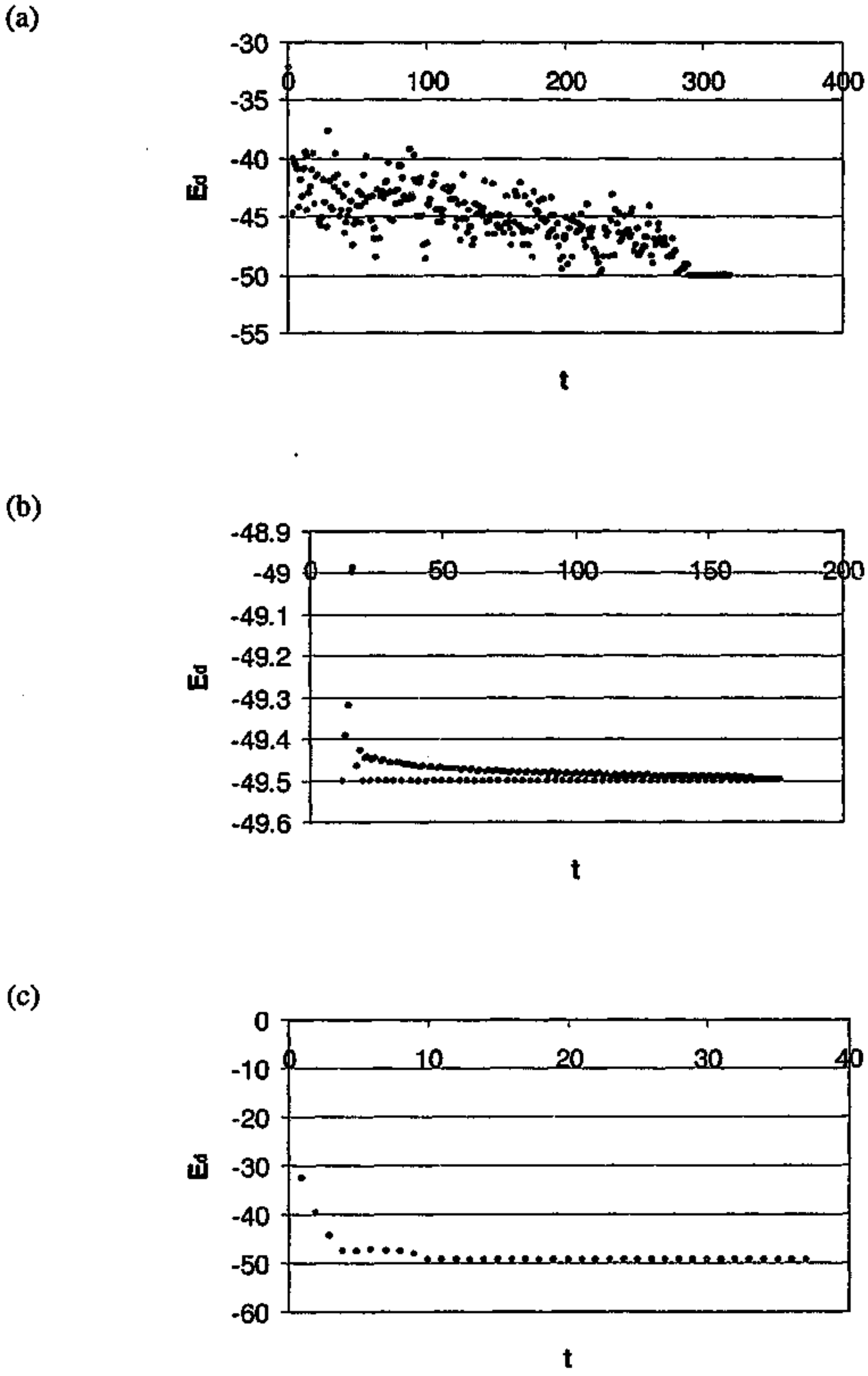


Figure 4.9: Time evolution of E_d with τ = 0.5 and different values of $\Delta t(0)$. (a) $\Delta t(0) = 0.7$. (b) $\Delta t(0) = 0.5$. (c) $\Delta t(0) = 0.35$.

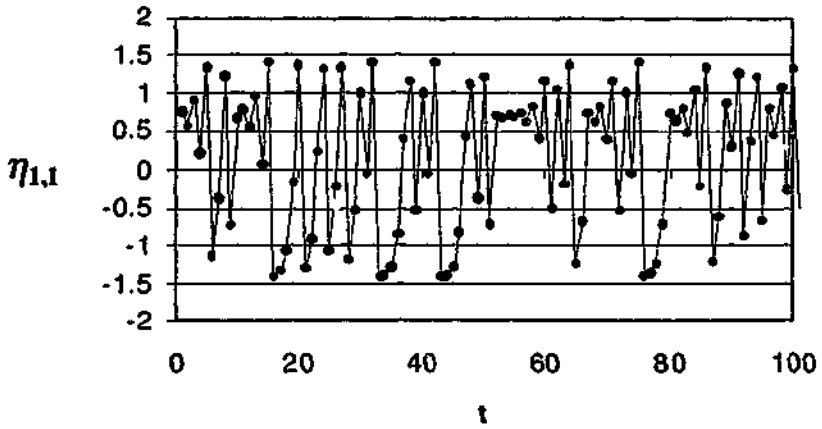


Figure 4.10: A typical chaotic time series generated from the normalized logistic map with bifurcation parameter $a = 4$.

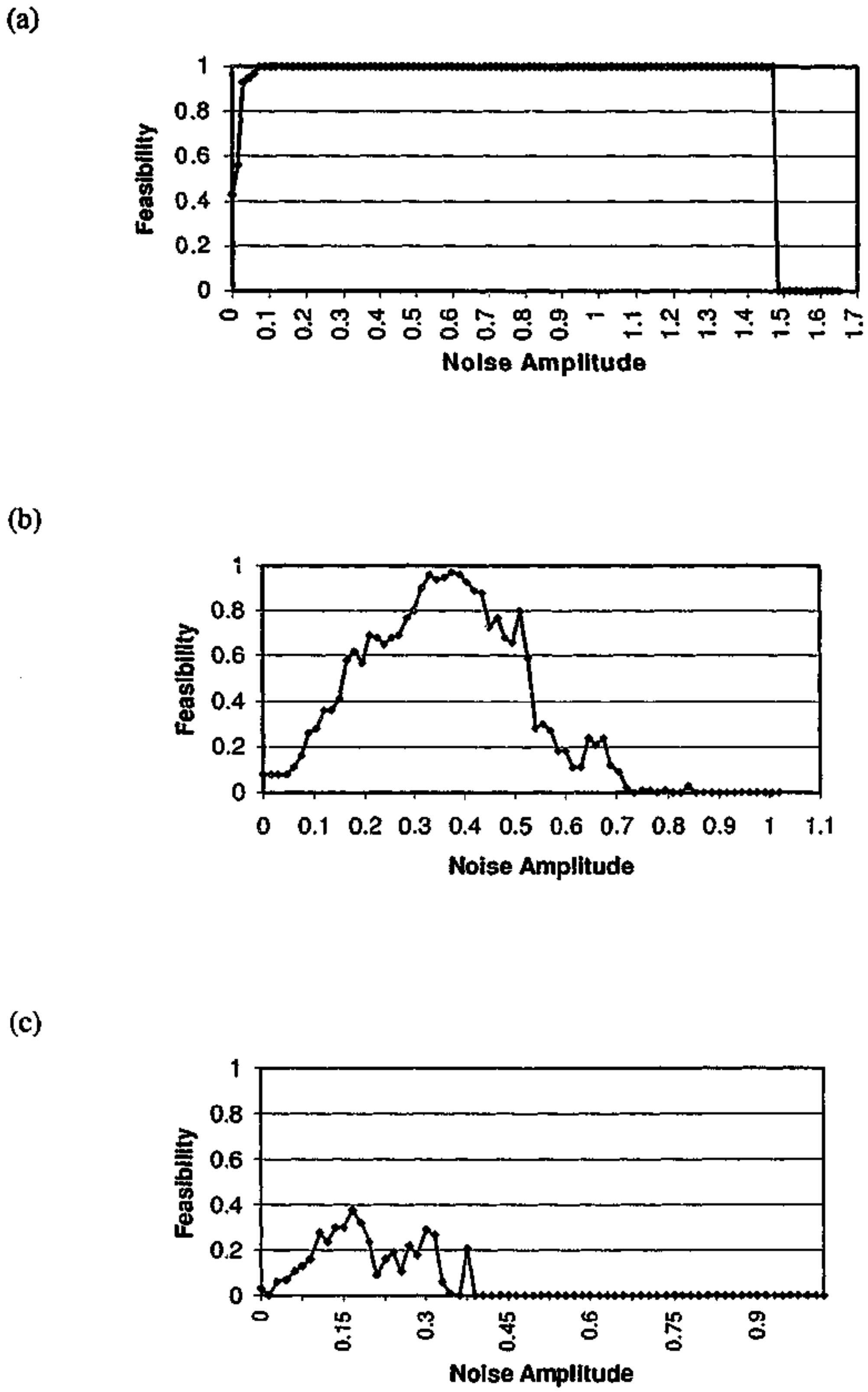


Figure 4.11: Feasibility achieved in solving the *N*-queen problem by the chaotic noise model with varying noise amplitudes. (a) $N = 5$; (b) $N = 10$; and (c) $N = 20$.

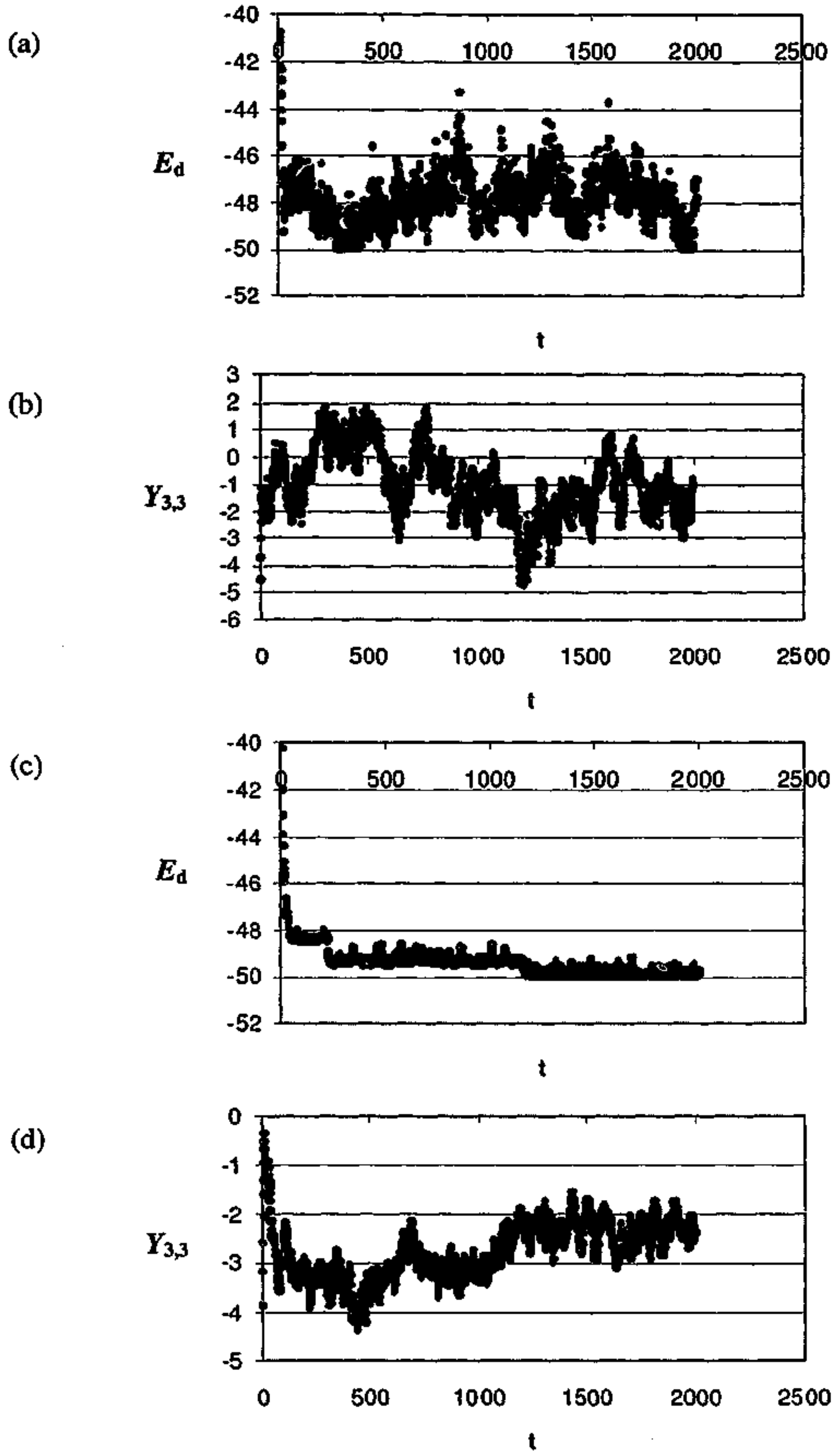


Figure 4.12: Time evolution of E_d and $Y_{3,3}$ of the chaotic noise model solving the 10-queen problem, with noise amplitude of 0.36 in (a), (b) and 0.15 in (c), (d).

Chapter 5

Self-Organizing Approaches to Combinatorial Optimization

5.1 Introduction

In this chapter, we discuss self-organizing approaches to solving COP's. The principle of self-organization in Kohonen's self-organizing feature map (SOFM) [62] lies at the heart of these approaches for the solution process. Being a feed-forward neural network, the SOFM relies on the presentation of input patterns as well as a competition based learning mechanism. It is thus clear that any optimization approach incorporating self-organizing or SOFM related principles would be quite different from the Hopfield-type neural network approaches. In Section 5.2.1, an introductory account of the SOFM is given to outline its major characteristics.

As in the case of Hopfield-type approaches, the TSP has been a major benchmark problem in the literature of self-organizing techniques for combinatorial optimization. But for the more traditional self-organizing approaches, the TSP or other problems defined in the Euclidean plane are the *only* problems they can solve. This is mainly due to the fact that these methods derive from the *elastic net* method [30]. This method was the pioneering work of Durbin and Willshaw, and was one of the very first self-organizing approaches designed to solve the TSP in the Euclidean plane. The method relies on an imaginary ring of nodes defined in the Euclidean plane. The ring can be "stretched" or deformed in the solution process such that each node matches with exactly one city, thus forming a valid tour. Since the existence and operation of such an elastic net are only meaningful in the Euclidean plane, it cannot be used to solve other non-Euclidean or general "0-1" COP's. The details of the method are given in Section 5.2.2.

Since the elastic net method, there have been other self-organizing approaches developed to solve the TSP. Many of these approaches employ the same idea of matching nodes to cities geometrically. Instead of a deformable ring, Fort [37] apply Kohonen's SOFM onto a circular array of nodes. The topographical ordering property of the SOFM is used to form a valid tour. It was found that the solution quality is affected by the problem of nodes being associated with more than one city at some stage of the solution process. Discussed in Section 5.2.3 are many techniques proposed by various researchers to tackle this problem, such as the creation/deletion of nodes [11], a *conscience* mechanism to penalize *greedy* nodes [22, 21], or by including explicit statistics into the learning [13]. However, the fundamental geometrical limitations of these methods prohibit their application to more general COP's.

Based on the SOFM, Smith [87] proposed a generalized self-organizing neural network (SONN) for solving a broad class of 0-1 optimization problems. Since it is not based on the elastic net method or other geometrical techniques, it is purely combinatorial in nature. A hybrid neural network incorporating both the generalized SONN and the Hopfield network was also proposed [90, 91]. These general approaches are reviewed in Section 5.3.1.

It has been observed in the generalized SONN that the solution quality is affected by the convergence process, in which unwanted oscillations sometimes occur [91]. In order to suppress the oscillations, various methods have been employed, such as explicit penalization of cheaper nodes [90, 91] and the conscience mechanism [45]. Of particular importance is the weight normalization procedure for constraint satisfaction [45], in which a process similar to simulated annealing is used. An overview of these techniques is presented in Section 5.3.2.

In subsequent chapters of this thesis, we will examine the SONN with weight normalization (SONN-WN) from a dynamical systems perspective, and reveal the role of nonlinear system dynamics (NSD) in optimization improvements. Thus, this chapter serves as a review of the relevant literature for subsequent chapters.

5.2 Traditional self-organizing approaches

In this section, we present an overview of traditional self-organizing approaches to combinatorial optimization. As the foundation of these approaches, Kohonen's SOFM [62] is first introduced. We then examine the elastic net method by Durbin and Willshaw [30] for solving the TSP, which is a geometrical approach derived from mechanisms of topographically ordered neural mappings. This is followed by a discussion on related efforts from various researchers to combine this method with SOFM and other approaches.

5.2.1 Self-organizing feature maps

Kohonen's SOFM belongs to a class of neural networks based on *competitive learning*, in which the neurons compete with each other to facilitate the learning process. As in other unsupervised learning algorithms such as the adaptive resonance theory model of Grossberg and Carpenter [43], the neural network can learn features from the input data without guidance. The main characteristics of SOFM is its ability to create globally ordered maps in the layered network by extracting important features from the inputs. The correspondence between the input features and their relative positions mapped onto the network is a result of the self-organizing process prescribed by the SOFM algorithm. Such a feature map also shares many similarities with the topographic feature maps found in the brain, suggesting its functional role in biological information processing [14]. The SOFM can be regarded as a nonlinear generalization of the principal component analysis, with the main application of visualization and clustering of high-dimensional data in a low dimensional space [63, 83].

Figure 5.1 shows a schematic diagram of a feature map. A sheet of nodes (neurons) forms the low dimensional space where the response of the SOFM is registered. When a typically high dimensional input pattern (represented by vector x) is presented to the SOFM, a local response belonging to a node i is triggered. The topographic or distance relations between different input patterns are reflected, or mapped, onto the local responses registered by the nodes, with responses from neighboring nodes preserving the topographic relations. This is how the features of high-dimensional inputs are *topographically mapped* onto a low-dimensional space. Although the interconnections between the nodes and the inputs are not shown in Figure 5.1, such a mapping is facilitated by weights connecting the inputs and the nodes. A self-organizing process in the SOFM algorithm measures the *similarity* of an input pattern to the weight vector W_i of each node i with some metric, and then modifies the strength of association between the input and the nodes by adjusting the weights. Such a process is performed in parallel among nodes, and can be implemented on parallel computational devices. As will be seen later in this thesis, such a weight adjusting self-organizing process is rich in nonlinear system dynamics, which have important implications to the functionality of the SOFM.

Mathematically speaking, the SOFM depicted in Figure 5.1 exists on an array of nodes. All nodes are connected to the input in parallel, and each node i is connected to the input vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T$ with weight $W_{ij}, j = 1, 2, \ldots, N$. It is convenient to view $\boldsymbol{W_i} = (W_{i1}, W_{i2}, \ldots, W_{iN})^T$ as a *reference vector* associated with each node *i*. To determine the mapping between a particular input pattern and a node, input patterns are randomly selected and then presented to the network. Each reference vector is compared to the input vector to determine the winning node i_0 of the best match according to

$$\left\| x - W_{i_0} \right\| = \min_i \left\{ \left\| x - W_i \right\| \right\} \tag{5.1}$$

where $\|..\|$ is the Euclidean distance (the dot product of $\boldsymbol{x}$ and $\boldsymbol{W_i}$ is sometimes used instead). Learning then takes place when the weight vectors for all neurons are updated according to the *feature map update rule* [62]

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t) + \alpha(i_0, i, t)\left[\boldsymbol{x}(t) - \boldsymbol{W}_i(t)\right] \tag{5.2}$$

where $\alpha(i_0, i, t)$ is the neighborhood function, with $\alpha(i_0, i, t) \rightarrow 0$ as $t \rightarrow \infty$. The neighborhood function controls the magnitude of updating to W_i's, with weaker updates for nodes further away from the winning node i_0. Upon repeated input presentations and weight updates, the neighborhood function allows the formation of a feature map, in which nodes responding to similar input patterns lie topographically close to each other in the array. A common choice for the neighborhood function is the Gaussian function

$$\alpha(i_0, i, t) = \beta(t) \exp\left(- \frac{\left\| r_{i_0} - r_i \right\|^2}{2\sigma^2(t)} \right) \tag{5.3}$$

where $0 < \beta(t) < 1$ is the learning rate, r_{i_0} and r_i are the position vectors of the winning node i_0 and i respectively, and $\sigma(t)$ is the parameter controlling the width of the Gaussian function. Figure 5.2 illustrates the relationship between the winning neighborhood and node positions. A hexagonal array of nodes is chosen [62] around the winning node i_0. The neighborhood of the winning node consists of nodes within the index set $N_{i_0}(t)$. Initially a relatively large $N_{i_0}(t)$ is chosen. As time increases ($t_1 < t_2 < t_3$), the neighborhood is decreased in order to sharpen the feature map. If the initial $N_{i_0}(t)$ is chosen to be too small, discontinuously ordered subgroups are formed instead of a topographically ordered feature map.

The learning rate $\beta(t)$ should be chosen close to 1 at the start of the iterations, and then gradually decreased for the fine adjustment of the map [63]. It should remain small after the ordering phase. The actual rule for decreasing $\beta(t)$ is not crucial, unless for very large maps. Similarly, a relatively large $\sigma(t)$ is usually chosen during the initial ordering phase for a more effective ordering process, and then gradually reduced. It is important that $\sigma(t)$ is large enough such that the neighborhood function $\alpha(i_0, i, t)$ is convex for nodes within $N_{i_0}(t)$. Otherwise, metastable states due to "topological defects" (or conflicts between locally ordered groups) may result [32].

It is well known that rigorous mathematical proofs of convergence for the SOFM algorithm are hard to establish [63]. For the one-dimensional case, Cottrell and Fort [26] proved the convergence for uniformly distributed stimuli, which was later extended to general distributions by Bouton and Pagès [19]. A dynamical system approach using the ODE method was used by Fort and Pagès [38] to prove the uniqueness of an equilibrium point after self-organization, together with convergence results with a general class of stimuli distributions and neighborhood functions. By using the Fokker-Planck equation to describe the SOFM learning process, Ritter and Schulten [84] studied the stability and fluctuations of the equilibrium map, and provided a criterion of convergence to the equilibrium map in terms of the time dependent learning step size. For the two-dimensional case, a restricted proof of convergence is given by Erwin *et al* [33]. They also proved that although the SOFM algorithm cannot be derived from energy functions (in contrast to Hopfield-type networks), a set of potential functions (one for each neuron) can be constructed, which are independently minimized with stochastic gradient descent.

Apart from the SOFM, there are other similar models related to the formation of topographical mappings between different dimensions. Derived from one of those models is the elastic net method for solving the TSP, which represents a major step in solving COP's with self-organizing approaches.

5.2.2 The elastic net method

In 1987, Durbin and Willshaw [30] proposed an optimization technique called the elastic net method to solve the TSP. It is a parallel analogue algorithm derived from a model of topographically ordered projections in the brain. Being a geometrical method, a TSP tour is viewed as a mapping between the cities on the plane and the same number of nodes on a circle. Such a circle resembles an elastic band in the sense that it is initially situated in the middle of the plane, but is gradually stretched nonuniformly to pass through all cities and become a path (see Figure 5.3). An energy function can be defined as follows,

$$E = -\alpha K \sum_{i=1}^{N} \ln \sum_{j=1}^{M} \exp\left(-\frac{\left| x_i - y_j \right|^2}{2K^2} \right) + \beta \sum_{j=1}^{M} \left| y_j - y_{j+1} \right|^2 \tag{5.4}$$

where x_i represents the position of city i ($i = 1, 2, \ldots, N$), y_j represents the position of node j ($j = 1, 2, \ldots, M$) on the elastic band, and α and β are constants. Since the length parameter K is gradually reduced, the first term of E means that the path must pass through all cities for E to remain bounded. As E reaches a minimum, so does its second term which represents the length of the path. For each iteration of the algorithm, the positions of the nodes are updated by

$$\Delta y_j = -\alpha \sum_{i=1}^{N} W_{ij} \left(x_i - y_j \right) + \beta K \left(y_{j+1} + y_{j-1} - 2y_j \right) \tag{5.5}$$

$$= -K \frac{\partial E}{\partial y_j} \tag{5.6}$$

where

$$W_{ij} = \frac{\exp\left[-\left| x_i - y_j \right|^2 / (2K^2) \right]}{\sum_{k=1}^{M} \exp\left[-\left| x_i - y_k \right|^2 / (2K^2) \right]}. \tag{5.7}$$

From (5.6) it can be seen that the updating in (5.5) represents a gradient descent on the energy landscape. As $K \to 0$, the algorithm converges to a stable local minimum as $\Delta y_i \to 0$.

The elastic method was applied to solve the 30-city TSP by Durbin and Willshaw [30], and obtained the shortest known tour in 1000 iterations. This was better than

Hopfield and Tank's result on the same set of cities [54], whose shortest tour was 19% longer. A 100-city problem was also solved with their best tour within 1% of the best tour obtained by any method. The method has the advantages of having high quality feasible solutions, parallel computation, good scalability, and is not too sensitive to parameter values. A recent efficiency improvement using hierarchical strategies has also been proposed [102]. However, the elastic net method is limited to solving general TSP's in the Euclidean plane only, and cannot solve the random TSP (whose distance matrix is non-symmetric and contains randomly generated elements) or other problems not defined in the Euclidean plane.

5.2.3 Self-organizing approaches to the TSP

In 1988, Fort [37] proposed a self-organizing approach based on Kohonen's SOFM to solve the TSP. It is similar to the elastic net method in the use of topographically ordered mappings for the formation of tours. A circular array of neurons is used in this approach in order to map the cities of the TSP, such that a tour is represented by the ordering of the mapping. In contrast to the deterministic nature of the elastic net method, stochasticity is used in this approach for weight updates during the learning process. Experimental results obtained with this method using Hopfield and Tank's TSP data set are not as good as those obtained by the elastic net method in general, with many local minima solutions.

By incorporating the elastic net concept into Kohonen's SOFM implementation, Angéniol *et al* [11] were able to obtain improved results on the TSP over the elastic net method. In this method, nodes are linked together by a ring and are assigned coordinates in the Euclidean plane. Coordinates of the cities are presented as input vectors in the SOFM algorithm. A competition among the nodes is held to decide the winning node, which is the one closest to the city being presented as input. The coordinates of the nodes within the winning neighborhood are then updated, pulling them toward the particular city. The iterative process continues until the total number of nodes is equal to the number of cities, with the resulting tour given by the ring of nodes. One important feature of this method is that the total number of nodes can be changed via a node creation/deletion process. Initially there is only one node located at the origin of the plane. A node is duplicated when it has become the winner of two cities, and the new node is assigned as a neighbor of the original. Both these nodes are not to be updated for the next competition. A node is deleted when it has not been a winner for three complete presentations of all cities.

Using this method, Angéniol *et al* first solved the 30-city problem with Hopfield and Tank's data set [54]. A good result of less than 3% greater than the optimum was obtained on average. The authors also solved the same 50-city problems from Durbin and Willshaw [30]. Comparable results to the elastic net method were obtained, but with improved efficiency. A feasible solution was also obtained for a 1000-city problem in 20 minutes on a workstation computer (the best solution took much longer however).

Favata and Walker [34] also proposed a similar approach of implementing the SOFM principle to solve the TSP. The mapping between a ring of nodes and the cities on the

Euclidean plane is also adopted, but without the node creation/deletion process as employed ty Angéniol *et al* [11]. Although some nodes as a result may claim more than one city and some cannot claim any, their solutions using Hopfield and Tank's 30-city TSP data set [54] were only around 5% more costly on average than those by using simulated annealing [61]. Good scalability and efficiency were also acknowledged by the authors.

The *guilty net* proposed by Burke and Damany [22] is another SOFM based approach to solve the TSP without node creation/deletion processes. As mentioned above, such processes are designed for the purpose of node separation when a node claims more or fewer than one city at the end of the training. Since the number of nodes is fixed in the guilty net, a *conscience mechanism* introduced by DeSieno [29] was employed to achieve feasible tours efficiently, thus in effect replacing the node creation/deletion process. A bias term is defined such that nodes winning too often are restrained for further excessive wins (hence the name *guilty*):

$$bias_j = \frac{win_j}{1+\sum_{k=1}^{N} win_k} \tag{5.8}$$

where win_j is the number of accumulated wins for node j. Such a bias term is added to the calculation of the Euclidean distance between the input vector and the weight vector of node j. Clearly a large bias term would lower the chance of a node to win the competition for the minimum distance. Experimental results obtained by the authors in solving the TSP of 10, 30, 50 and 100 cities show that they are on average 9.5% poorer than the elastic net method, and 2.2% poorer than simulated annealing. However, this method is more efficient because of its ability to obtain approximate tours before complete node separations (each node distinctively claiming only one city).

More recently Burke [21] proposed the *vigilant net* as an extension of the guilty net. This development aims to alleviate the trial-and-error tuning of the effective bias term weight in the guilty net, which affects the solution quality. The aim was to devise an automated mechanism for tackling the node separation issue without time-consuming parameter tuning. The vigilance parameter concept borrowed from adaptive resonance theory [43] was used to resolve this issue. The vigilance parameter determines the maximum number of wins for a node before it is reset, and the competition restarted with other uncommitted nodes. Computational comparisons between the guilty net and the vigilant net show encouraging results for the latter. For nonuniformly distributed TSP's, tours obtained with the vigilant net are as good or sometimes better than the guilty net. Although the performance of the vigilant net on uniformly distributed TSP's is not as good as the guilty net, the author acknowledged that the difference appears to become smaller as problem size increases. A performance comparison with the space-filling curve heuristic by Platzman and Bartholdi [80] showed the vigilant net can achieve comparable solution quality on average.

In 1999, Aras *et al* [13] proposed the Kohonen Network Incorporating Explicit Statistics (KNIES), so called because of its ability to preserve and utilize statistical information of the data points. The motivation was to include global information (statistics) from the data points at every iteration of the learning process, in contrast with the SOFM where the nodes can only learn such information *asymptotically*. Just like other approaches to the TSP, a ring of nodes is used to gradually move toward the cities on the Euclidean plane. In the KNIES, the mean of the positions of the cities is the statistical quantity being captured and assimilated into the node positions. This is done by moving the nodes outside the winning neighborhood in such a way that the mean of all the nodes matches the mean of the cities. According to the authors, the usefulness of such a scheme is due to the dependence of the final solution quality on the transient stochastic distribution of the nodes on the ring. The method was used to solve the TSP of various sizes from 51 to 532 cities. By comparing the tour lengths obtained by KNIES and other self-organizing approaches mentioned above, it was found that KNIES gives the best result for most of the problem instances. The method was also implemented by the same authors to solve the Hamiltonian path problem [10].

All the self-organizing approaches described so far are limited to solving the TSP in the Euclidean plane only, and there are other similar approaches as reviewed by Potvin [81]. The limitation is due to the common adoption of the elastic net idea of matching a ring of nodes to the cities in the Euclidean plane. In order to generalise the self-organizing approaches for broader problems like non-Euclidean TSP or other general COP's, a more general way to implement the SOFM principle is needed.

5.3 Self-organizing neural networks for general COP's

In this section, we review self-organizing neural networks (SONN's) that solve general 0-1 optimization problems. The main feature of these SONN's is that their computations do not rely on nodes confined to the Euclidean plane, but on weights which lie in multi-dimensional space. These weights linking the inputs and the nodes now represent the variables of the optimization problems. Within this class of SONN's, we outline various developments which differ in their ways of enforcing solution feasibility and improving solution quality.

5.3.1 The generalized self-organizing neural network

The original form of the generalized SONN that is capable of solving general 0-1 COP's was proposed by an Australian researcher Kate A. Smith [87] in 1995. The generalized quadratic assignment problem was used as an example for illustration:

$$\text{minimize} \qquad f(\mathbf{X}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{l=1}^{M} p_{ik} q_{jl} X_{ij} X_{kl}$$

$$\text{subject to} \qquad \sum_{j=1}^{M} X_{ij} = 1 \qquad \forall i = 1,\ldots,N$$

$$\sum_{i=1}^{N} X_{ij} = d_j \qquad \forall j = 1,\ldots,M$$

$$X_{ij} \in \{0,1\} \tag{5.9}$$

where **X** is the solution matrix. For the case of $d_j = 1$ for all j, any **X** satisfying the constraints in (5.9) can be regarded as a permutation matrix whose rows are permutation of the row vectors: (1, 0, 0,..., 0), (0, 1, 0,..., 0),..., (0, 0, 0,..., 1), etc. The basic structure of the SONN consists of an input layer and an output layer, where lie the input nodes and the output nodes respectively, as shown in Figure 5.4. Connecting the two layers are the weights denoted by W_{ij}, which is a continuous variable representing the probability that the variable in the *i*th row and *j*th column of the solution matrix **X** is 'on'. For the learning process of the SOFM, row vectors described above are presented to the input layer. A cost is then calculated for each node at the output layer according to the COP being solved. For the example in (5.9), the cost (or potential) of node *k* for input vector *x* (such that $x_{j*} = 1$) is given by

$$V_{k,j^*} = \sum_{i=1}^{N}\sum_{j=1}^{N} p_{ik} q_{jj^*} W_{ij} + \beta_V \sum_{i=1,i\neq k}^{N}\ \sum_{j=1,j\neq j^*}^{N} \left(W_{ij^*} + W_{kj}\right) \qquad (5.10)$$

where β_V is a parameter for controlling oscillations in the learning process. The winning node denoted by index i_0 is defined as the node having the lowest cost for a particular input presentation:

$$V_{i_0,j^*} \le V_{i_k,j^*} \qquad \text{for all other nodes } k.$$

A winning neighborhood is also defined:

$$V_{i_0,j^*} \le V_{i_1,j^*} \le V_{i_2,j^*} \le \ldots \le V_{i_\eta,j^*}$$

where η is the size of the neighborhood. So unlike other self-organizing approaches based on the elastic net where the winning neighborhood is defined spatially, this approach operates in the cost potential space. Because of this, the SONN can solve generalized 0-1 COP's and is not limited to Euclidean problems. Once the winning neighborhood has been determined, the weights are updated according to Kohonen's SOFM updating rule. In order to retain the weights on the constraint plane after updating, the weight matrix's projection on the constraint plane is taken as the new weight matrix for the next round of learning. In an illustrative numerical example of solving the quadratic assignment problem (QAP), the author showed that reasonably good results were obtained by the SONN. The global minimum was obtained in 3 out of 10 random starts of the algorithm, with good local minima for the remaining.

More recently, Smith *et al* [90, 91] extended the generalized SONN described above, and proposed a hybrid neural approach to combinatorial optimization. It combines the generalized SONN and the Hopfield network into a design that eliminates infeasible solutions, in addition to its capability of solving a broad class of 0-1 optimization problems. There are two stages in the algorithm, in which the generalized SONN described above is the first stage. The role of the first stage is to minimize the objective function, while the second stage is to ensure the solution remains feasible and is directed toward a vertex. A Hopfield network is used in the second stage to enforce solution feasibility. An energy function consisting of only a single constraint term and a 0-1 enforcement term is minimized with the Hopfield network. Without any term in the energy function representing the objective function, the trade-off problem between constraint terms and the objective function term was alleviated. A set of theoretical remarks was also proposed to address the convergence of the scheme [91].

The hybrid neural network was applied to solve the car sequencing problem, a postal delivery problem (a p-hub location-allocation problem) [90, 91], the static and dynamic channel assignment problems [89], etc. Comparative results with simulated annealing and a commercial integer programming package (GAMS/MINOS-5) showed that the hybrid network can match, and sometimes outperform the other two methods in solution quality. The authors also anticipated the accelerated processing speed of the neural approach brought about by recent developments in parallel hardware, which makes the network an effective alternative to heuristics in many practical applications.

In both the generalized SONN and the hybrid neural network, the parameter β_V in the cost potential as given in (5.10) needs to be adjusted. Such a parameter is used to encourage constraint satisfactions, and a suitable choice of its value is problem dependent. It was found that the parameter affects convergence of the network, in the sense that oscillations may occur with an unsuitable value [91]. Such an oscillation is analogous to the problem associated with self-organizing approaches for the TSP discussed in Section 5.2.3, where a particular node wins too often and results in claiming more than one city. Although the COP's being tackled by the generalized SONN and the hybrid network are not Euclidean in general, the same phenomenon of a node winning too often still occurs, resulting in oscillations in an otherwise well-behaved convergence toward the vertex of the hypercube. The value of β_V was chosen by Smith *et al* [90, 91] with a problem-dependent expression in such a way that a *greedy* node (a node whose cost potential is inherently lower than others because of the problem specification) is not allowed to win unless its cost potential is significantly cheaper than others. Since such a method of parameter tuning is problem dependent and may hinder the deployment of the networks as a readily applicable optimization tool, a more general approach is needed.

5.3.2 SONN with weight normalization

In order to control oscillations in the convergence process of the generalized SONN and the hybrid network, Guerrero *et al* [45] incorporated the *conscience* mechanism in the determination of the cost potential constraint parameter (β_V). Such a mechanism as discussed in Section 5.2.3 has been proposed to solve the problem of greedy nodes among self-organizing approaches for solving the TSP [22, 21]. In their paper, the quadratic assignment problem is solved as an example (same as (5.9) except $d_j = 1$ and $N = M$), in which the cost potential of node i is given by

$$V_{ik^*} = \sum_{j=1}^{N}\sum_{l=1}^{N} p_{ij} q_{k^*l} W_{jl} + \beta_V(i) \sum_{l=1,l\neq i}^{N} W_{lk^*} \qquad (5.11)$$

where $\beta_V(i)$ is the β_V value of node i. According to the generalized SONN algorithm, the winning node is the one having the lowest V_{ik^*} when an input is presented to the network. Instead of using trial-and-error or an *ad hoc* formula to work out the suitable value of β_V, a scheme based on the conscience mechanism was introduced by the authors:

$$\beta_V(i) = \frac{f}{N} win(i) \qquad (5.12)$$

where f is the objective function to be optimized, N is the number of output nodes in the network, and $win(i)$ is the number of wins for node i in the current epoch of input presentations. The function of (5.12) is to penalize nodes that win more times than the others, so as to encourage the claiming of only one input pattern by each node, thus suppressing oscillations.

In the same paper, the authors also introduced another method for constraint satisfaction. In order to explicitly enforce the constraints

$$\sum_{j=1}^{N} X_{ij} = 1 \quad \forall i \qquad \text{and} \qquad X_{ij} \in \{0,1\} \tag{5.13}$$

a normalization procedure for the weights is used:

$$W_{ij} \leftarrow \frac{\exp\left[-\left(1-W_{ij}\right)\big/T\right]}{\sum_{k=1}^{N}\exp\left[-\left(1-W_{ik}\right)\big/T\right]} \tag{5.14}$$

where T is a parameter akin to the temperature parameter in simulated annealing, which is gradually lowered as the learning process proceeds. Since the weight W_{ij} represents the continuous relaxation of the decision variable X_{ij}, the constraints in (5.13) are satisfied when T becomes asymptotically small. It should be noted that a very similar normalization procedure was proposed by Van den Bout and Miller [103, 104] for constraint satisfaction in their Hopfield network approach to the TSP.

By controlling oscillations and enforcing constraint satisfactions with the respective methods described above, Guerrero *et al* applied the generalized SONN to solve the facilities location problem [45] and a cellular manufacturing problem [44]. It was shown that the introduced methods of improvements were effective to obtain feasible and reasonably good quality solutions. However, there are questions still to be answered as to how the annealing scheme of the normalization procedure affects the solution process, and its ultimate consequence on the feasibility and solution quality.

5.4 Concluding remarks

In this chapter we have reviewed the development of self-organizing approaches for solving the TSP and more general COP's. From the Euclidean based approaches for solving the TSP to the generalized SONN, we have seen the effectiveness of the SOFM principle in combinatorial optimization. Throughout the literature it has been suggested that the solution quality of self-organizing approaches rely on the quality of the corresponding competition-based learning process. For this a variety of techniques have been discussed to regulate the competition for input patterns (or cities) among nodes, or in other words, to control oscillations in the convergence process. For the SONN to be a versatile and flexible optimization tool, we find the weight normalization procedure employed by Guerrero *et al* [45] a promising approach for further development. The normalization process achieves constraint satisfaction via an annealing scheme such that "decision making" becomes a gradual process. Since the relationship between annealing schedules (initial T and its *cooling* rate) and solution quality remains an important open question, a detailed investigation is needed. In the next chapter, we study such a relationship by solving the N-queen problem via computer simulation, which also reveals the intricate roles played by nonlinear system dynamics and noise in the self-organization scenario.

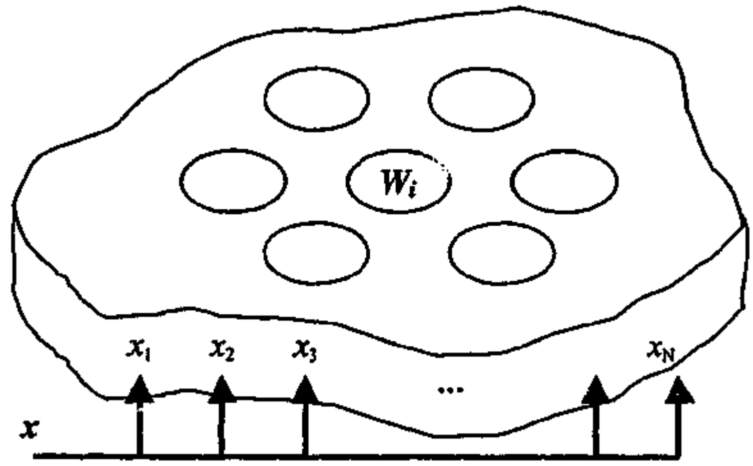


Figure 5.1: Schematic diagram of a feature map showing a two-dimensional array of neurons (nodes) with weight **W** and input vector x.

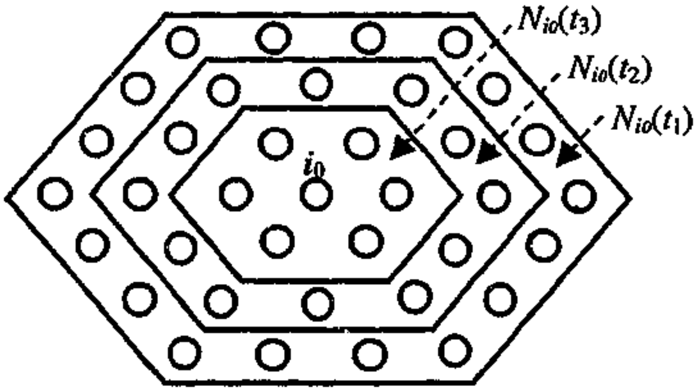


Figure 5.2: A Hexagonal array of nodes showing the time-dependent sets of nodes included in the winning neighborhood.

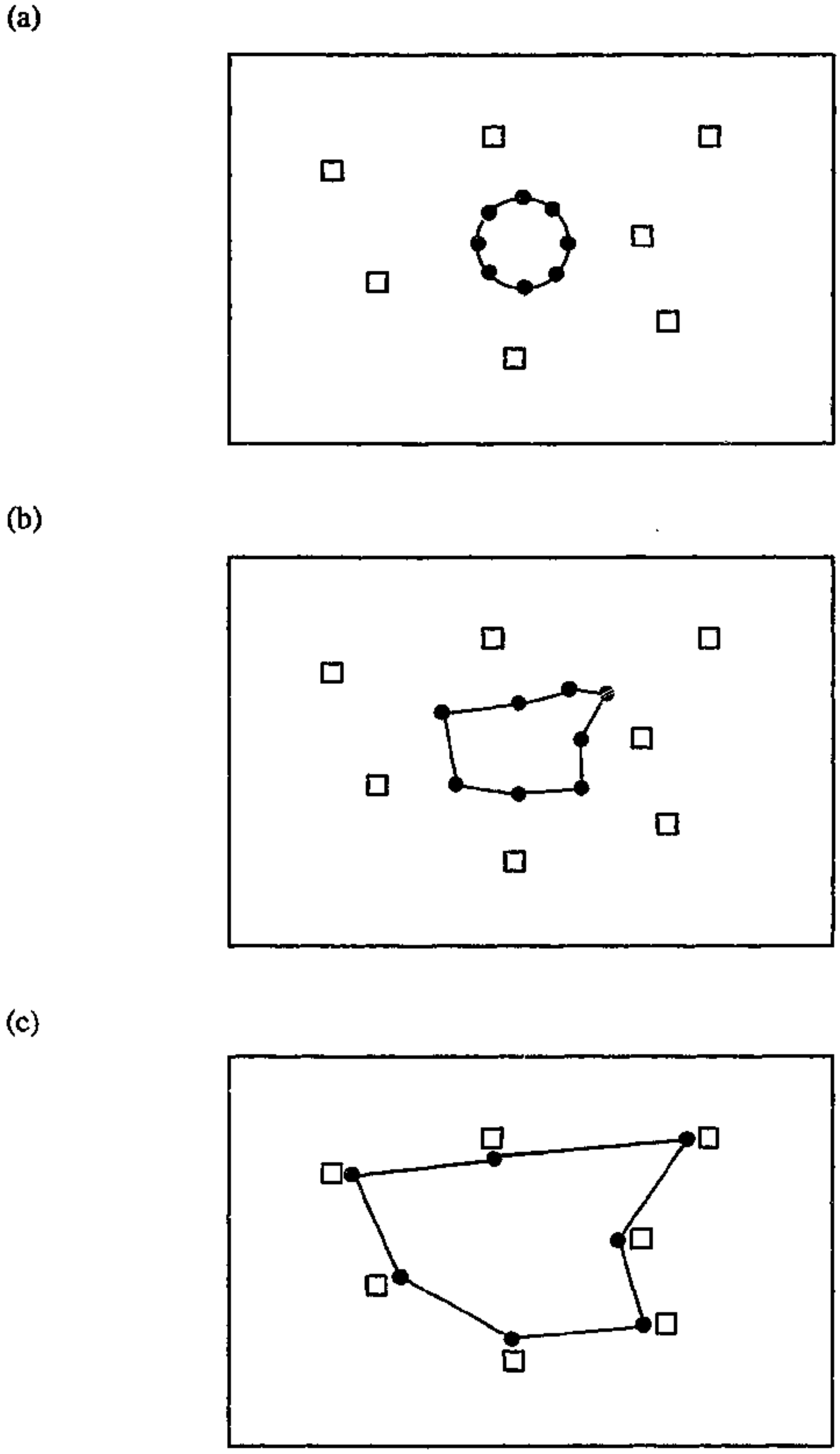


Figure 5.3: An example of the progress of the elastic net method for a 7-city TSP over time: (a) → (b) → (c). Nodes are represented by •, and cities are denoted by □.

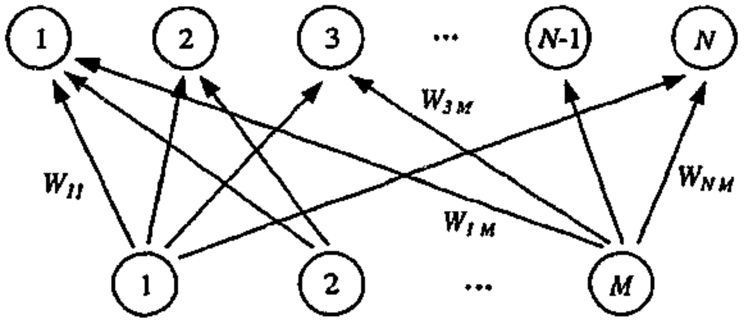


Figure 5.4: Architecture of the SONN.

Chapter 6

Computational Performance of the Self-Organizing Neural Network with Weight Normalization

6.1 Introduction

In this chapter, we present the computational results of applying the self-organizing neural network with weight normalization (SONN-WN) to solve a COP. Our aim is to investigate the theoretical basis behind the effective optimization ability of the SONN-WN. We achieve this by implementing the neural network model to solve the *N*-queen problem via computer simulation, and then study its optimization performance from the perspective of characteristic network dynamics. The

SONN-WN algorithm together with its implementation for the *N*-queen problem is presented in Section 6.2.

Since the weight normalization procedure within the SONN-WN is the focus in our study for improved convergence toward effective optimization, we seek to gain a deeper understanding on how solution qualities are affected by its various annealing schedules. For this we plot the obtained feasibilities in the normalization parameter space, and study the distribution of high feasibility regions. An important facet of this investigation is the effect of improved feasibilities and robustness due to random noise added to the cost potentials, which reveals interesting band patterns in the normalization parameter space. The influence of other important parameters in the self-organization process is also studied. These results are presented in Section 6.3.1 with detailed discussions.

In order to study the complex relationship between the weight normalization process, noise, and the resultant optimization quality, we present in Section 6.3.2 some experiments detailing the time evolution dynamics of the cost potentials (V_{ij}) and the weights (W_{ij}). By studying how unwanted oscillation during convergence can be controlled by random noise added to the cost potentials, we explain the improved feasibilities obtained in Section 6.3.1. The interaction between noise and weight normalization is also investigated, which highlights the importance of the temperature parameter T in the convergence process.

By combining all the major observations and findings in Section 6.3.1 and 6.3.2, we present in Section 6.3.3 a guide for choosing the most effective annealing schedules for optimization results. A detailed account on the structure of feasibility patterns obtained in Section 6.3.1 in terms of convergence dynamics is also given. Finally, we close this chapter with the concluding remarks in Section 6.4.

6.2 A SONN-WN approach to the *N*-queen problem

In this section, we present the SONN-WN formulation and its implementation to solve the *N*-queen problem. As in Section 4.2, the chessboard is represented by the $N \times N$ matrix **X**, with the decision variable $X_{ij} \in \{0, 1\}$ denoting the position of queens. The overall cost (*f*) of violating the *N*-queen constraints (4.1) remains the same as in (4.2):

$$f = \frac{A}{2}\sum_{i,j=1}^{N}\sum_{l=1,l\neq j}^{N} X_{ij}X_{il} + \frac{B}{2}\sum_{i,j=1}^{N}\sum_{k=1,k\neq i}^{N} X_{ij}X_{kj} + \frac{C}{2}\left(\sum_{i,j=1}^{N} X_{ij} - N\right)^2$$
$$+\frac{D}{2}\left(\sum_{i,j=1}^{N}\sum_{\substack{p\neq 0\\ 1\leq i-p\leq N\\ 1\leq j-p\leq N}} X_{ij}X_{i-p,j-p} + \sum_{i,j=1}^{N}\sum_{\substack{p\neq 0\\ 1\leq i-p\leq N\\ 1\leq j+p\leq N}} X_{ij}X_{i-p,j+p}\right) \tag{6.1}$$

To solve the *N*-queen problem of minimizing (6.1), we formulate a SONN-WN based on that used in [45] where a quadratic programming problem has been solved. The architecture of the SONN-WN for this problem is shown in Figure 6.1. It is a feed-forward neural network with an input layer of *N* nodes, and an output layer of *N* nodes, representing the *N* columns and *N* rows of the chessboard respectively. The weight between an input node *j* and an output node *i* is given by W_{ij} and represents the continuous relaxation of the decision variable X_{ij} in (6.1). Thus, $/_{ij}$ is the probability of placing a queen in row i, column j. For any $\mathbf{X}$ that represents a feasible solution of the N-queen problem, it can be treated as a permutation matrix whose rows belong to a permutation of the following row vectors, each of dimension N:

$$\begin{array}{ll} (1, 0, 0, \ldots, 0) & \text{for } j^* = 1 \\ (0, 1, 0, \ldots, 0) & \text{for } j^* = 2 \\ & \vdots \\ (0, 0, 0, \ldots, 1) & \text{for } j^* = N \end{array}$$

where j^* denotes the column having an element of 1. For the learning process, these row vectors are used as the input vectors that are presented at the input layer of the SONN-WN (Figure 6.1). The task for the SONN-WN is to find the cheapest permutation of these N rows. For each input pattern presented to the network, a cost potential V is calculated for each node i in the output layer:

$$V_{ij^*} = \sum_{k,l=1}^{N}(1-\delta_{ik})(\delta_{i+j^*,k+l}+\delta_{i-j^*,k-l})W_{kl} + B\sum_{k=1}^{N}(1-\delta_{ik})W_{kj^*} \qquad (6.2)$$

where

$$\delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases}$$

The cost potential V_{ij^*} in (6.2) is the diagonal and column contribution to the objective function f of placing a queen on column j^* of row i, and is related to the partial derivative of f. From the viewpoint of a permutation matrix, V_{ij^*} can be thought of as the potential cost to the objective function (6.1) if the input pattern j^* is placed in row i. B is a penalty parameter which is arbitrarily chosen to be 1 in this investigation, and determines the balance between violation of diagonal constraints and column constraints. If all W_{ij}'s are roughly the same, which is a typical initial condition of the network before any updating, the cost potential as evaluated from (6.2) reflects an inherent symmetry in the N-queen problem definition. Figure 6.2 illustrates this cost symmetry, with higher expected costs around the central region of the chessboard. This also means there are more expected clashes when placing a queen around the center. While the column and diagonal constraints are handled by the cost potential (or objective) function (6.2), the row constraints are treated separately, and are enforced by the normalization procedure described below during the weight updates.

The competition between the output nodes gives the winning node i_0 corresponding to the row with minimum potential

$$i_0 = \arg\min_i V_{ij^*} \quad . \tag{6.3}$$

For self-organization to develop we define a neighborhood of the winning node containing those nodes with the least potential

$$N(i_0, j^*) = \{i_0, i_1, i_2, \ldots, i_\eta\} \tag{6.4}$$

where

$$V_{i_0,j^*} \le V_{i_1,j^*} \le V_{i_2,j^*} \le \ldots \le V_{i_\eta,j^*} \le \ldots \le V_{i_{N-1},j^*} \tag{6.5}$$

and $\eta \ge 0$ is the neighborhood size. Thus the neighborhood is defined according to relative cost, rather than spatially as in Kohonen's SOFM [62].

Once both the winning node and its neighborhood have been determined, the weights are updated according to a modification of Kohonen's weight adaptation rule. In order to explicitly enforce the constraint of one queen in each row, the weight normalization for each output node as described in [45] is used:

$$W_{ij} \leftarrow \frac{\exp\left(-\left(1-W_{ij}\right)\big/T\right)}{\sum_{k=1}^{N}\exp\left(-\left(1-W_{ik}\right)\big/T\right)} \tag{6.6}$$

where T is a parameter (called temperature), which is lowered as the learning process proceeds. The normalization operation guarantees that when convergence is completed, only one queen is assigned to each row. To obtain a good quality solution (or feasible solution for N-queen) within an effective number of iterations, an *adequately controlled* solution process is needed. It is crucial to choose an appropriate *annealing schedule*, which comprises an initial temperature $T(0)$ and a "cooling rate" r, such that high quality solutions are obtained efficiently. As pointed out in the last chapter, the weight normalization procedure provides the basic mechanism for suppressing oscillations during convergence via its role in constraint enforcement.

During the learning process, the neighborhood size, the magnitude of the weight adaptations, and the temperature are gradually decreased. The complete algorithm follows:

1. Randomly initialize the weights around 0.5.
2. From the permutation matrix, randomly choose a row as the input vector and present it to the input layer. The j^*-th element of the input vector is 1, which means column j^* of the chessboard is being computed.
3. Compute the potential V_{ij^*} for each output node i according to (6.2).

4. Determine the winning node, i_0, as well as its neighboring nodes according to (6.3)-(6.5).

5. Update weights, W_{ij^*}, connecting input node j^* with every output node i based on Kohonen learning:

$$\Delta W_{ij^*}(t) = \alpha(i,t)(1 - W_{ij^*}) \quad \forall i \in N(i_0, j^*) \tag{6.7}$$

$$\Delta W_{ij^*}(t) = 0 \quad \forall i \notin N(i_0, j^*) \tag{6.8}$$

where

$$\alpha(i,t) = \beta(t)\exp\left[-\frac{\left|V_{i_0 j^*} - V_{i, j^*}\right|}{\left|V_{i_0 j^*} - V_{i_{N-1} j^*}\right|}\right] \tag{6.9}$$

The updated weights are:

$$W_{ij^*} \leftarrow W_{ij^*} + \Delta W_{ij^*} \tag{6.10}$$

6. Normalize W_{ij}'s to enforce the one-queen-per-row constraint using (6.6).

7. Repeat from Step 2 until all columns have been selected as input patterns. This is one training epoch. Repeat from Step 2-6 for L_1 epochs, then anneal T to encourage 0-1 solutions and decrease $\beta(t)$, according to the following:

$$T(t + L_1) = rT(t) \tag{6.11}$$

where $T(t)$ represents the temperature at epoch t, and $0 < r \leq 1$ is the cooling rate. $\beta(t)$ also decays in a similar manner with r_β.

8. Repeat Step 7 L_2 times, each time subtract η by 1. Halt when $\eta < 0$. W_{ij}'s are then rounded to 0 or 1 as follows to obtain the final solution:

$$W_{ij}^{final} = \begin{cases} 1 & if \quad W_{ij} \geq 0.5 \\ 0 & \quad\;\; W_{ij} < 0.5 \end{cases} \quad \forall i, j \tag{6.12}$$

As the normalization procedure almost always ensures the final weight are near 0 or 1, this rounding only serves to obtain strict 0-1 solutions.

6.3 Optimization performance of the SONN-WN – computational results

In this section, we present the computational results of implementing the SONN-WN algorithm to solve the N-queen problem with $N = 5$. In order to show how solution qualities are affected by the choice of annealing schedules that characterize the weight normalization process, we present in Section 6.3.1 the measured feasibilities in the normalization parameter space. We also introduce the use of external noise in the competition-based learning process, which acts as an aid to our parametric study of weight normalization, as well as being a technique to enhance solution quality in its own right. The feasibility results are then supplemented by a detailed examination of the corresponding network dynamics in Section 6.3.2, from which we investigate the combined effects of noise and weight normalization on controlling oscillations during convergence. We then present in Section 6.3.3 a practical guide on choosing normalization parameters for effective optimization, together with a detailed account on the structure of feasibility patterns observed in the normalization parameter space.

6.3.1 Improving feasibilities: A parametric study

In this section, we examine results on improved feasibilities by means of a parametric study. A feasibility measure is used to quantify the optimization ability of the network, which is defined as the proportion of globally optimal solutions obtained with 100 different initial **W**'s. For example, a feasibility of 0.8 means 80 solutions out of 100 random starts are globally optimal. A global minimum solution satisfies all the

N-queen constraints, and a local minimum is a solution with at least one constraint violated. For each parameter setting of the network, feasibility is measured in the two dimensional parameter space (r, $T(0)$). Each point in the plane represents a particular annealing scheme for the weight normalization. Another main parameter to be adjusted in our study is the neighborhood size η, which strongly affects the extent of weight updating for each node. Because of this effect and the tendency of the winner to oscillate between two equally attractive (and symmetric) nodes according to the expected cost distribution (Figure 6.2), random noise of various amplitudes is added to the cost potential V_{ij^*} according to

$$V_{ij^*} \leftarrow V_{ij^*} + A\xi_i(t), \qquad \forall i \tag{6.13}$$

where $\xi_i(t)$ represents normalized random noise varying between 0 and 1, and A is the amplitude. This noisy measure is employed here as a general (problem independent) and effective way to enhance solution quality. Values for other parameters were chosen as follows: $\beta(0) = 0.95$, $r_\beta = 0.95$, $L_1 = 6$, and $L_2 = 5$, unless specified otherwise.

Figure 6.3 shows the typical feasibility plots in the normalization parameter space, where the annealing rate r varies along the horizontal axis and initial temperature $T(0)$ along the vertical axis. As an illustration, annealing schedules close to the lower left corner of the parameter space correspond to fast convergence to 0-1 solutions, while those close to the upper right corner correspond to slow ones. The shading represents the feasibility measured, with darker shades for higher feasibilities. Noise as described above (with $A = 0.05$) is used in Figure 6.3(b) but not in Figure 6.3(a). It

N-queen constraints, and a local minimum is a solution with at least one constraint violated. For each parameter setting of the network, feasibility is measured in the two dimensional parameter space (r, $T(0)$). Each point in the plane represents a particular annealing scheme for the weight normalization. Another main parameter to be adjusted in our study is the neighborhood size η, which strongly affects the extent of weight updating for each node. Because of this effect and the tendency of the winner to oscillate between two equally attractive (and symmetric) nodes according to the expected cost distribution (Figure 6.2), random noise of various amplitudes is added to the cost potential V_{ij*} according to

$$V_{ij*} \leftarrow V_{ij*} + A\xi_i(t), \qquad \forall i \tag{6.13}$$

where $\xi_i(t)$ represents normalized random noise varying between 0 and 1, and A is the amplitude. This noisy measure is employed here as a general (problem independent) and effective way to enhance solution quality. Values for other parameters were chosen as follows: $\beta(0) = 0.95$, $r_\beta = 0.95$, $L_1 = 6$, and $L_2 = 5$, unless specified otherwise.

Figure 6.3 shows the typical feasibility plots in the normalization parameter space, where the annealing rate r varies along the horizontal axis and initial temperature $T(0)$ along the vertical axis. As an illustration, annealing schedules close to the lower left corner of the parameter space correspond to fast convergence to 0-1 solutions, while those close to the upper right corner correspond to slow ones. The shading represents the feasibility measured, with darker shades for higher feasibilities. Noise as described above (with $A = 0.05$) is used in Figure 6.3(b) but not in Figure 6.3(a). It can be observed that extra dark bands of good feasibility appear in Figure 6.3(b), which are caused by the introduction of noise. Thus the simple measure of adding random noise to the cost potentials leads to a drastically improved feasibility over large regions of the parameter space. This expands the choice of faster annealing schedules associated with good feasibility, which translates into higher efficiencies and robustness (coverage of high feasibility regions in the parameter space).

In order to examine the effect of noise amplitudes on feasibility, we choose a typical parameter pair $(r, T(0)) = (0.38, 0.5)$ which lies around midway on the first curved dark band from the left of Figure 6.3(b), and solve the 5-queen problem with the SONN-WN using different values of A. The result is plotted in Figure 6.4, which shows the dependence of feasibility on noise amplitudes. Without noise ($A = 0$), a low feasibility of 0.06 is obtained; with the presence of noise, feasibility rises sharply with increasing noise amplitudes, and peaks at around $A = 0.075$ with a high feasibility of around 0.8, before it gradually tails off with larger noise amplitudes. While the ways in which weak external noise can enhance solution quality is to be illustrated later in Section 6.3.2 with further experimental results, the detrimental effect associated with large amplitude noise can be readily explained. With the addition of large amplitude noise, the true cost potentials of the nodes are disguised, thereby disrupting the competition-based convergence process toward feasible solutions.

In Figure 6.5, two different $\beta(0)$ values are used, $\beta(0) = 0.5$ for Figure 6.5(a) and $\beta(0) = 0.95$ for Figure 6.5(b). These two values are not decreased according to the SONN algorithm, but fixed throughout each run, i.e. $r_\beta = 1$. Moreover, noise is used for both figures. The results show that annealing β (or the lack of it) does not alter the primary structure of the band patterns. It can also be observed in Figure 6.5(a) that a smaller β decreases feasibility, which causes the bands to be lighter and thinner. Also, the increased feasibility due to a constant $\beta = 0.95$ in Figure 6.5(b) causes the bands to appear darker and wider, especially when compared to Figure 6.3(b). The fact that a larger value of β favors higher feasibility may be explained as follows: since β is the updating step size of W_{i0,j^*}, small values of β may not be sufficient to cause a significant update of W_{i0,j^*} *after the normalization process*, where $W_{i0,j}$'s of smaller values are suppressed (especially when T is small).

Since the neighborhood size η affects the quality of topographical mappings in Kohonen's SOFM [63], we investigate its effect on our optimization problem. In Figure 6.6, the neighborhood size η is kept constant at $\eta = 4$, in contrast to the linearly decreasing η in the typical case. β is fixed to 0.95 as a control for simple comparisons. It can be seen that in Figure 6.6(a) where noise is absent, most of the high feasibility bands disappear; while in Figure 6.6(b) where noise is present, an identical result to Figure 6.5(b) is obtained. Since the case of $\eta = 4$ (fixed) retains most of the symmetry of the cost potential distribution (Figure 6.2), this case illustrates dramatically the beneficial effect of noise toward solution quality.

By the same argument, other means of symmetry breaking toward the cost potential distribution could improve feasibilities as well, as long as the cost potentials are not too disrupted. One way to achieve this is to use a neighborhood size such that the updating of nodes retain as little as possible the symmetry property of the cost distribution. In Figure 6.7, a fixed $\eta = 0$ is used (i.e. only the weights of the winning node are updated), with no noise in Figure 6.7(a) and noise applied in Figure 6.7(b). It can be observed that both feasibility and robustness are good, and the improvement due to noise is minimal. Since only the winning node's weights are updated when $\eta = 0$, the other node having a symmetric cost potential cannot be updated at the same time, thus breaking the unwanted symmetry that leads to oscillation as shown later in the next section. The use of any constant $\eta < (N - 1)$ leads to similar results (see Figure 6.8 with a constant $\eta = 3$). In fact, the result of using a linearly decreasing η as shown in Figure 6.3 and 6.5 reflects the same phenomenon. As η is decreased in the convergence process, updating involves fewer and fewer nodes, so less cost symmetry is retained. This explains why Figure 6.3(a) (η decreasing from 4 to 0) lies between Figure 6.6(a) ($\eta = 4$) and 6.7(a) ($\eta = 0$) in terms of robustness without the use of noise. As to be discussed in Section 6.3.3, some annealing schedules are less affected by this symmetry issue than others, as revealed by some parameter regions invariant to all the η manipulations as shown in Figure 6.3, 6.6 and 6.7.

From the results given here, it is thus illustrated that a decreasing neighborhood size is not necessarily an effective method to obtain good feasibility and robustness when using the SONN-WN for our problem, and more emphasis should be directed to the choice of certain neighborhood sizes to avoid the oscillation problem. Similar outcomes are expected when the SONN-WN is applied to other COP's with some kinds of inherent cost potential symmetry.

6.3.2 Controlling oscillations: A dynamical study

In order to investigate how the added noise actually affects the convergence of the network and the resulting solution quality, we plot in Figure 6.9 the node potentials V_{ij} for $i = 1\ldots5$ against each input presentation. Since the weights are updated following each input presentation, this kind of plot can be viewed as a time evolution of the system dynamics. As the competitive learning is based on the ranking of cost potentials, plots of V_{ij}'s are most useful for monitoring the nodes' competitiveness upon each input presentation. The five colored lines represent V_{ij}'s of the five nodes, for $i = 1\ldots5$. The black line represents $T(t)$, and is used to visualize the annealing schedule. In Figure 6.9, the annealing schedule $(r, T(0)) = (0.26, 0.7)$ and a fixed $\eta = 4$ are used. It should be noted that the only difference in configuration between Figure 6.9(a) and 6.9(b) is the absence of noise in Figure 6.9(a), while noise of $A = 0.05$ is used in Figure 6.9(b). Figure 6.9(a) corresponds to a low feasibility point in Figure 6.6(a), while Figure 6.9(b) corresponds to a high feasibility point in Figure 6.6(b).

The main observation from Figure 6.9(a) is that after two drops in temperature T, nodes form pairs of the same potentials whenever possible, e.g. $V_{2,j} = V_{4,j}$ and $V_{1,j} = V_{5,j}$. Such a specific pairing of V's is caused by the symmetric distribution of expected cost discussed previously (see Figure 6.2). Since the winning node is the one having the lowest V for each input presentation, it can be observed that no clear winner results for most presentations. The situation remains unchanged for the remaining iterations (750 input presentations in total). Such a pairing phenomenon translates into having more than one queen for each column on the chessboard, so the solution obtained in Figure 6.9(a) is a local minimum. When noise is present as in Figure 6.9(b), a "fairer" competition results, where each node takes turn to attain the lowest V as each input is presented. This can be observed as five distinct colors being clearly visible in the second half of the picture. This means for each column on the chessboard, there is only one queen. The actual solution obtained (not shown) that corresponds to Figure 6.9(b) is indeed a global minimum, which implies that diagonal constraints are also satisfied. The corresponding weight values $W_{1,j}$ for $j = 1\ldots5$ of the two scenarios are plotted as Figure 6.10(a) (no noise) and Figure 6.10(b) (with noise). $W_{1,j}$'s are the weights of node 1, whose behaviors described here are found to be typical among other nodes. A phase transition-like behaviour can be observed in both Figure 6.10(a) and (b) when T is first decreased at presentation 30, where the values of W's are sharply redistributed towards 0 and 1 due to the normalization function (6.6) (a manifestation of this can also be observed in Figure 6.9). This phenomenon is caused by the bifurcation dynamics of the weights to be explained by the equilibrium model in the next chapter. Comparing Figure 6.10(a) and (b), the weight values differ only slightly for the first half of the pictures. After the second T drop however, oscillatory behaviors in $W_{1,1}$ and $W_{1,3}$ are clearly visible in Figure 6.10(a), but not in Figure 6.10(b) where noise is present. These oscillations in the weights are the direct results of an undesirable oscillatory weight updating cycle caused by the pair-wise competitions observed in Figure 6.9(a). On the other hand, Figure 6.10(b) shows clearly that with the addition of weak noise, convergence becomes well behaved or oscillation free (which may sound counter-intuitive if without the evidence and reasoning provided herein). Since the network dynamics in Figure 6.9(a) and 6.10(a) are associated with the point (0.26, 0.7) in Figure 6.6(a) (low feasibility), and those in Figure 6.9(b) and 6.10(b) are associated with the same point in Figure 6.6(b) (high feasibility), it is demonstrated that small amplitude noise added to the cost potentials acts as a stochastic device to break up unwanted dynamics, resulting in an improved competition among nodes that leads to higher solution quality. Moreover, the obtained high feasibility means that the likelihood of obtaining a global minimum solution is very high for any particular set of initial network states. This is a much desirable property for practical deployments of the SONN-WN when time or other resources limit the number of runs with different initial states.

Upon close inspection, it can be observed that differences between Figure 6.10(a) and (b) arise at around the 42nd presentation, where T is around 0.2. This suggests that with a certain noise amplitude, the SONN-WN dynamics are noise sensitive *only* at a specific temperature range. As shown in Figure 6.10, this particular range of T also causes the weight values to have a stronger tendency toward 0-1 values because of the normalization function (6.6). This means that even though the beneficial effect of noise is mainly concerned with the competition of nodes, such an improvement only occurs when the normalization temperature is *favorable*. A detailed investigation is to be given in Chapter 7 for the meaning of this *favorable temperature range* in terms of nonlinear system dynamics (NSD). However, here we can say that the normalization function encourages a *moderately strong* tendency of 0-1 behavior in this temperature range, which is required for the convergence of W_{ij}'s toward 0 or 1 decision values. The important point is that if the tendency for 0-1 convergence is either too weak (resulting in W_{ij} values around $1/N$ after each normalization) or too strong (W_{ij} values too close to 0 or 1 after each normalization), the effect of noise would be washed out by the weight normalization effect. Because of this noise sensitive 0-1 convergence property, we call this temperature range the *flexible decision temperature interval*, or FDT interval for short.

6.3.3 From the structure of feasibility bands to effective annealing

In this section, we illustrate how to choose an effective annealing schedule for the weight normalization by studying the structure of feasibility bands observed in Section 6.3.1. It should be noted that the existence of the FDT interval described above is crucial in understanding the band patterns of the feasibility plots. For this reason, the actual FDT interval should be determined first. Although we are presenting a theoretical model in Chapter 7 in which the FDT interval is deduced from NSD, here we seek the FDT interval in the reverse direction of the cause-effect relationship between dynamics and feasibility.

We begin by inspecting the feasibility plots. From Figure 6.6(a), a dark horizontal band can be observed in the lower part of the plot. This band represents annealing schedules having a low starting temperature $T(0)$ roughly in the range of 0.12 ~ 0.22. It can be seen from Figure 6.6(a) and 6.6(b) that the feasibility improvements on this band due to noise is not as dramatic as some other regions. To look into the network dynamics typical of this horizontal band, we present in Figure 6.11 the time evolution of V_i's, with the noise free configuration of Figure 6.6(a). Figure 6.11(a) corresponds to the annealing schedule of $(r, T(0)) = (0.26, 0.18)$, representing a fast annealing rate with low initial temperature; Figure 6.11(b) corresponds to $(r, T(0)) = (1, 0.18)$, representing *no* annealing with the same $T(0)$. Since the initial network states are equivalent, both figures have identical behaviors for the first 30 input presentations. After that, T is lowered in Figure 6.11(a), and a uniform, converged state results because of the very low temperature in the normalization function (6.6). If T is not lowered at all, i.e. no annealing, the long-term evolution would be different, and such a case is shown in Figure 6.11(b). It can be observed from Figure 6.11(b) that a kind of unsettled behavior remains, with ever fluctuating V_i values, but nonetheless sustaining an oscillation free competition (judged by the absence of paired winners). Although both kinds of annealing schedules yield high feasibility as shown in Figure 6.6(a), the no annealing scenario of Figure 6.11(b) corresponds to an even higher feasibility. This means that for 100 runs of the algorithm with different initial network states, oscillations occur more often for the annealed case than the one with constant T. Thus the particular temperature $T = 0.18$ is a preferred temperature for yielding very high feasibilities, and should be maintained as long as possible *for the highest feasibility*. But for practical applications, 0-1 values of W_{ij}'s converged within a limited number of iterations are required, which is not satisfied by the slow convergence of the no annealing case. To satisfy this need, points of various shades on the horizontal dark band (Figure 6.6) serve as a guide for choosing the suitable annealing rate r according to the user's time and feasibility requirements.

It should be noted that the use of $T = 0.18$ for effective optimization described above is for illustration purpose only. In fact all temperatures roughly in the range $0.12 \sim 0.22$ would yield similar results. Since the horizontal dark band in

Figure 6.6(a) and 6.6(b) corresponds to this temperature range as described in the last paragraph, we conclude that the temperature range of 0.12 ~ 0.22 is the FDT interval for the network configuration of Figure 6.6 ($\beta = 0.95$, $\eta = 4$ (fixed)). For a different updating step size β and neighborhood size η, the FDT interval would be different. This is expected because of the central roles played by these two parameters in the competitive self-organization process. As the horizontal dark band is a visible representation of the FDT interval in the (r, $T(0)$) space, any change of the FDT interval due to changes in β and η would result in a visible change in the horizontal dark band as well. This is confirmed by the thinning of the band with a reduced β (discussed in Section 6.3.1, also see Figure 6.5), as well as an upward widening of the band with a reduced $\eta = 0$ (fixed) (see Figure 6.7).

Now that the horizontal dark band has been associated with the FDT interval, other dark bands of feasibility in the (r, $T(0)$) space can be readily explained. The key point for the existence of dark bands is this: a (r, $T(0)$) point becomes dark whenever $T(t)$ falls into the FDT interval anytime within the prescribed annealing schedule. It would be helpful to understand the same fact by looking at the white regions (very low feasibility). For example, let us look at Figure 6.3(b) in which the bands are clearly visible and well separated. All the white points in the figure represent annealing schedules whose stepwise decreasing $T(t)$ skips the FDT interval. If $T(t)$ never falls into the FDT interval in a particular run, 0-1 convergence for the weight values (whose converged state represents the solution matrix) is described here as either *under-converged* or *over-converged* with respect to an effective solution process. Weights are under-converged when the W_{ij} values lie around $1/N$ instead of 0 or 1, which happens when $T(t)$ is too high during convergence (above FDT interval); Weights are over-converged when the W_{ij} values approach 0 or 1 values too readily, which happens when $T(t)$ is too low during convergence (below FDT interval). Figure 6.12(a) is a typical example of under-convergence with $(r, T(0)) = (1, 0.26)$. Since the weight values never converge to either 0 or 1 throughout the solution process, we call the parameter regime with this behavior an *under-decision* region. On the other hand, Figure 6.12(b) is a typical example of over-convergence with $(r, T(0)) = (0.26, 0.06)$, where the weights lock into 0-1 values too readily before undergoing a proper solution process. We call such a parameter regime the *over-decision* region. There is also a situation where an under-converged phase is immediately followed by an over-converged phase. This happens when the annealing schedule starts with a $T(0)$ above the FDT interval, then followed by $T_m = r^m T(0)$ that is lower than the FDT interval after the m^{th} temperature drop, for $m = 1, 2, 3, \ldots$, etc. Thus the FDT interval is skipped. This case corresponds to white regions *between* the dark bands.

Using a similar reasoning based around the FDT interval, we can explain the curved dark bands that appear in some of the feasibility plots. Let us take Figure 6.3(b) as a typical example. The first curved dark band (or the left most) in the $(r, T(0))$ space corresponds to annealing schedules that fall into the FDT interval after the first T drop. Similarly, the m^{th} curved dark band (counting from left to right of the plot) corresponds to annealing schedules having a $T_m = r^m T(0)$ that falls into the FDT interval, for $m = 1, 2, 3\ldots$etc. In fact, the horizontal dark band belongs to the case of

$m = 0$. Again, this confirms our theory that the FDT interval is essential for good feasibility.

Almost all the major phenomena concerning the band structure in the feasibility plots have now been accounted for, with one question remaining: by comparing Figure 6.6(a) and (b), why is noise necessary for the curved bands to achieve high feasibilities, but unnecessary for the horizontal band? The answer is that the network dynamics associated with annealing schedules on the horizontal band start with T within the FDT interval, which means every run can start under the favorable condition of noise sensitive 0-1 convergence that we have discussed in Section 6.3.2. But for the curved bands, the starting temperature $T(0)$ is always too high (above FDT interval). Since $T(0)$ persists for the first 30 input presentations, the early dynamics suffer from under-convergence as discussed previously. In fact, this is the case for Figure 6.10 whose annealing schedule lies on the first dark band. The weight matrix after the first 30 iterations for the run in Figure 6.10(b) is given by

$$\mathbf{W} = \begin{pmatrix} 0.105 & 0.082 & 0.083 & 0.090 & 0.640 \\ 0.022 & 0.020 & 0.020 & 0.020 & 0.918 \\ 0.020 & 0.014 & 0.014 & 0.014 & 0.939 \\ 0.012 & 0.011 & 0.011 & 0.011 & 0.954 \\ 0.102 & 0.081 & 0.083 & 0.090 & 0.644 \end{pmatrix}$$

which is severely imbalanced, with the fifth column having almost all the updating due to the expected cost distribution (Figure 6.2). This **W** is clearly not in a proper "state-of-mind" leading to a feasible 5-queen solution. This is the reason why even when T is lowered to within the FDT interval after the first temperature drop, extra help in the form of external noise is needed to revert **W** into a potentially feasible solution matrix.

From the above results and observations, we conclude for this section that the most effective annealing schedules are found on the horizontal dark band that appears on the feasibility plots in $(r, T(0))$ space, i.e. annealing schemes having $T(0)$ within the FDT interval. Furthermore, the slower the annealing rate is *within this category*, the higher the feasibilities obtained. Lastly, Figure 6.13 is presented as a diagrammatic summary of all the major findings concerning the structure of feasibility bands in terms of characteristic convergence dynamics.

6.4 Concluding remarks

In this chapter, we have experimentally investigated the optimization performance of the SONN-WN, with a primary focus on its weight normalization effects. Fundamentally the weight normalization process is used for constraint satisfaction, and in our N-queen implementation, it is used to enforce both the row constraint and the 0-1 integrality constraint. In the dynamically rich self-organizing environment of the SONN-WN, such a constraint enforcement procedure is transformed into a decision-making process facilitated by convergence behaviors of a complex system. Based on this dynamical scenario of decision-making, we have investigated the role of weight normalization in the convergence dynamics of the network, and its subsequent effect on feasibilities.

By plotting feasibilities in the normalization parameter space, we have revealed that only certain annealing schedules can result in high feasibilities, and that a slower annealing rate does not correspond to better solution quality or higher feasibilities in general, as in the case of simulated annealing [1]. With the addition of random noise to the cost potential, we have been able to improve feasibilities and robustness dramatically, which are revealed in the $(r, T(0))$ space in the form of dark bands. We have also demonstrated the use of small neighborhood sizes ($\eta < N - 1$) for improving feasibilities and robustness. These results also serve to emphasize the importance of symmetry issues in the cost function to be minimized.

To explain the noise sensitive improvements on feasibilities depending on the annealing schedules, we have proposed a theory on controlling oscillatory convergence behaviors via a special condition of the weight normalization process: within a special temperature range called the flexible decision temperature (FDT) interval, the weight normalization process becomes sensitive to noise of certain amplitudes, resulting in a particular mode of 0-1 convergence that can alleviate the cost symmetry problem that causes unwanted oscillations. Various plots on the dynamics of the cost potentials and weights have provided a detailed picture of the convergence dynamics of the SONN-WN, and have been the indispensable tools in our investigation.

By applying our proposed theory on the effective optimization within the FDT interval, we have accounted for the structure of feasibility band patterns in terms of convergence dynamics. Different annealing schedules lead to different phases of convergence depending on the sequence of temperature reductions. An annealing schedule with good feasibility corresponds to a sequence of phase transitions that include the FDT as the normalization temperature. On the other hand, FDT is never attained in annealing schedules yielding poor feasibilities. Based on this anatomy of the feasibility plots, we conclude that an effective annealing schedule for good solution quality and feasibility should start with an initial temperature within the FDT interval, together with a slow annealing rate determined by the user's feasibility requirement and time limitations.

The purpose of this chapter has been to examine the effect of annealing schedules, noise and other parameters on the optimization performance of the SONN-WN. We have demonstrated and explained some interesting phenomena for the 5-queen problem. Similar phenomena exist for larger problem sizes, but a comprehensive examination of the SONN-WN's performance on larger problems is beyond the purpose of this research. Lacking in this chapter though is the origin of the FDT interval in relation to the normalization function (6.6), which is the focus of investigation for Chapter 7.

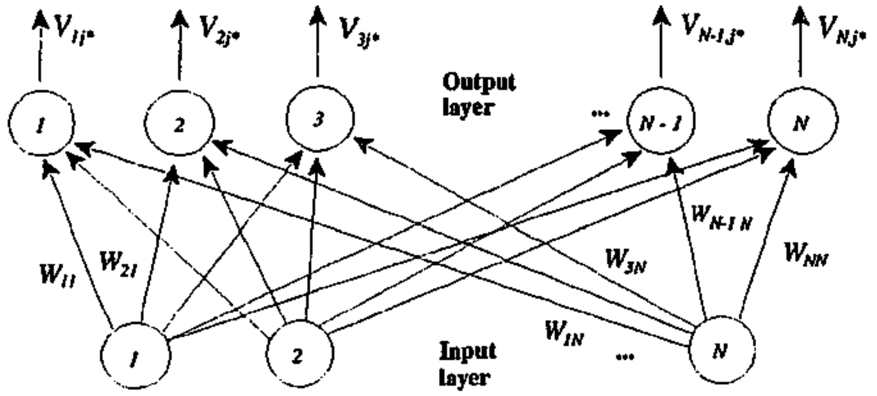


Figure 6.1: Architecture of the SONN-WN.

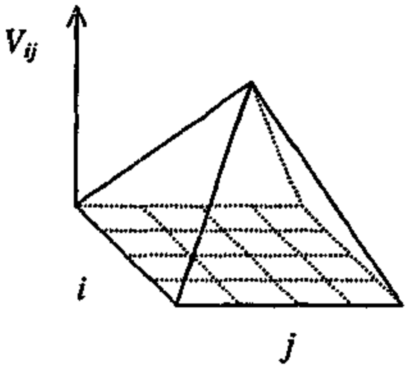


Figure 6.2: Expected distribution of cost V_{ij} on the chessboard.

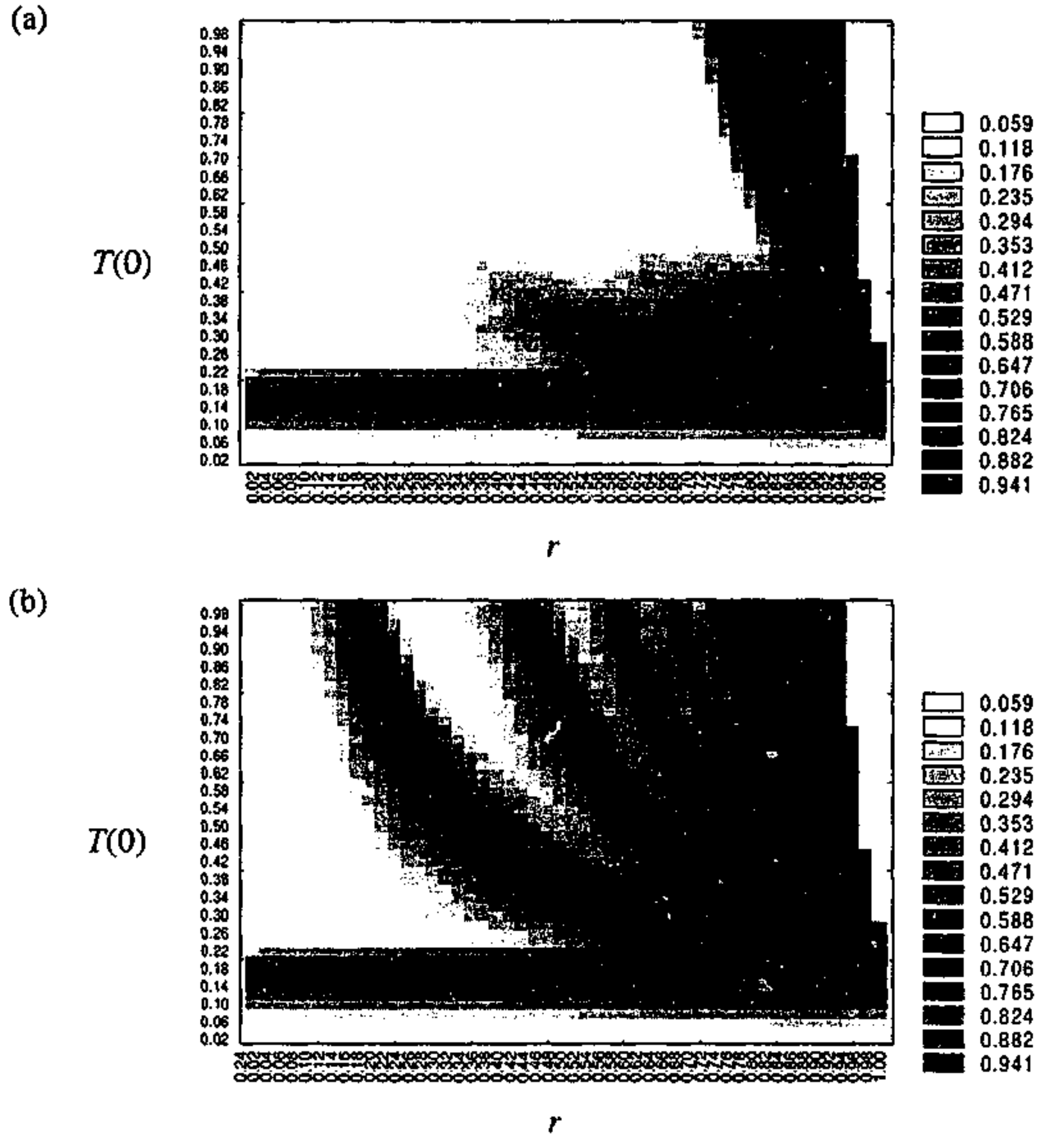


Figure 6.3: Feasibility with $\beta(0) = 0.95$, $r_\beta = 0.95$, and (a) $A = 0$; (b) $A = 0.05$.

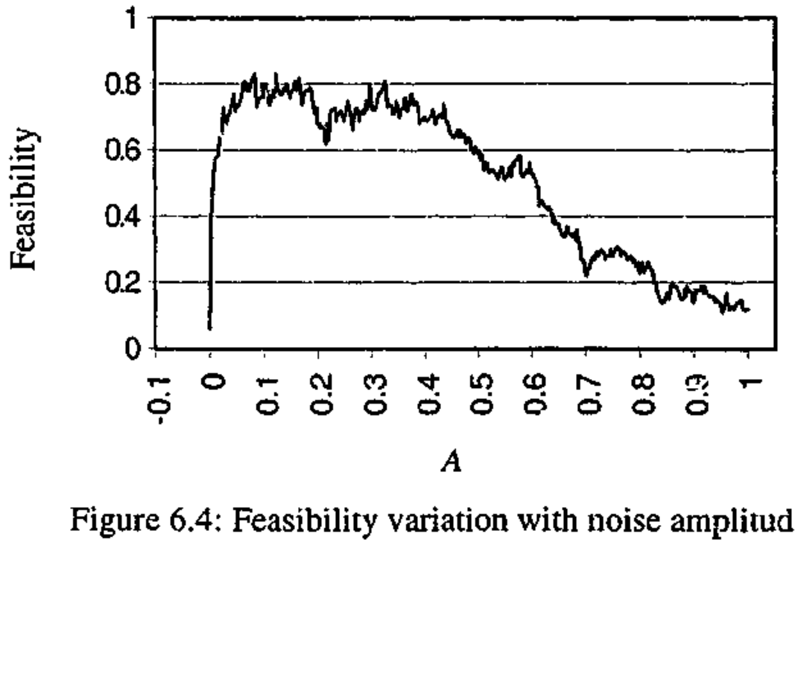


Figure 6.4: Feasibility variation with noise amplitude A.

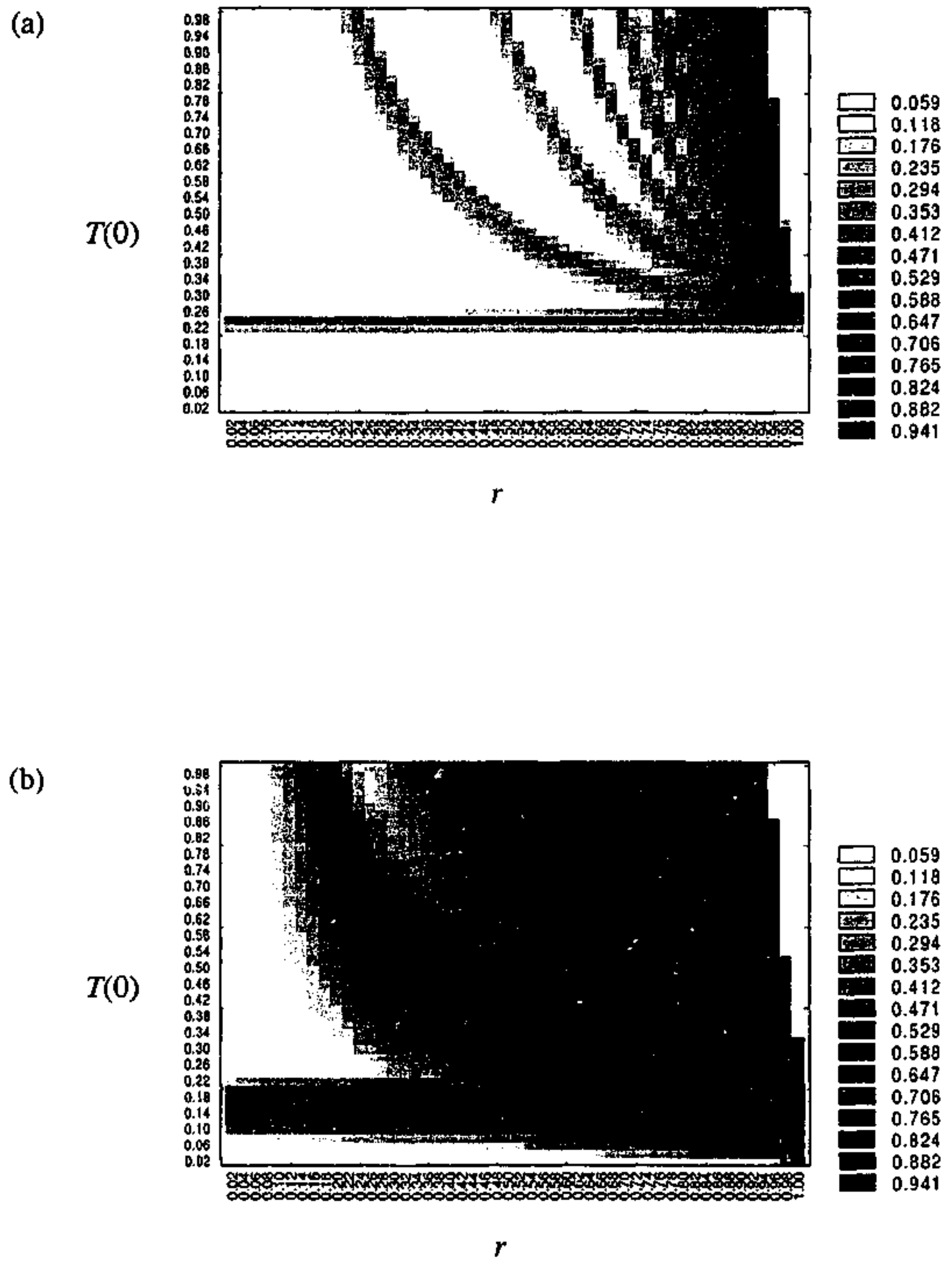


Figure 6.5: Feasibility with $r_\beta = 1$, $A = 0.05$, and (a) $\beta(0) = 0.5$; (b) $\beta(0) = 0.95$.

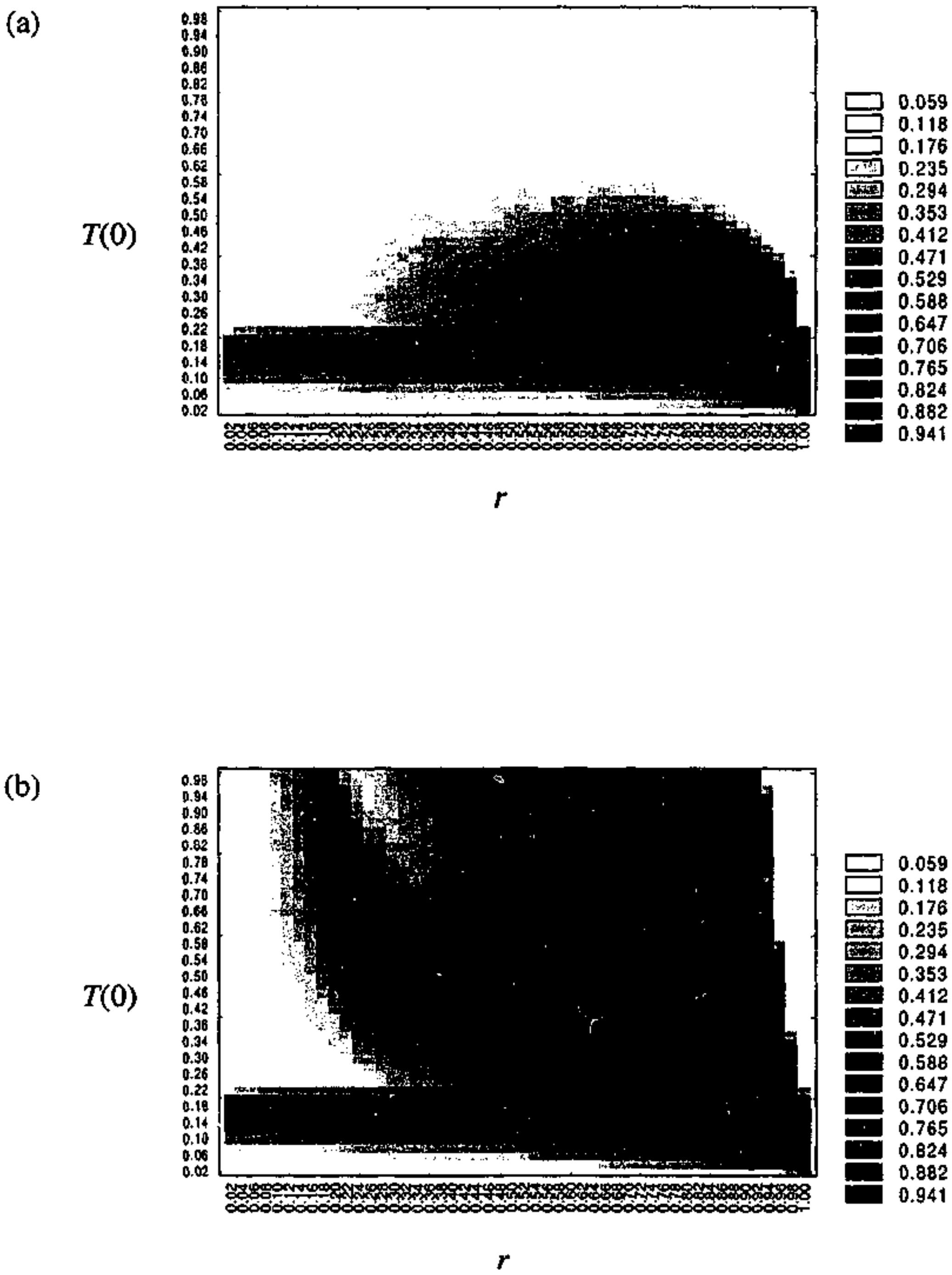


Figure 6.6: Feasibility with $\beta(0) = 0.95$, $r_\beta = 1$, $\eta = 4$ (fixed), and (a) $A = 0$; (b) $A = 0.05$.

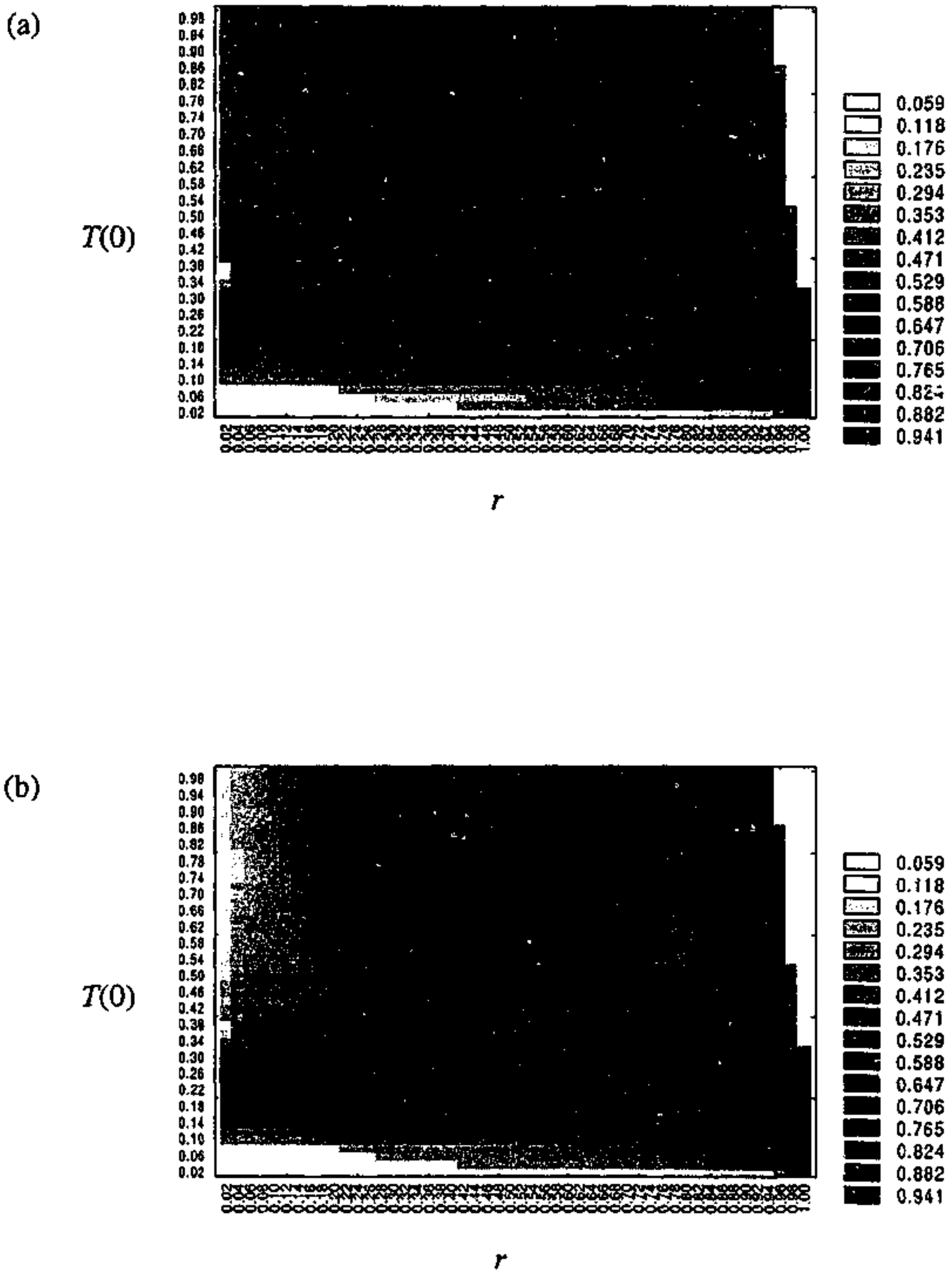


Figure 6.7: Feasibility with $\beta(0) = 0$[illegible], $r_\beta = 1$, $\eta = 0$ (fixed), and (a) $A = 0$; (b) $A = 0.05$.

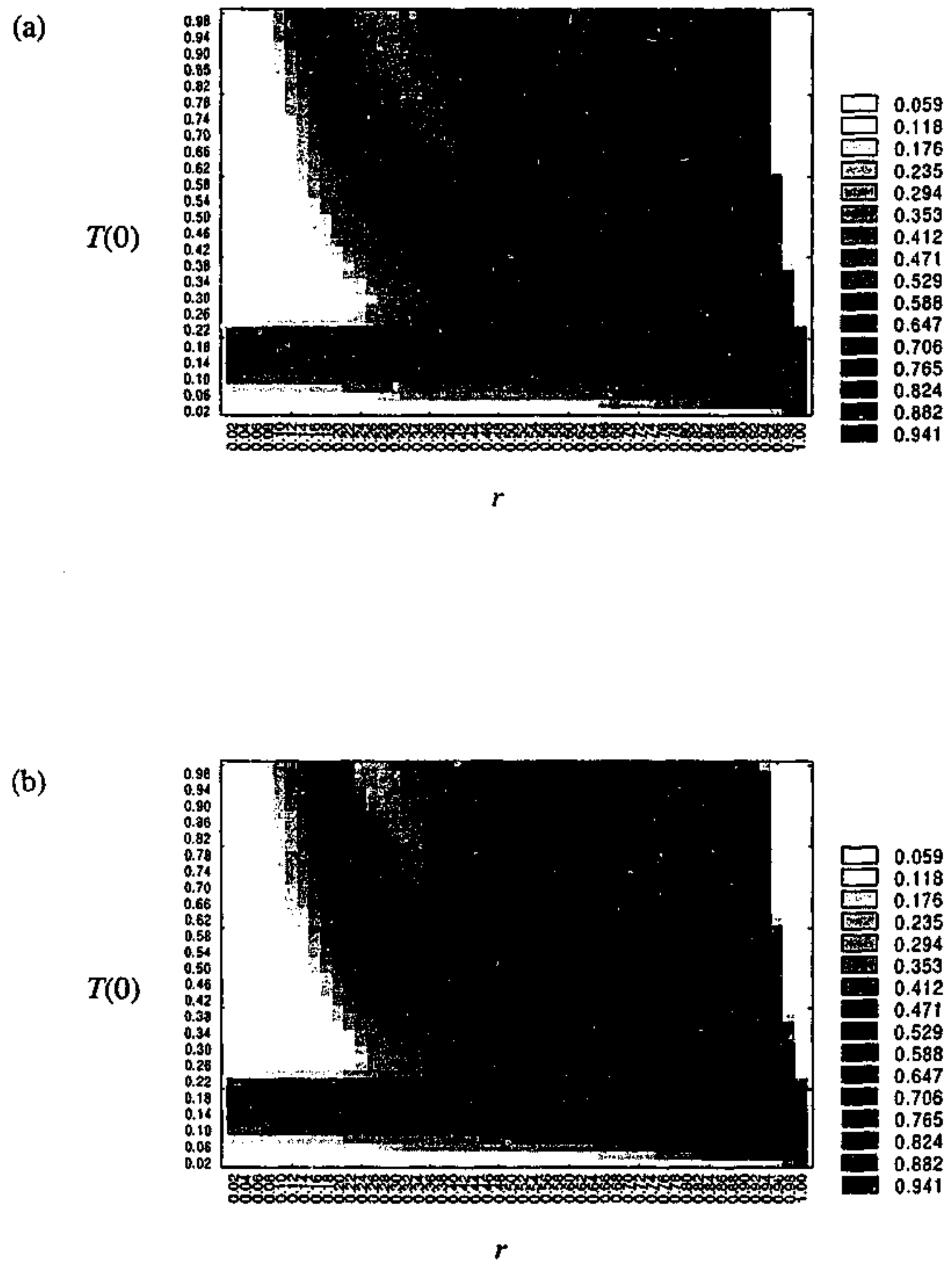


Figure 6.8: Feasibility with $\beta(0) = 0.95$, $r_\beta = 1$, $\eta = 3$ (fixed), and (a) $A = 0$; (b) $A = 0.05$.

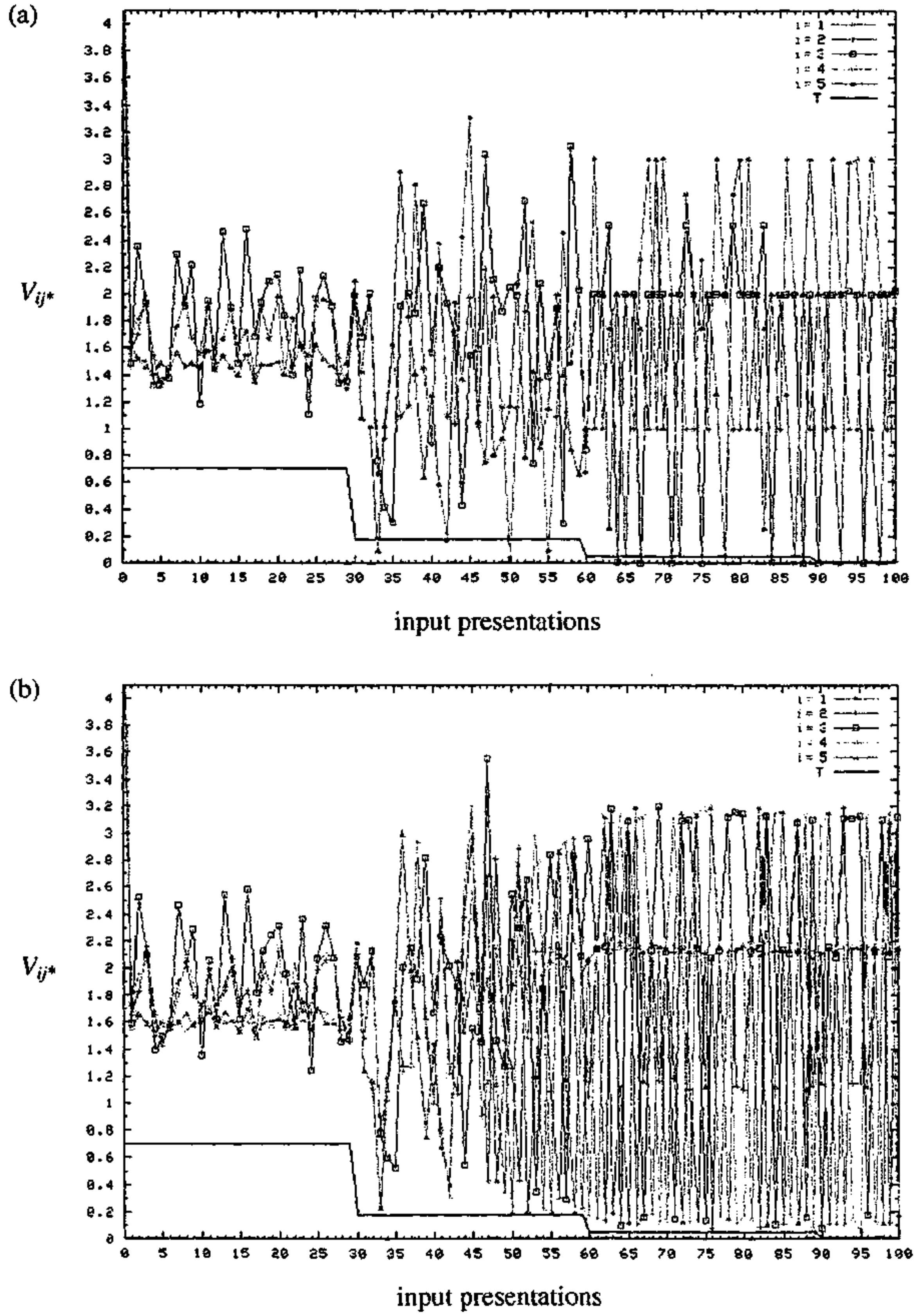


Figure 6.9: Time evolution of cost potentials V_{ij^*} for i = 1...5, with $T(0)$ = 0.7, r = 0.26, $\beta(0)$ = 0.95, r_β = 1, η = 4 (fixed), and (a) A = 0; (b) A = 0.05.

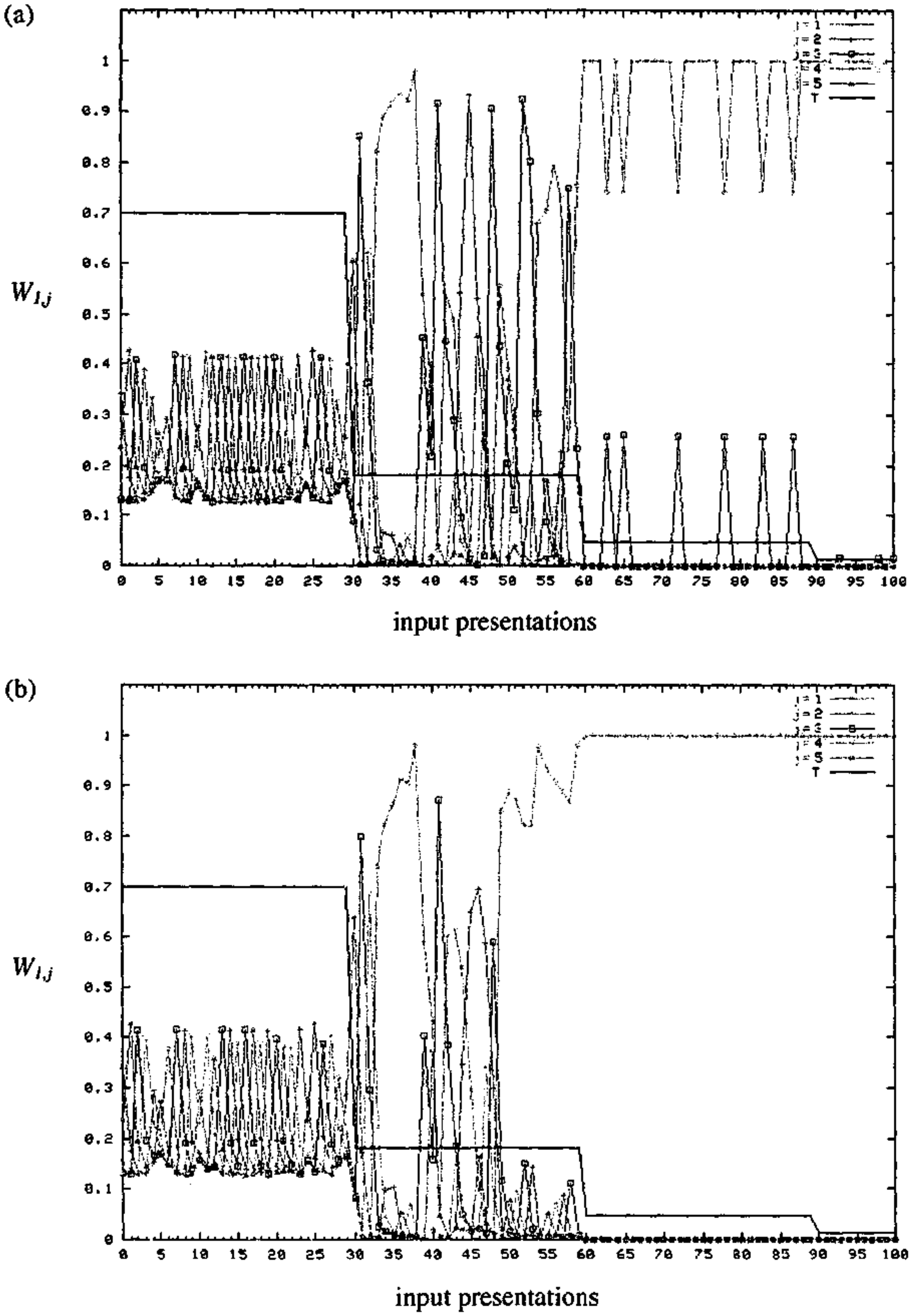


Figure 6.10: Time evolution of weights $W_{1,j}$ for $j = 1 \ldots 5$, with $T(0) = 0.7$, $r = 0.26$, $\beta(0) = 0.95$, $r_\beta = 1$, $\eta = 4$ (fixed), and (a) $A = 0$; (b) $A = 0.05$.

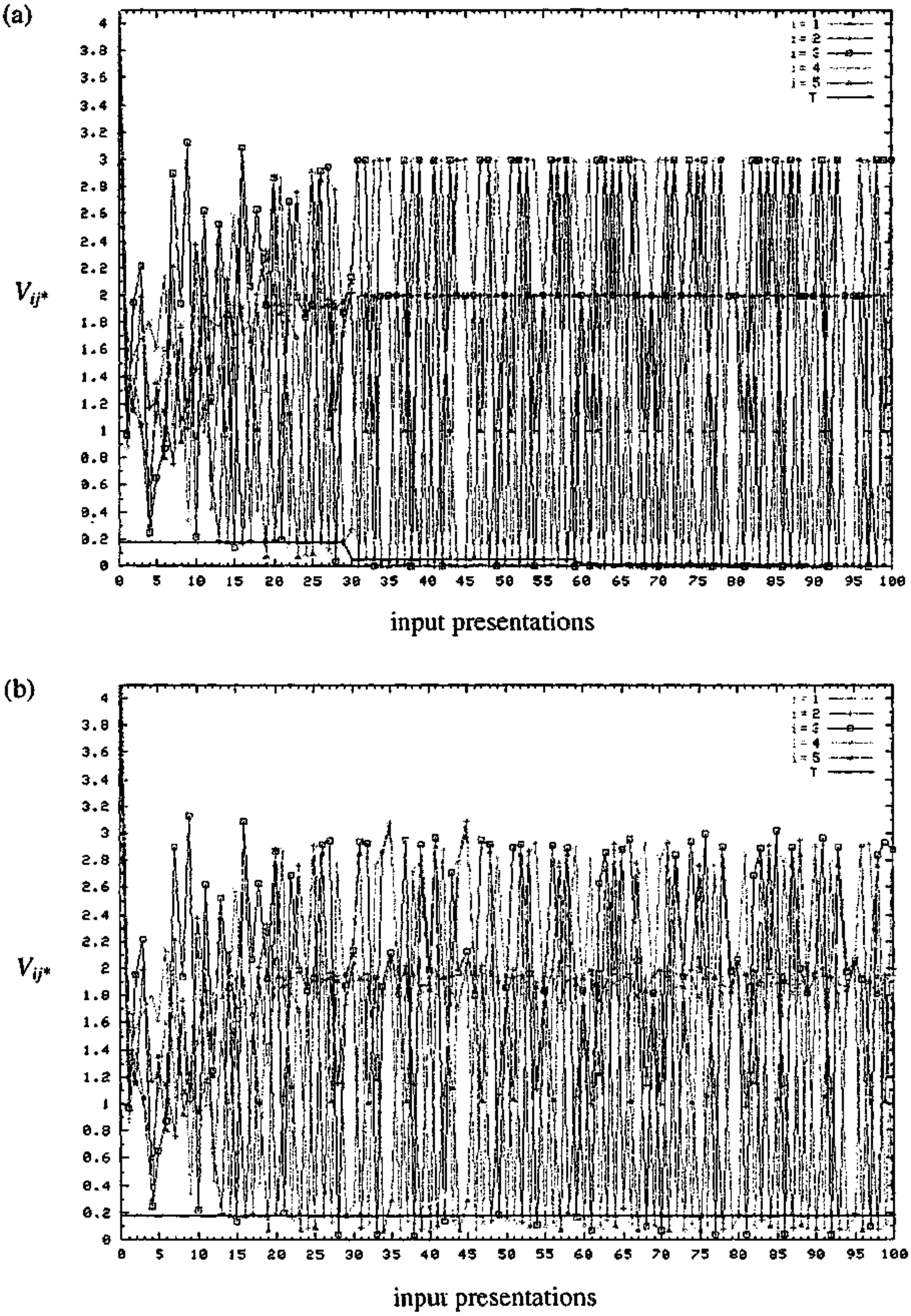


Figure 6.11: Time evolution of cost potentials V_{ij^*} for i = 1...5, with $T(0)$ = 0.18, $\beta(0)$ = 0.95, r_β = 1, η = 4 (fixed), A = 0, and (a) r = 0.26; (b) r = 1.

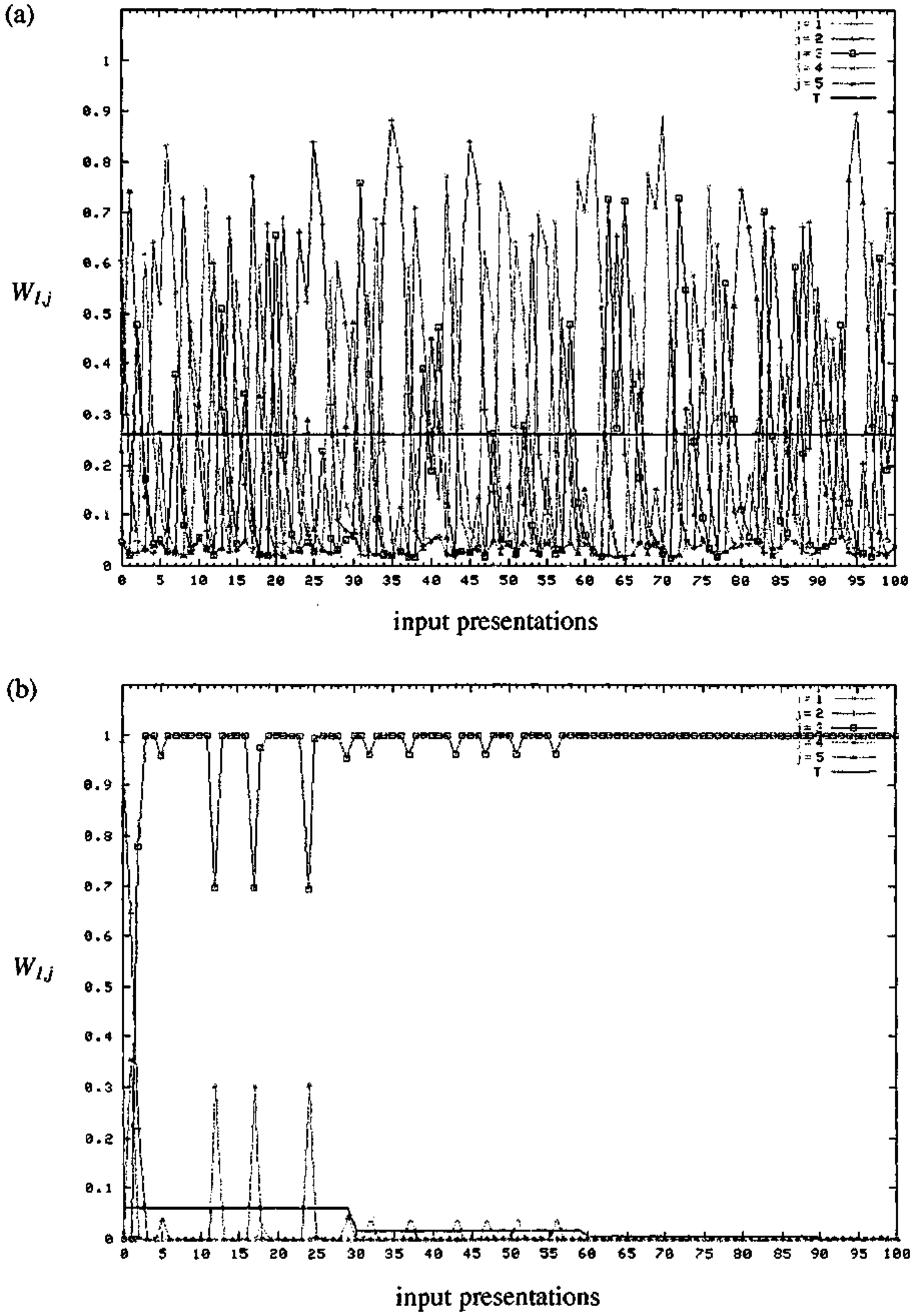
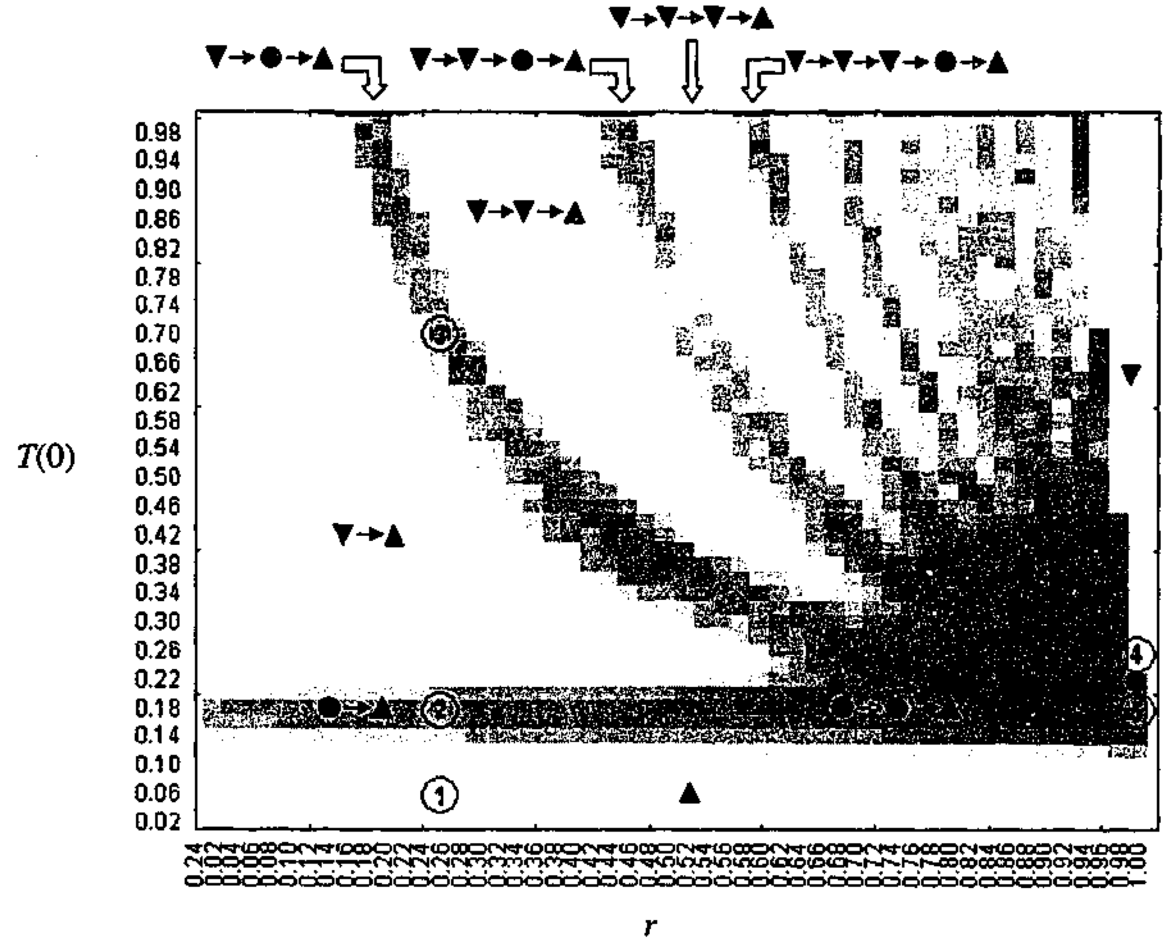


Figure 6.12: Time evolution of weights $W_{1,j}$ for $j = 1 \ldots 5$, with $\beta(0) = 0.95$, $r_\beta = 1$, $\eta = 4$ (fixed), $A = 0$, and (a) $(r, T(0)) = (1, 0.26)$; (b) $(r, T(0)) = (0.26, 0.06)$.



Symbols for characteristic dynamics:	▼ : under-convergence phase ● : effective optimization phase ▲ : over-convergence phase ➔ : annealing triggered phase transition (via *T* reduction)
Pointers to figures shown elsewhere:	① : Figure 6.12(b) ② : Figure 6.11(a) ③ : Figure 6.9, 6.10 ④ : Figure 6.12(a) ⑤ : Figure 6.11(b)

Figure 6.13: A diagrammatic representation of characteristic convergence dynamics for each normalization parameter regime. A typical pattern of high feasibility bands is used for reference. Time sequential phases of convergence dynamics for each parameter regime are represented by strings of symbols. Circled numbers are pointers referring to correspcnding figures detailing specific *V* or *W* dynamics.

Chapter 7

Understanding the SONN-WN Dynamics through an Equilibrium Model

7.1 Introduction

We have seen in the last chapter the influence of the SONN-WN's weight normalization temperature on its convergence dynamics, and the importance of choosing the right annealing schedules for effective optimization. In this chapter, we present a dynamical systems view of the SONN-WN with a theoretical approach.

It has been shown in the last chapter that only certain annealing schedules of the weight normalization process yield good quality solutions with a high feasibility, and that the choice exclusively depends on whether the annealing temperature at any point in time falls into the flexible decision temperature (FDT) interval. According to our computational results, we have seen that if the normalization temperature is too high (above FDT interval), under-convergence results, leading to under-decision; if the temperature is too low (below FDT interval), over-convergence occurs, resulting in over-decision. These two situations are both undesirable for an effective convergence process. The most favorable condition occurs when the temperature is *just right* (within the FDT interval), which gives rise to a convergence phase with moderately strong 0-1 tendency that is amenable to noise enhancements. Up till now we have not shown much about the origin and *modus operandi* of the FDT interval, except that it varies with different updating step sizes β and neighborhood sizes η. Thus it is our main objective in this chapter to investigate how the FDT interval arises, and through what mechanisms it achieves such an *attractive* convergence behavior.

Our investigation starts from the derivation of a theoretical model called the equilibrium model, which captures the essential aspects of the SONN-WN's weight updating and normalization dynamics. The use of a model allows us to focus on the elements crucial to the SONN-WN's long-term convergence dynamics without being overly complicated. The derivation of the equilibrium model is given in Section 7.2.

Because of the nonlinear nature of the equilibrium model, a numerical investigation is carried out to solve for the corresponding equilibrium states. The dynamics of these equilibrium states represent the convergence behavior resulting from the mutual interactions among the weights, and we reveal their sensitive dependence on the normalization temperature, thereby giving a detailed account on the FDT interval from a dynamical perspective. We also reveal a rich collection of nonlinear phenomena from the equilibrium model, which are previously unknown to be connected to the SONN-WN. This numerical investigation is presented in Section 7.3.

A discussion combining the theoretical outcomes of this chapter with the optimization results from the last chapter is presented in Section 7.4, followed by the concluding remarks in Section 7.5.

7.2 Derivation of the equilibrium model

In this section, we construct a theoretical model to approximate the weight updating and normalization dynamics of the SONN-WN. For simplicity, the model only describes the weight dynamics of the winning node, and we seek the equilibrium states of its weights under repeated updating and normalization. Note that the process of nodes competing to be the winner, together with the associated effects of non-zero neighborhood sizes (η), are not included in this model. The advantage of making these assumptions is that complex spatio-temporal *transient* dynamics occurring in the full network, especially after each decrease in T, β, and η, can now be excluded from our study. This allows us to focus on the SONN-WN's characteristic long-term dynamics due to weight updating and normalization.

Figure 7.1 is a schematic diagram showing how the weights of a given neuron are updated and normalized as a two-stage process. For convenience, w_j denotes the weight in this analysis, with $j = 1 \ldots N$. In Figure 7.1, w_1 is updated at stage $t + ½$, and is "normalized" using an approximation of (6.6) at stage $t+1$. To express mathematically the first stage using (6.7) and (6.9),

$$w_{j^*}(t+\tfrac{1}{2}) = w_{j^*}(t) + \beta\left(1 - w_{j^*}(t)\right) \tag{7.1}$$

with other w_j's being unchanged from t to $t + ½$. For the second stage,

$$w_j(t+1) = \frac{\exp\left(-\left(1 - w_j(t+\tfrac{1}{2})\right) \big/ T \right)}{\sum_{k=1}^{N} \exp\left(-\left(1 - w_k(t)\right) \big/ T \right)}, \qquad j = 1 \ldots N \tag{7.2}$$

which can be simplified to

$$w_j(t+1) = \frac{\exp\left(w_j(t+\tfrac{1}{2}) \big/ T \right)}{\sum_{k=1}^{N} \exp\left(w_k(t) \big/ T \right)}, \qquad j = 1 \ldots N. \tag{7.3}$$

The difference between (7.2) and (7.3) is that the former is more suitable for computational purpose to prevent overflow problems, and the latter is a theoretically equivalent version to be used in this analysis for its algebraic simplicity. Note that both (7.2) and (7.3) are approximations to the exact normalization procedure in (6.6), but the deviation tends to zero as $\beta \to 0$. This is because when β vanishes, $w_k(t)$ in the denominator of (7.2) and (7.3) would be equal to $w_k(t + ½)$ by using (7.1), then (7.2) and (7.3) would become the exact normalization in (6.6). The two-stage process described by (7.1) - (7.3) is repeated for $j^* = 1 \ldots N$ (an epoch), with j^* chosen in the order: 1, 2, 3,..., N, until all the weights at stage $t + 1$ are "normalized". It should be noticed that (7.1) is used whenever ◆ changes into ❖, and (7.3) is used whenever ❖

changes into ●. For asynchronous updating, $w_k(t+1)$ is used whenever available in the denominator of (7.3), while for synchronous updating, $w_k(t)$ is used for all k's.

Although this is a one-neuron model, the fact that the neuron belongs to a population of N neurons of the SONN-WN is also incorporated. Let P be the probability that a given neuron becomes the winner, and is hence updated. If the probability of being a winner is the same for every neuron, then

$$P = \frac{1}{N}. \tag{7.4}$$

This greatly simplifies the analysis and is justified for a one-neuron, equilibrium scenario. However, when the full SONN-WN is used to solve COP's, each neuron has a different winning probability depending on the expected cost potential distribution (see Figure 6.2 for example) or even time-varying. For each neuron in the population, the average change to each w_j after each epoch is given by (using (7.3))

$$Av[w_j(t+1) - w_j(t)] = \frac{\exp\left[\left(w_j(t) + Av[\Delta w_j(t)]\right)/T\right]}{\sum_{k=1}^{N} \exp\left(w_k(t)/T\right)} - w_j(t) \tag{7.5}$$

where $Av[\ldots]$ denotes the average of the argument, and

$$\Delta w_j(t) = w_j(t + 1/2) - w_j(t) \; . \tag{7.6}$$

Using (7.1) and (7.4) - (7.6), it follows that

$$Av[\dot{w}_j(t)] = \frac{\exp\left\{\left[w_j(t) + P\beta\left(1 - w_j(t)\right)\right]/T\right\}}{\sum_{k=1}^{N} \exp\left(w_k(t)/T\right)} - w_j(t) \tag{7.7}$$

where $Av[\dot{w}_j(t)]$ is the average rate of change of w_j at any time t. For equilibrium states, we set $Av[\dot{w}_j(t)]=0$ in (7.7). Let $\rho = P\beta$, it follows that

$$\hat{w}_j(t) = \frac{\exp\{[\hat{w}_j(t) + \rho(1-\hat{w}_j(t))]/T\}}{\sum_{k=1}^{N}\exp(\hat{w}_k(t)/T)} \tag{7.8}$$

where $\hat{w}_j(t)$ denotes weights at equilibrium states, and is time dependent for unstable equilibria. The expression of equilibrium states by (7.8) is the main result of this section. It describes the behavior of the equilibrium model of a one-neuron system capturing the long-term weight updating and normalization characteristics of the full SONN-WN described in Chapter 6. This allows us to explain the temperature dependent convergence dynamics presented in the last chapter, which have been shown to play a *decisive* role in affecting solution qualities. Because of the model's nonlinearity, equilibrium states in (7.8) are investigated computationally in the following section.

7.3 A numerical investigation of equilibrium dynamics

We present here the equilibrium states obtained computationally by solving (7.8). Note that any use of $N < 4$ here is for illustrating the equilibrium model only, since for such N the N-queen problem has no feasible solution. In Figures 7.2 to 7.6 we plot the weight values solved numerically by an iterative method given by the procedure depicted in Figure 7.1, with *asynchronous* updating. After 30,000 iterations, the last

50 values were plotted to show the equilibrium states, with the normalization parameter T being varied along the horizontal axis.

Figure 7.2 shows the attained equilibrium states for the case of $N = 1$. In Figure 7.2(a), $\beta = 0.5$ is chosen, and a bifurcation point can be observed at $T = 0.5$, where the system becomes unstable and switches into a two-cycle. If $\beta = 0.95$ is used as in Figure 7.2(b), the bifurcation now occurs at $T = 0.95$. Theoretical analysis of (7.8) shows that bifurcation occurs at $T = \beta$ for the case of $N = 1$.

Figures 7.3 to 7.5 correspond to the case of $N = 2$, with increasing β. In Figure 7.3 a small $\beta = 0.05$ is used. The two weights w_1 and w_2 have the same equilibrium value of around 0.5 (i.e. $1/N$ for $N = 2$). A symmetry-breaking bifurcation occurs at $T^* = 0.5$, where w_1 tends to 0 and w_2 tends to 1 for $T < T^*$. Note that for $T < T^*$, the equilibrium state (w_1, w_2) is still a fixed point in two-dimensional space. Since the system of equations in the equilibrium model (7.8) is symmetrical between w_1 and w_2, the apparent "decision" of which w_j tends to 1 (or 0) is a symmetry-breaking phenomenon in which tiny fluctuations in the convergence process or in the initial conditions can alter the outcome. In general, the parameter T plays the role of a *bifurcation parameter* that determines the structural stability of the equilibrium states, which illustrates how the overall behavior of the system can be changed abruptly by a slight change of a single parameter. Since a small $\beta = 0.05$ is used in Figure 7.3, the equilibrium model (7.8) approaches the actual weight normalization process (6.6) to a certain extent. Indeed, two observations from Figure 7.3 reaffirm this point: (1) the sum of the weights is maintained to be almost exactly 1 across all T values; (2) the weights tend toward 0-1 values for low temperatures. Note that it is for these two characteristics that the weight normalization process (6.6) is used in the SONN-WN for constraint enforcement.

Since the equilibrium model also captures the weight updating process of the SONN-WN, we investigate the effects of the learning rate β on the overall dynamics. For $N = 2$, we use a larger $\beta = 0.5$, and the result is shown in Figure 7.4. A tree-like bifurcation structure can be observed in Figure 7.4(a), with successive bifurcations as T decreases. The two weights have the same value before a symmetry-breaking bifurcation at around $T = 0.47$. This is followed by successive period-doubling bifurcations for each of w_1 and w_2, with the first one occurring at around $T = 0.23$. Let T_{b1} be the temperature at which the first period-doubling bifurcations occur, with successive period-doubling bifurcations at T_{b2}, T_{b3}, etc. At T_{b1} the stable fixed point of (w_1, w_2) becomes unstable, and a two-cycle sets in. This means between T_{b1} and T_{b2}, the system oscillates between two states $(w_1(n), w_2(n))$ and $(w_1(n + 1), w_2(n + 1))$ in successive iterations n and $n + 1$. It can be observed that the condition of $w_1 + w_2 = 1$ no longer holds for $T < T_{b1}$, which is caused by the use of a large β value. This violation only occurs in the equilibrium model, as $\sum_{j=1}^{N} w_j = 1$ is always ensured in the normalization function (6.6) used for the SONN-WN. In the equilibrium model, the approximated normalization (7.2) (or (7.3)) is used, which makes $\sum_{j=1}^{N} w_j \neq 1$ when β is non-zero. Such a discrepancy between the equilibrium model and the actual SONN-WN should not be regarded as a mere inability of the former to fully represent the latter. Rather, the richer dynamics exhibited by the equilibrium model suggest that it should be viewed as a more general dynamical construct than the SONN-WN processes being modeled. In other words, the model not only encompasses the major aspects of the SONN-WN, but also opens up unknown dynamical possibilities.

In order to look more closely into the bifurcation structure, the low temperature region of Figure 7.4(a) is presented with a higher resolution in Figure 7.4(b). The details of successive period-doubling bifurcations can be observed, with intermingled chaotic regimes and windows of periodicities. Such a *cascade* of period-doubling bifurcations leading to chaos is a universal phenomenon that occurs in nonlinear systems of diverse origins (see Section 1.3). One interesting phenomenon can be observed in Figure 7.4(b): for $T > 0.171$ (approx.) each w_j has its own attractor that does not fully merge into the other, which can be judged by the well-separated "trees" of distinct colors. But for $T < 0.171$ each attractor suddenly increases its size, and they merge into a single global attractor, which can be judged by the mixing of the two colors representing w_1 and w_2. This phenomenon shows that the *extent* of interactions between the weights is also T dependent, as evident from the separation and merging of the component attractors.

A different result as shown in Figure 7.5 is obtained when an even larger $\beta = 0.95$ is used. The overall bifurcation structure is presented in Figure 7.5(a) for comparison with previous plots. Apart from the apparent horizontal compression in the overall structure, there are significantly new features previously not found with smaller β's. Figure 7.5(b) is a high-resolution close-up of Figure 7.5(a) for $0.245 < T < 0.26$. Extra bifurcation trees that are smaller in size and "detached" from the periodic branches can be observed around $T = 0.257$ and 0.2515. The full cascade of period-doubling bifurcations is expected to be found inside each of these smaller trees, which is an example of the *self-similarity* feature commonly found in chaotic attractors. A *crisis* can also be seen at around $T = 0.249$ where the attractor suddenly increases in size. These two nonlinear phenomena are common to two-dimensional chaotic attractors [78].

For $N = 5$, the model becomes more relevant to our optimization results of the 5-queen problem in the last chapter, and the equilibrium states are plotted in Figure 7.6. First, a small value of $\beta = 0.05$ is used in Figure 7.6(a). A symmetry-breaking bifurcation can be observed at around $T^* = 0.26$, with a much more step-like transition than the case for $N = 2$ with the same β (Figure 7.3). For $T > T^*$, all the weights have exactly the same value very close to 0.2 (i.e. $1/N$ for $N = 5$), forming a stable fixed point. As T becomes slightly smaller than T^*, the symmetry of the weights is broken, with w_4 tending toward 1 and the rest toward 0. As in the case for $N = 2$, no cascades leading to chaos exist when a small β is used. However, the normalization condition of $\sum_{j=1}^{N} w_j = 1$ is met almost exactly throughout all temperatures.

For $\beta = 0.95$, cascades of bifurcations leading to chaos appear in Figure 7.6(b), with 5 individual bifurcation trees coexisting. The symmetry-breaking bifurcation now occurs at around $T^* = 0.295$, with the first period-doubling bifurcations at around $T_{b1} = 0.175$. As in the case for $N = 2$ with a large β, the normalization condition of

$\sum_{j=1}^{N} w_j = 1$ is violated for $T < T_{b1}$. The merging of individual attractors into a global attractor can also be observed at around $T = 0.12$, where the colored points representing the five components are mixed into each other.

Finally, we present in Figure 7.7 a strange attractor corresponding to the case of $N = 2$, with $\beta = 0.95$ and $T = 0.23$, where w_1 and w_2 are plotted against each other. Such a two-dimensional strange attractor resembles the Hénon map, where repeated stretching and folding of the state space gives rise to its peculiar form [78]. Such a resemblance may suggest a stretching and folding of the state space happening in our case due to repeated updating and normalization described by the equilibrium model.

In summary, we have obtained the equilibrium states by numerically solving the equilibrium model in (7.8) with $N = 1, 2$ and 5. The solution states exhibit various nonlinear dynamical phenomena such as fixed points, symmetry-breaking bifurcation, cascades of period-doubling bifurcations to chaos, and other attractor properties of multi-dimensional nonlinear systems. The nature of these phenomena is found to be heavily dependent on the parameters T and β, with T acting as the bifurcation parameter and β determining the existence of various phenomena. A two-dimensional strange attractor is also revealed, suggesting a dynamical metaphor of stretching and folding of state space that corresponds to the SONN-WN's updating and normalization processes. From these results, it is clear that under certain conditions, the SONN-WN can be seen as a dynamical system with self-organizing properties. This enables us to understand the SONN-WN's performance characteristics from an NSD perspective.

7.4 Discussions

In this section, we discuss the dynamics of the equilibrium model in relation to the optimization performance results presented in Chapter 6. It has been shown in the last chapter that when applying the SONN-WN to solve the N-queen problem, there are certain annealing schedules that yield good quality solutions with a high feasibility measure. More importantly, for these annealing schedules there exists a convergence phase in which the annealing temperature lies within the flexible decision temperature (FDT) interval. It is the purpose of this discussion to seek an explanation for such a temperature dependent response in terms of nonlinear systems dynamics (NSD) revealed by the equilibrium model.

In order to draw on the equilibrium dynamics presented above for discussions of our 5-queen optimization results, we refer to Figure 7.6 where the equilibrium states obtained correspond to $N = 5$. First of all, we use the feasibility plot of Figure 6.7 ((a) or (b)) from the last chapter for illustration. The figure is chosen here because of its parameter setting of $\eta = 0$ (fixed) and $\beta = 0.95$ (fixed), which is the closest SONN-WN configuration to the equilibrium model in Figure 7.6 for $T > T_{b1}$. It can be observed in Figure 6.7 ((a) or (b)) that the horizontal dark band of high feasibilities (> 0.9) spans roughly between $0.24 < T(0) < 0.3$. Thus for the case of Figure 6.7, the FDT interval is around $0.24 < T < 0.3$, which are the normalization temperatures included in the annealing schedules associated with the horizontal band. Now if we look at the equilibrium states as shown in Figure 7.6(b), this temperature range lies within a dynamical regime associated with near 0-1 fixed point convergence, or equivalently $T_{b1} < T < T^*$. Since the equilibrium model approximates the dynamics of the SONN-WN, we claim that the interval $T_{b1} < T < T^*$ of the equilibrium model gives rise to the FDT interval of the SONN-WN.

Our claim above can be justified both experimentally and theoretically. For the case of $\beta = 0.95$ as shown in Figure 7.6(b), we need to show that, at least for $T > T_{b1}$, the obtained equilibrium dynamics realistically reflect the convergence behaviors of the SONN-WN. From the feasibility plot of Figure 6.7 ((a) or (b)), it can be seen that the feasibility drops sharply for $T(0) > 0.3$. Using our equilibrium dynamics in Figure 6.7(b), this corresponds to the symmetry-bifurcation at around $T^* = 0.295$, at which there is a sudden transition of the fixed point configurations from perfect symmetry (all w_j's identical) to broken-symmetry (w_4 tends to 1, others tend to 0). For our arguments to hold, we now need to show that the symmetrically configured fixed point gives rise to poor solution quality of the SONN-WN, and that the one with 0-1 configuration leads to good solution quality. Theoretically speaking, it is clear that the weights of the SONN-WN must converge toward 0-1 values in order to yield good quality solutions leading to high feasibilities, thus proving our case. Experimentally, the supporting evidence can be found in Figure 6.10 and Figure 6.13, which show that under-convergence phases are detrimental to effective convergence toward good quality solutions. Moreover, the sharp contrast in the distribution of weight values before and after T^* in the equilibrium model (Figure 7.6(b)) is visually reflected in Figure 6.10, where the weights start with an under-convergence phase when T is above the FDT interval, then suddenly transit into an effective optimization phase when T falls within the FDT interval. It is due to such evidence and justifications that we conclude the following: the FDT interval observed in Chapter 6 is a consequence of nonlinear system dynamics of the SONN-WN, and has its origin from the interval $T_{b1} < T < T^*$ in the equilibrium model with the associated 0-1 fixed point convergence. An interesting analogy exists in mean field annealing models where a similar normalization procedure was used to improve optimization [104]. An analogous phenomenon of effective optimization happening at a so-called *critical temperature* (T_c) was observed, and T_c was evaluated using mean field theory. Despite the many differences between our model and the mean field annealing models, it is interesting to see that the similar normalization procedures used in two different approaches give similar results. Once again, this illustrates the intricate relationship between the two approaches in the study of neural networks for combinatorial optimization, namely, the dynamical systems view and the statistical mechanics view.

7.5 Concluding remarks

By focusing on the essential aspects of the SONN-WN's weight updating and normalization process, we have theoretically constructed an equilibrium model of the SONN-WN to describe its long-term convergence dynamics. The equilibrium states obtained from the model reveal a rich collection of nonlinear phenomena controlled by the SONN-WN parameters T and β. In particular, the symmetry-breaking bifurcation of equilibrium states with bifurcation parameter T has been found to be the key mechanism underlying the convergence characteristics of the SONN-WN. By combining evidence from the optimization results in the last chapter and the theoretical model in this chapter, we have deduced that the flexible decision temperature interval, which characterizes effective convergence during optimization, has its dynamical origin of $T_{b1} < T < T^*$. It has been shown that such a temperature range corresponds to the regime of fixed points with 0-1 broken-symmetry, which acts as an attracting force toward 0-1 convergence for the most effective annealing temperatures of the SONN-WN. Thus the equilibrium model has opened up a dynamical perspective for understanding and enhancing the optimization performance of the SONN-WN.

For certain parameter regimes, the equilibrium model extends its dynamical possibilities beyond fixed points and into the realms of more general nonlinear phenomena, such as cascades of period-doubling bifurcations to chaos, crisis, and merging of local attractors into a global attractor. Surprisingly, these phenomena have been previously found to be the underlying dynamics behind the chaotic search mechanism in Hopfield-type chaotic neural networks [99]. This may suggest a possible chaotic search mechanism to be devised for the SONN-WN, with the nonlinear phenomena revealed herein as functional elements. Indeed, some other evidence already exists to further support such a possibility. For example, the weight dynamics of Kohonen's SOFM have been examined in the light of self-organized criticality, which is a class of noise sensitive dynamical systems whose critical states feature power law and fractal characteristics [36, 27, 16]. A SONN-WN with attractor nodes has also been proposed [64], in which the iterative logistic map, well known for its nonlinear properties, is utilized for enhancing solution qualities of COP's. It is thus anticipated that a dynamical systems or complex systems viewpoint of the general self-organizing approach to COP's would lead to important developments in the future.

stage	w_j					process
	j =1	2	3	...	N	
t :	◆	◆	◆	...	◆	———
t + ½ :	❖	◆	◆	...	◆	updating w_1
t + 1 :	●	◆	◆	...	◆	approximate normalization

symbols: ◆ initial weight ❖ updated weight ● "normalized" weight

Figure 7.1: Schematic diagram of the equilibrium model showing weight updating for $j^* = 1$, followed by an approximate normalization.

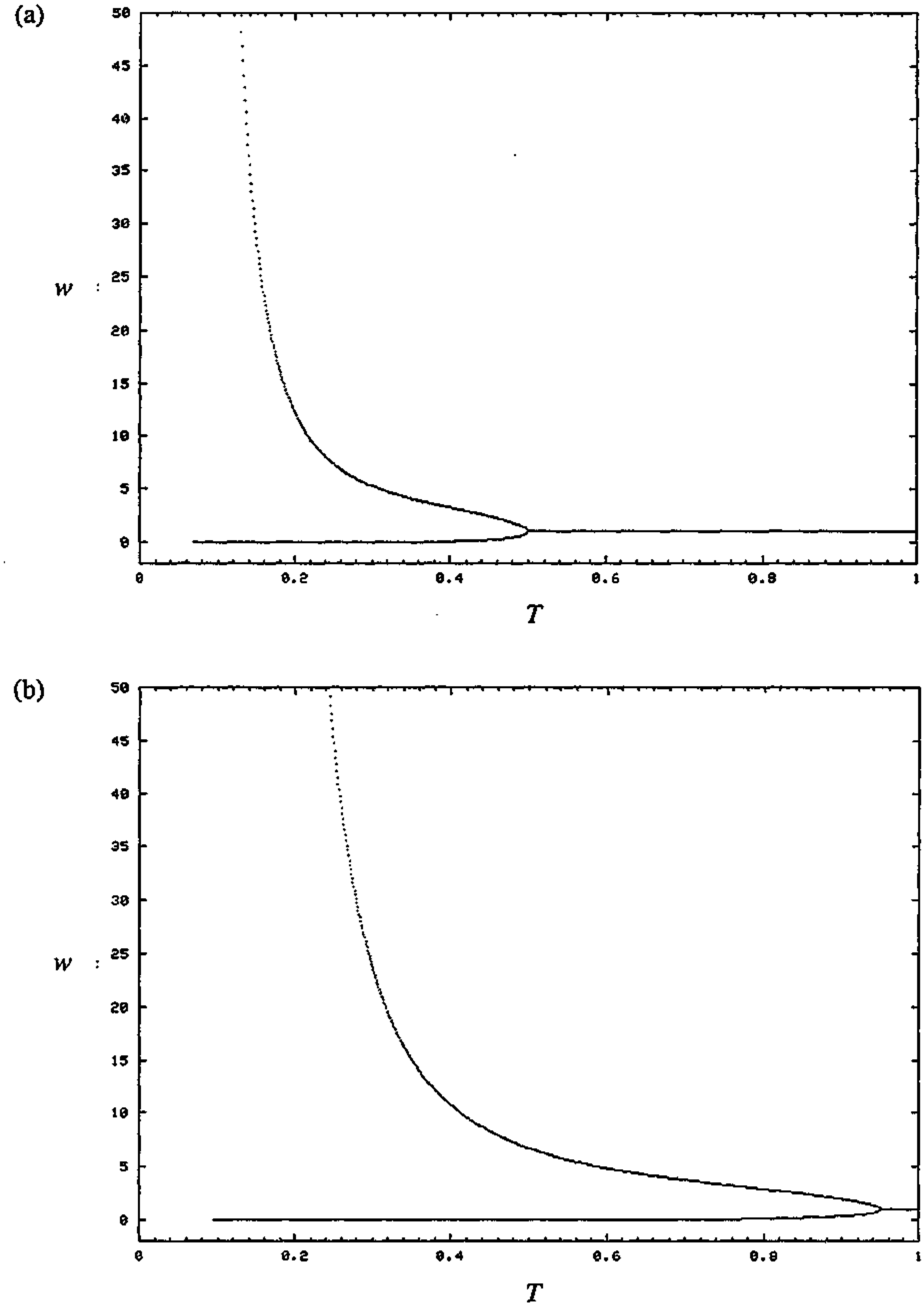


Figure 7.2: Bifurcation diagram of w with $N = 1$ and (a) $\beta = 0.5$; (b) $\beta = 0.95$.

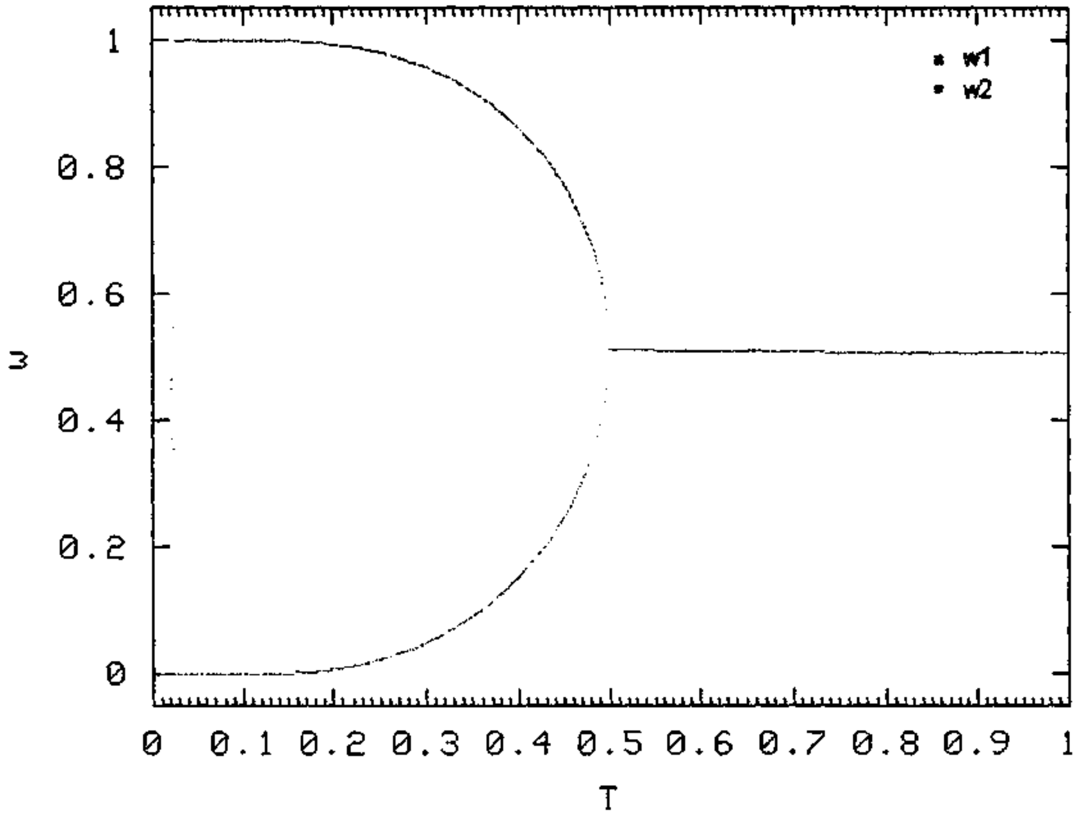


Figure 7.3: Bifurcation diagram of w_1 and w_2 with $N = 2$ and $\beta = 0.05$.

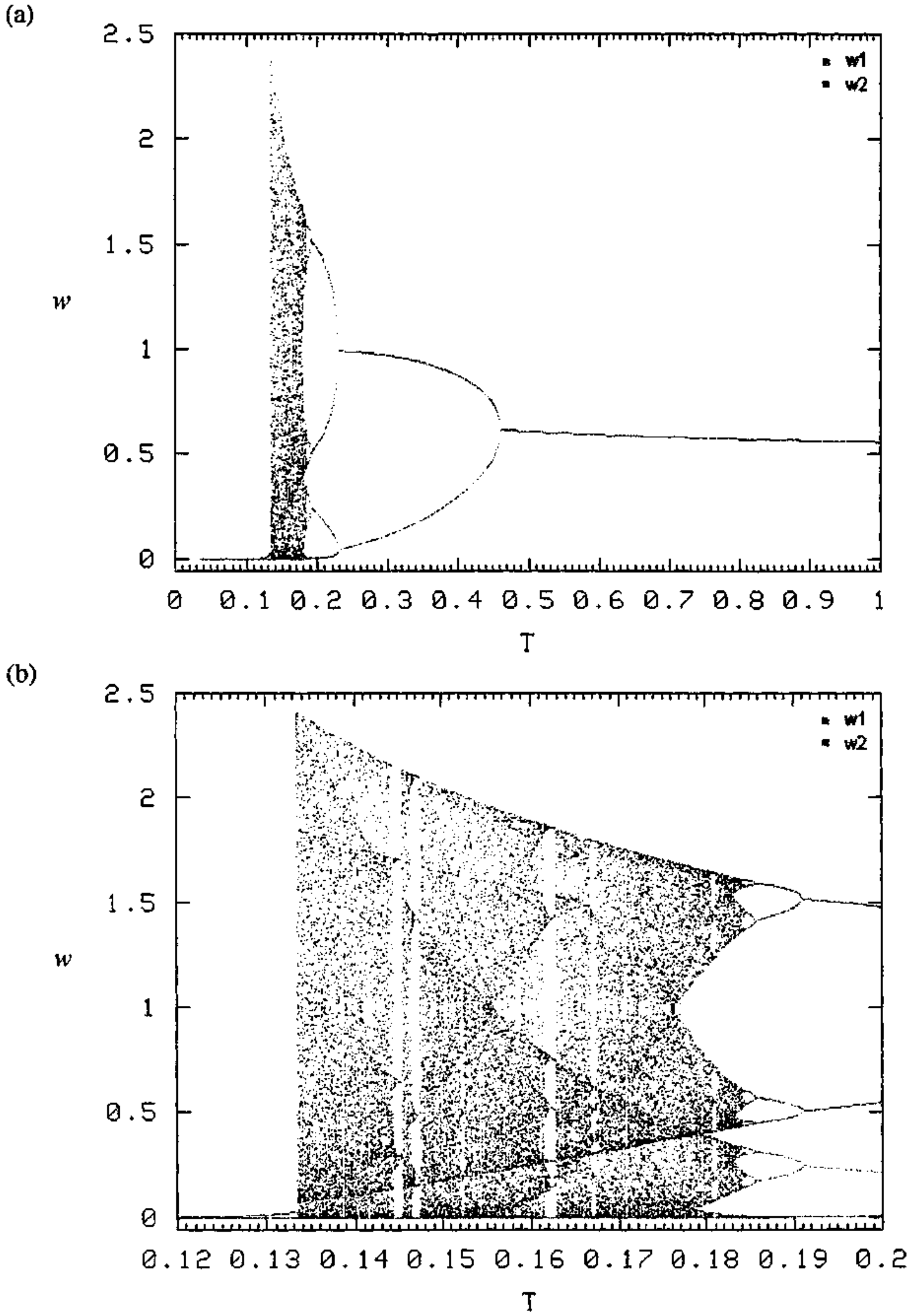


Figure 7.4: Bifurcation diagram of w_1 and w_2 with $N = 2$ and $\beta = 0.5$.

(b) is a close-up of (a).

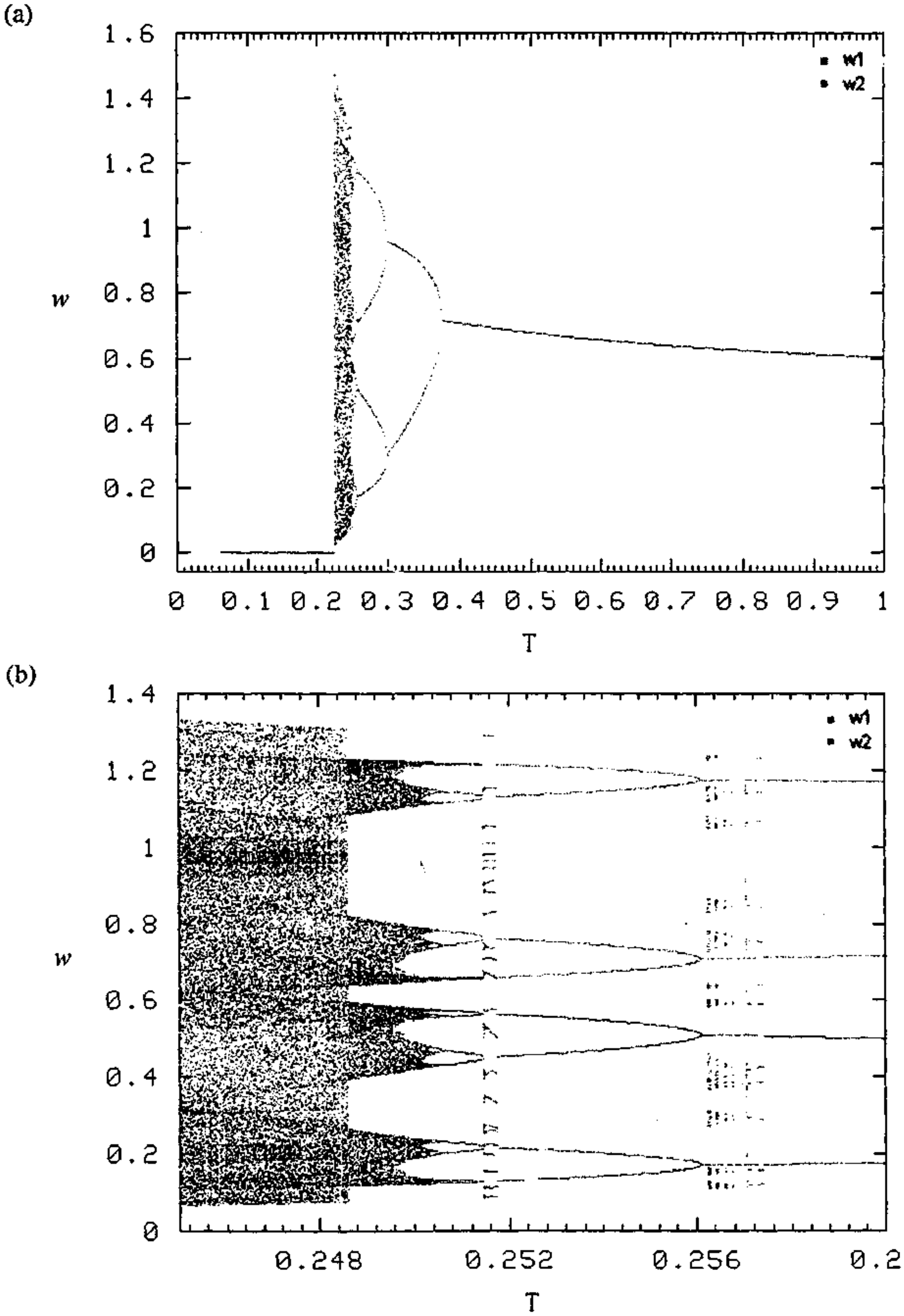


Figure 7.5: Bifurcation diagram of w_1 and w_2 with $N = 2$ and $\beta = 0.95$.

(b) is a close-up of (a).

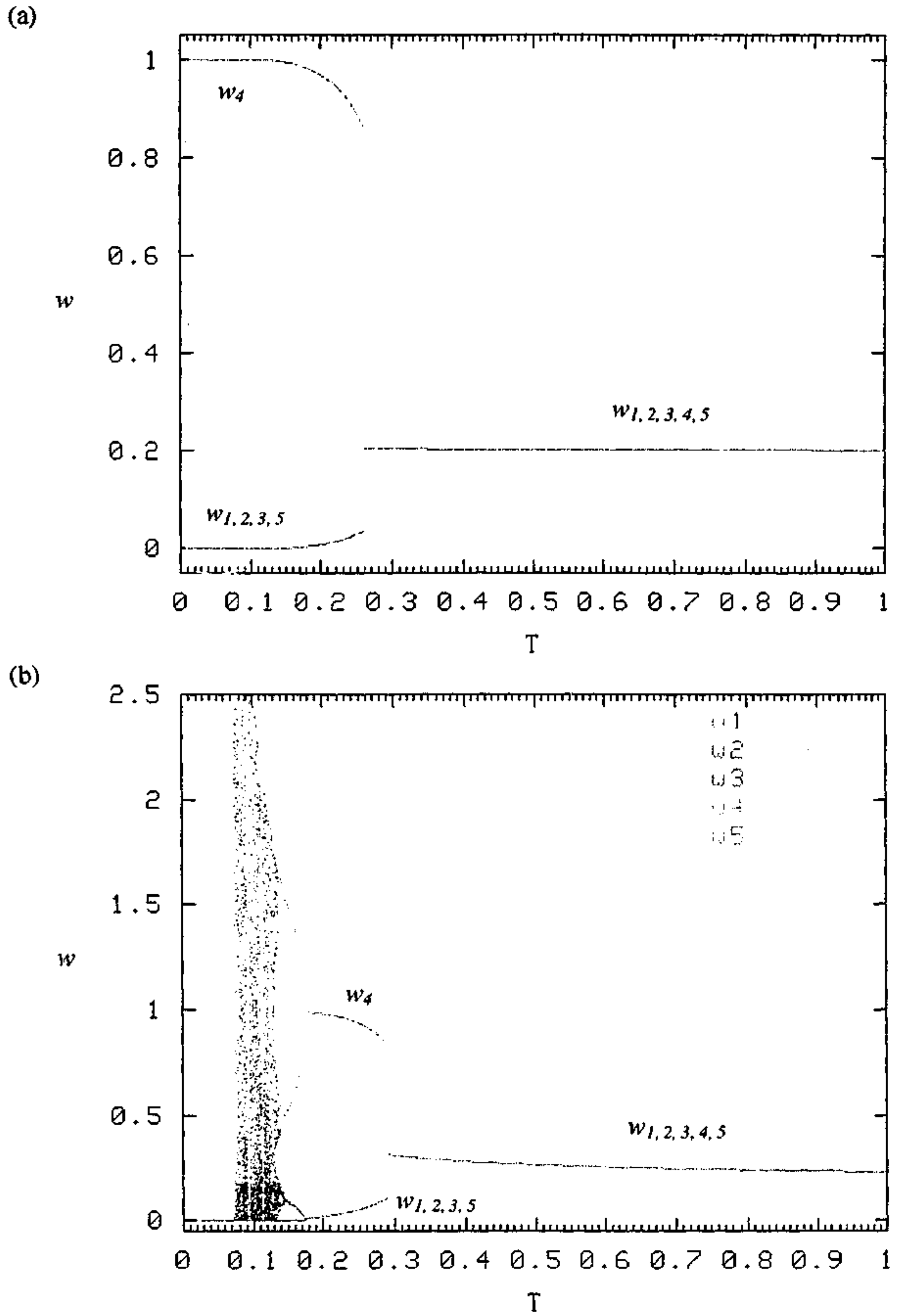


Figure 7.6: Bifurcation diagram of $w_{1,2,3,4,5}$ with $N = 5$ and (a) $\beta = 0.05$; (b) $\beta = 0.95$.

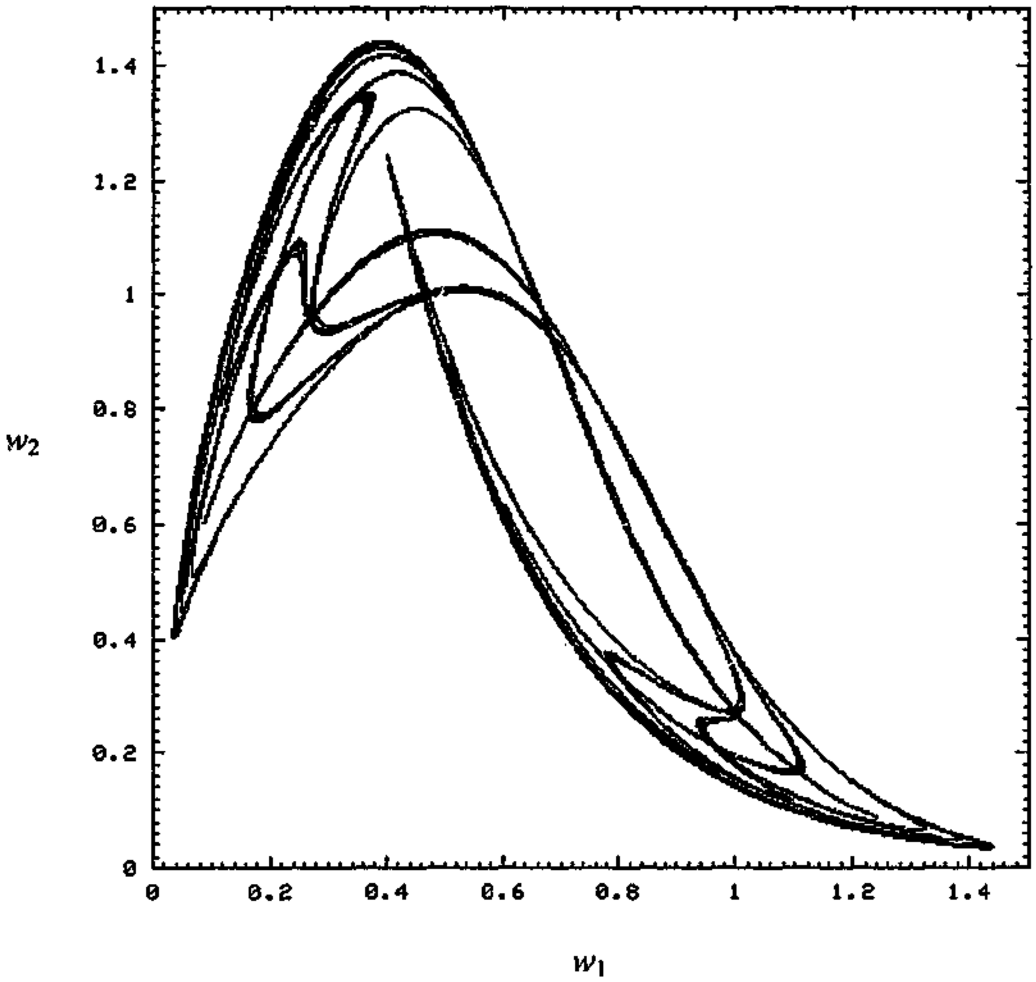


Figure 7.7: A strange attractor in (w_1, w_2) with $N = 2$, $\beta = 0.95$ and $T = 0.23$.

Chapter 8

Conclusions

8.1 Outcomes of the thesis

This thesis has been concerned with neural networks featuring nonlinear system dynamics (NSD) for solving combinatorial optimization problems (COP's). We have investigated the roles of NSD in enhancing the optimization performance of two major neural approaches, namely the Hopfield-based and the self-organizing neural networks. As outlined in Chapter 1, there is a current need to fine-tune neural network algorithms for improved convergence toward optimal solutions, so that they can be more effective tools for practical combinatorial optimization. Thus a theoretical point of view has been adopted in our study, focusing on how nonlinear phenomena arise in the networks, their properties, and their functional roles in the solution process. In other words, our investigation can be seen as an attempt to extend current neural approaches to include more general dynamics for the purpose of solving COP's more effectively. To support our theoretical study, the *N*-queen problem has been used throughout the thesis to produce experimental results that expose various differences in optimization capabilities and the associated network dynamics. Once these behaviors have been exposed, they can be explained by the theoretical insights provided by the thesis. Thus the computational results presented are not intended to be exhaustive, and merely serve to expose phenomena worthy of explanation.

For the traditional Hopfield-type approaches to combinatorial optimization, the main limitation lies in the Hopfield network's steepest descent dynamics, which causes the Hopfield-Tank (H-T) method to result in local minimum solutions. The optimization performance is further affected by the sensitive dependence of solution feasibility on the choice of penalty parameters in the H-T formulation. Over the years, a number of measures have been devised to improve the optimization performance by targeting these two shortcomings of the Hopfield-type approaches. In Chapter 2, a new breed of neural networks called the Hopfield-type chaotic neural networks (HCNN's) were discussed. These Hopfield-based networks also use the H-T approach for optimization, but the associated limitations are overcome by the introduction of chaotic dynamics in the networks. In this new direction, the HCNN models employ global search dynamics with chaotic properties to escape local minima during the convergence process, as well as to alleviate the sensitivity problem of the penalty parameters.

We have broadly classified the HCNN models into two major classes: one is the internal approach, where chaos is generated within the network; the other is the external approach, where an externally generated chaotic signal is added to the network as perturbation. Three models belonging to the internal approach have been discussed: using chaotic neurons with strong feedback (Chen-Aihara model); adding a negative self-feedback term to the Euler-discretized Hopfield model (Nozawa model); and using a sufficiently large time-step to the Euler-discretized Hopfield model (Wang-Smith model). The close relationships between these models suggest that they share a common chaotic search mechanism proposed by Tokuda *et al* [99]. Moreover, the Chen-Aihara model and the Wang-Smith model both incorporate chaotic simulated annealing (CSA) for effective convergence. On the other hand, HCNN models belonging to the external approach have a very different way of escaping local minima, which is based on the autocorrelation property in the chaotic noise.

In order to study systematically the various HCNN models having different means of generating a~d utilizing chaos, we have proposed in Chapter 3 a theoretical framework that unifies the two classes of HCNN models. The framework is based on the addition of an energy modifier term to the original Hopfield energy function, thus forming a new energy landscape that characterizes various chaotic dynamics. By inspecting the form of the energy modifier, we have been able to gain insights into the relationships between the structure of the energy landscapes and the corresponding optimization dynamics. This global perspective on the classification of the HCNN models has allowed us to account for the various differences between the internal and external approaches of HCNN models in terms of the different energy modifiers. New insights concerning the convergence properties of the CSA models have also been obtained from the energy landscape perspective.

In Chapter 4, the dynamical behaviors and optimization performance of the HCNN models were investigated. We have implemented the three models described by the unified framework to solve the *N*-queen problem via computer simulation. The two CSA models have been found to have strong optimization performance by having good feasibility, efficiency, robustness and scalability. By combining parametric studies of these measurements with observations of the dynamical behaviors during the solution process, we have concluded that the characteristic chaotic dynamics of the CSA models are beneficial to optimization. Similar experiments with the chaotic noise model have also shown an improved performance, but with lesser robustness and scalability. Important insights concerning the specific properties of the HCNN models have also been obtained by examining the experimental results in the light of the unified framework.

Apart from the Hopfield-based networks, we have also included the self-organizing approaches for combinatorial optimization in our study. Although algorithms employing the self-organizing feature map principle of Kohonen have been shown by various researchers to be capable of solving COP's, most of them are limited to solving Euclidean problems. We have discussed in Chapter 5 a collection of these approaches, together with a more recent development of the self-organizing neural network (SONN) which is capable of solving general 0-1 problems. Common among these self-organizing approaches is the solution quality problem caused by unwanted oscillations during the competition-based solution process. For this we have focused on the SONN with weight normalization (SONN-WN), whose intricate interactions between the Kohonen's updating and normalization processes have prompted our investigations into the role of NSD in improving the solution quality.

We began our investigation into the relationship between the SONN-WN's dynamics and its optimization performance in Chapter 6 by implementing the algorithm to solve the *N*-queen problem computationally. Through the parametric study of annealing schedules and feasibilities, we have highlighted the importance of symmetry issues in cost potentials, from which we have demonstrated the effectiveness of using noise and small neighborhoods for improvements in feasibility and robustness. By combining the feasibility patterns revealed in the parametric study with a detailed investigation of the convergence dynamics, we have discovered the normalization conditions for ensuring effective convergence to high quality solutions. The discovery has also allowed us to explain the noise-induced mechanism of oscillation control during convergence, as well as providing a guide for choosing annealing schedules of the normalization process.

By capturing the essential aspects of the SONN-WN's weight updating and normalization process, we have derived an equilibrium model of the SONN-WN in Chapter 7, such that the network's long-term convergence dynamics can be described theoretically. From a NSD perspective of the SONN-WN, we have been able to show the key role of bifurcation dynamics in characterizing the convergence behaviors shown in our experimental results. Together with other nonlinear phenomena obtained, the equilibrium model has opened up a dynamical perspective for understanding and enhancing the optimization performance of the SONN-WN.

8.2 Suggestions for future research

In this thesis, we have investigated the respective ways in which NSD are used to improve the optimization performance of the two major neural approaches. One of the advantages of neural networks over other methods for solving COP's, like heuristics, is their parallel architecture of simple elements, which makes them suitable for hardware implementation. As can be seen from this thesis, neural networks using NSD for performance improvements require only simple modifications to existing neural architectures, thus future research in hardware deployments should study the incorporation of NSD for enhanced performance.

On the other hand, further testing of the models described in this thesis should be performed by applications to larger problem sizes as well as to other COP's, so that the models can be made more robust and flexible for practical use. This is particularly needed for the SONN-WN, where larger problem instances should see a benefit from our proposed guide to annealing schedules for effective convergence.

On theoretical fronts, the unified framework of HCNN's may be used as a starting point for the development of new models, or as a unifying theme when examining further experimental results. Obviously, the framework itself relies on further research for refinement. For the SONN-WN, more research should be directed into the potential exploitation of the general nonlinear phenomena revealed by the equilibrium model for improving optimization, much as the chaotic search scenario has been used among HCNN's. Also, it would be worthwhile to investigate if the interesting dynamics of the equilibrium model could be found in a simple gradient descent technique by incorporating a similar normalization process. In general, for HCNN's and SONN's alike, further research into the properties of solution state attractors and their perturbation by noise should lead to a better understanding of the fundamental role of NSD in solving difficult COP's.

8.3 Concluding remarks

In conclusion, this thesis has studied the theoretical foundations of neural networks with nonlinear system dynamics for combinatorial optimization. The generation, characterization, and exploitation of nonlinear system dynamics for improving optimization performance have been investigated for both the Hopfield-type and self-organizing neural approaches. Under the proposed unifying framework, we have computationally demonstrated the effectiveness of chaotic dynamics in enhancing the performance of Hopfield-type neural networks. Through a deeper understanding of the SONN-WN convergence process via the proposed equilibrium model, the capability of bifurcation dynamics in improving solution quality has been proven both theoretically and experimentally. Based on these results, neural approaches with nonlinear system dynamics should see a promising future in practical combinatorial optimization.

Bibliography

[1] E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, Essex, 1989.

[2] S. Abe, "Global convergence and suppression of spurious states of the Hopfield neural networks", *IEEE Transactions on Circuits and Systems*, vol. 40, no. 4, pp. 246-257, 1993.

[3] B. Abramson and M. Yung, "Divide and conquer under global constraints: A solution to the *N*-queens problem", *Journal of Parallel and Distributed Computing*, vol. 6, no. 3, pp. 649-662, 1989.

[4] K. Aihara, "Chaos in neural response and dynamical neural network models: Towards a new generation of analog computing", in *Towards the Harnessing of Chaos*, M. Yamaguti (ed.), pp. 83-98, Elsevier Science, Amsterdam, 1994.

[5] K. Aihara, T. Takabe and M. Toyoda, "Chaotic neural networks", *Physics Letters A*, vol. 144, no. 6,7, pp. 333-340, 1990.

[6] S. V. Aiyer, M. Niranjan and F. Fallside, "A theoretical investigation into the performance of the Hopfield model", *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 204-215, 1990.

[7] Y. Akiyama, Y. Takefuji and H. Aiso, "Conductance programmable neural chips employing switched resistors", *Proceedings of the 3rd Joint Technical Conference on Circuits, Systems, Computers and Communications*, November, 1988.

[8] Y. Akiyama, A. Yamashita, M. Kajiura and H. Aiso, "Combinatorial optimization with Gaussian machines", *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 533-540, 1989.

[9] Y. Akiyama, A. Yamashira, M. Kajiura, Y. Anzai and H. Aiso, "The Gaussian machine: A stochastic neural network for solving assignment problems", *Journal of Neural Network Computing*, vol. 2, pp. 43-51, 1991.

[10] I. K. Altinel, N. Aras and B. J. Oommen, "Fast, efficient and accurate solutions to the Hamiltonian path problem using neural approaches", *Computers and Operations Research*, vol. 27, pp. 461-494, 2000.

[11] B. Angéniol, G. de La Croix Vaubois and J. Y. Le Texier, "Self-organizing feature maps and the travelling salesman problem", *Neural Networks*, vol. 1, no. 4, pp. 289-293, 1988.

[12] S. M. Aourid, X. D. Do and B. Kaminska, "Penalty formulation for 0-1 linear programming problem: A neural network approach", *Proceedings of the International Conference on Neural Networks*, vol. 4, pp. 1690-1693, 1995.

[13] N. Aras, B. J. Oommen and I. K. Altinel, "The Kohonen network incorporating explicit statistics and its application to the travelling salesman problem", *Neural Networks*, vol. 12, pp. 1273-1284, 1999.

[14] M. A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, Massachusetts, 1995.

[15] H. Asai, K. Onodera, T. Kamio and H. Ninomiya, "A study of Hopfield neural networks with external noises", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1584-1589, 1995.

[16] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality", *Physical Review A*, vol. 38, no. 1, pp. 364-374, 1988.

[17] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[18] S. Bharitkar and J. M. Mendel, "The hysteretic Hopfield neural network", *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 879-888, 2000.

[19] C. Bouton and G. Pagès, "Self-organization and convergence of the one-dimensional Kohonen algorithm with non uniformly distributed stimuli", *Stochastic Processes and Their Applications*, vol. 47, pp. 249-274, 1993.

[20] R. D. Brandt, Y. Wang , A. J. Laub and S. K. Mitra, "Alternative networks for solving the travelling salesman problem and the list-matching problem", *Proceedings of the International Conference on Neural Networks*, vol. 2, pp. 333-340, 1988.

[21] L. Burke, "Neural methods for the Traveling Salesman Problem: Insights from operations research", *Neural Networks*, vol. 7, no. 4, pp. 681-690, 1994.

[22] L. Burke and P. Damany, "The guilty net for the travelling salesman problem", *Computers and Operations Research*, vol. 19, no. 3/4, pp. 255-265, 1992.

[23] S. Cavalieri and M. Russo, "Improving Hopfield neural network performance by fuzzy logic-based coefficient tuning", *Neurocomputing*, vol. 18, no. 1-3, pp. 107-126, 1998.

[24] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos", *Neural Networks*, vol. 8, no. 6, pp. 915-930, 1995.

[25] L. Chen and K. Aihara, "Chaos and asymptotical stability in discrete-time neural networks", *Physica D*, vol. 104, pp. 286-325, 1997.

[26] M. Cottrell and J. C. Fort, "A stochastic model of retinotopy: A self-organizing process", *Biological Cybernetics*, vol. 53, pp. 405-411, 1986.

[27] I. Csabai, T. Geszti and G. Vattay, "Criticality in the one-dimensional Kohonen neural map", *Physical Review A*, vol. 46, no. 10, pp. R6181-6184, 1992.

[28] G. W. Davis, "Sensitivity analysis in neural net solutions", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, pp. 1078-1082, 1989.

[29] D. DeSieno, "Adding a conscience to competitive learning", *Proceedings of the International Conference on Neural Networks*, vol. I, pp. 117-124, 1988.

[30] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method", *Nature*, vol. 326, pp. 689-691, 1987.

[31] S. Eberhardt, T. Daud, D. Kerns, T. Brown and A. Thakoor, "Competitive neural architecture for hardware solution to the assignment problem", *Neural Networks*, vol. 4, no. 4, pp. 431-442, 1991.

[32] E. Erwin, K. Obermayer and K. Schulten, "Self-organizing maps: Stationary states, metastability and convergence rate", *Biological Cybernetics*, vol. 67, no. 1, pp. 35-45, 1992.

[33] E. Erwin, K. Obermayer and K. Schulten, "Self-organizing maps: Ordering, convergence properties and energy functions", *Biological Cybernetics*, vol. 67, no. 1, pp. 47-55, 1992.

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

[35] G. W. Flake, *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*, MIT Press, Cambridge, Massachusetts, 1999.

[36] J. A. Flanagan, "Neuron weight dynamics in the SOM and self-organized criticality", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. V, pp. V-39-44, 2000.

[37] J. C. Fort, "Solving a combinatorial problem via self-organizing process: An application of the Kohonen algorithm to the travelling salesman problem", *Biological Cybernetics*, vol. 59, pp. 33-40, 1988.

[38] J. C. Fort and G. Pagès, "Convergence of the Kohonen maps: A dynamical system approach", *Proceedings of the International Conference on Artificial Neural Networks*, pp. 631-636, 1997.

[32] E. Erwin, K. Obermayer and K. Schulten, "Self-organizing maps: Stationary states, metastability and convergence rate", *Biological Cybernetics*, vol. 67, no. 1, pp. 35-45, 1992.

[33] E. Erwin, K. Obermayer and K. Schulten, "Self-organizing maps: Ordering, convergence properties and energy functions", *Biological Cybernetics*, vol. 67, no. 1, pp. 47-55, 1992.

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

[35] G. W. Flake, *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*, MIT Press, Cambridge, Massachusetts, 1999.

[36] J. A. Flanagan, "Neuron weight dynamics in the SOM and self-organized criticality", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. V, pp. V-39-44, 2000.

[37] J. C. Fort, "Solving a combinatorial problem via self-organizing process: An application of the Kohonen algorithm to the travelling salesman problem", *Biological Cybernetics*, vol. 59, pp. 33-40, 1988.

[38] J. C. Fort and G. Pagès, "Convergence of the Kohonen maps: A dynamical system approach", *Proceedings of the International Conference on Artificial Neural Networks*, pp. 631-636, 1997.

[39] A. H. Gee, "Problem solving with optimization networks", Ph.D. thesis, Queen's College, Cambridge, 1993.

[40] R. E. Gomory, "Outline for an algorithm for integer solution to linear programs", *Bulletin of the American Mathematical Society*, vol. 14, pp. 275-278, 1958.

[41] C. Grebogi, E. Ott, F. Romeiras and J. A. York, "Criticai exponents for crisis-induced intermittency", *Physical Review A*, vol. 36, pp. 5365-5380, 1987.

[42] C. Grebogi, E. Ott and J. A. York, "Chaotic attractors in crisis", *Physical Review Letters*, vol. 48, pp. 1507-1510, 1982.

[43] S. Grossberg and G. A. Carpenter, "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.

[44] F. Guerrero, S. Lozano, K. A. Smith and T. Kwok, "Manufacturing cell formation using a new self-organizing neural network", *The 26th International Conference on Computers & Industrial Engineering*, vol. 1, pp. 668-672, 1999.

[45] F. Guerrero, K. A. Smith and S. Lozano, "Self-organizing neural approach for solving the quadratic assignment problem", *Proceedings of the International Workshop on Soft Computing in Industry '99*, pp. 13-18, 1999.

[46] R. M. Haralick and G. L. Elliot, "Increasing tree search efficiency for constraint satisfaction problems", *Artificial Intelligence*, vol. 14, pp. 263-313, 1980.

[47] M. Hasegawa, T. Ikeguchi, T. Matozaki and K. Aihara, "An analysis on additive effects of nonlinear dynamics for combinatorial optimization", *IEICE*

Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E80-A, no. 1, pp. 206-213, 1997.

[48] Y. Hayakawa, A. Marumoto and Y. Sawada, "Effects of the chaotic noise on the performance of neural network model for optimization problems", *Physical Review E*, vol. 51, no. 4, pp. R2693-R2696, 1995.

[49] S. Hegde, J. Sweet and W. Levy, "Determination of parameters in a Hopfield/Tank computational network", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 291-298, 1988.

[50] R. C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*, Oxford University Press, New York, 1994.

[51] G. E. Hinton, T. J. Sejnowski and D. H. Ackley, "Boltzmann machines: Constraint satisfaction networks that learn", Carnegie Melon University Technical Report CMU-CS-84-119, 1984.

[52] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, pp. 2554-2558, 1982.

[53] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 81, pp. 3088-3092, 1984.

[54] J. J. Hopfield and D. W. Tank, " "Neural" computation of decisions in optimization problems", *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.

[55] J. Hynek, "The *N*-queens problem revisited", *Proceedings of the ICSC Symposia on Intelligent Systems and Application (ISA'2000)*, paper no. 1513-159, 2000.

[56] L. V. Kale, "An almost perfect heuristic for the *N* nonattacking queens problem", *Information Processing Letters*, vol. 34, no. 4, pp. 173-178, 1990.

[57] B. Kamgar-Parsi and B. Kamgar-Parsi, "Dynamical stability and parameter selection in neural optimization", *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 566-571, 1992.

[58] K. Kaneko, "Clustering, coding, switching, hierarchical, ordering and control in a network of chaotic elements", *Physica D*, vol. 41, pp. 137-172, 1990.

[59] K. Kaneko, "Globally coupled circle maps", *Physica D*, vol. 54, pp. 5-19, 1991.

[60] R. M. Karp, "On the complexity of combinatorial problems", *Networks*, vol. 5, pp. 45-68, 1975.

[61] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimisation by simulated annealing", *Science*, vol. 220, pp. 671-680, 1983.

[62] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

[63] T. Kohonen, *Self-Organizing Maps*, Series in Information Sciences 30, Springer-Verlag, Heidelberg, 1995.

[64] T. Kwok and K. A. Smith, "A self-organizing neural network with attractor nodes for combinatorial optimization", in *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems*, C. Dagli *et al* (eds.), ASME Press, NJ, 2000.

[65] W. K. Lai and G. G. Coghill, "Genetic breeding of control parameters for the Hopfield/Tank neural net", *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 618-623, 1992.

[66] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems", *Econometica*, vol. 28, pp. 497-520, 1960.

[67] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Khan and D. B. Shmoys (eds.) *The Traveling Salesman Problem*, Wiley, New York, 1985.

[68] S. Z. Li, "Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers", *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1507-1516, 1996.

[69] J. T-H. Lo, "A new approach to global optimization and its applications to neural networks", *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 600-605, 1992.

[70] M. Martín-Valdivia, A. Ruiz-Sepúlveda and F. Triguero-Ruiz, "Improving local minima of Hopfield networks with augmented Lagrange multipliers for large scale TSPs", *Neural Networks*, vol. 13, pp. 283-285, 2000.

[71] T. Matsumoto, L. O. Chua and M. Komuro, "The double scroll", *IEEE Transactions on Circuits and Systems*, vol. CAS-32, no. 8, pp. 798-818, 1985.

[72] W. A. McCulloch and W. Pitts, "A logical calculus of the ideas imminent in nervous activity", *Bulletin Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.

[73] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, Massachusetts, 1969.

[74] H. Nonaka and Y. Kobayashi, "Sub-optimal solution screening in optimization by neural networks", *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 606-611 1992.

[75] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos", *Chaos*, vol. 2, no. 3, pp. 377-386, 1992.

[76] H. Nozawa, "Solution of the optimization problem using the neural network model as a globally coupled map", in *Towards the Harnessing of Chaos*, M. Yamaguti (ed.), pp. 99-114, Elsevier Science, Amsterdam, 1994.

[77] J. Pearl, *Heuristics*, Addison-Wesley, Reading, MA, 1984.

[78] H-O. Peitgen, H. Jürgens and D. Saupe, *Chaos and Fractals*, Springer-Verlag, New York, 1992.

[79] C. Peterson and B. Soderberg, "Artificial neural networks", in *Modern Heuristic Techniques for Combinatorial Problems*, C. R. Reeves (ed.), Blackwell Scientific Publications, Oxford, 1993.

[80] L. K. Platzman and J. J. Bartholdi, "Spacefilling curves and the planar travelling salesman problem", *Journal of the ACM*, vol. 36, no. 4, pp. 719-737, 1989.

[81] J. Y. Potvin, "The travelling salesman problem: A neural network perspective", *ORSA Journal on Computing*, vol. 5, pp. 328-348, 1993.

[82] C. R. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, 1993.

[83] H. Ritter, T. Martinetz and K. Schulten, *Neural Computation and Self-Organizing Maps*, Addison-Wesley, Reading, MA, 1992.

[84] H. Ritter and K. Schulten, "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection", *Biological Cybernetics*, vol. 60, pp. 59-71, 1988.

[85] B. Sheu, E. Chou, R. Tsai and D. Chen, "VLSI neural networks: Design challenges and opportunities", in *Computational Intelligence*, M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel and T. Fukada (eds.), IEEE Press, New York, 1995.

[86] B. M. Smith and M. E. Dyer, "Locating the phase transition in binary constraint satisfaction problems", *Artificial Intelligence*, vol. 81, pp. 155-181, 1996.

[87] K. A. Smith, "Solving the generalised quadratic assignment problem using a self-organising process", *Proceedings of the IEEE International Conference on Neural Networks*, vol. IV, pp. 1876-1879, 1995.

[88] K. A. Smith, "Neural networks for combinatorial optimisation: A review of more than a decade of research", *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 15-34, 1999.

[89] K. A. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, 1997.

[90] K. A. Smith, M. Palaniswami and M. Krishnamoorthy, "A hybrid neural approach to combinatorial optimization", *Computers and Operations Research*, vol. 23, pp. 597-610, 1996.

[91] K. A. Smith, M. Palaniswami and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications", *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1301-1318, 1998.

[92] H. Stone and J. Stone, "Efficient search techniques – an empirical study of the *N*-queens problem", *IBM Journal of Research and Development*, vol. 31, no. 4, 1987.

[93] H. Szu, "Fast TSP algorithm based on binary neuron output and analog input using zero-diagonal interconnect matrix and necessary and sufficient conditions of the permutation matrix", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 259-266, 1988.

[94] H. Szu and R. Hartley, "Fast simulated annealing", *Physics Letters A*, vol. 122, pp. 157-162, 1987.

[95] G. A. Tagliarini and E. W. Page, "Solving constraint satisfaction problems with neural networks", *Proceedings of the IEEE International Conference on Neural Networks*, vol. III, pp. 741-747, 1987.

[96] Y. Takefuji and H. Szu, "Design of parallel distributed Cauchy machines", *Proceedings of the International Joint Conference on Neural Networks*, vol. I, pp. 529-532, 1989.

[97] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian and J. D. Farmer, "Testing for nonlinearity in time series: The method of surrogate data", *Physica D*, vol. 58, pp. 77-94, 1992.

[98] I. Tokuda, K. Aihara and T. Nagashima, "Adaptive annealing for chaotic optimization", *Physical Review E*, vol. 58, no. 4, pp. 5157-5160, 1998.

[99] I. Tokuda, T. Nagashima and K. Aihara, "Global bifurcation structure of chaotic neural networks and its application to Travelling Salesman Problems", *Neural Networks*, vol. 10, no. 9, pp. 1673-1690, 1997.

[100] I. Tokuda, A. Tamura, R. Tokunaga, K. Aihara and T. Nagashima, "Learning algorithm for chaotic dynamical systems that solve optimization problems", *Electronics and Communications in Japan*, part 3, vol. 82, no. 3, pp. 10-21, 1999.

[101] T. Ueda, K. Takahashi, C. Y. Ho and S. Mori, "The scheduling of the parameters in Hopfield neural networks with fuzzy control", *IEICE Transactions on Information and Systems*, vol. E77-D, no. 8, pp. 895-903, 1994.

[102] A. I. Vakhutinsky and B. L. Golden, "A hierarchical strategy for solving travelling salesman problems using elastic nets", *Journal of Heuristics*, vol. 2, pp.67-76, 1995.

[103] D. E. Van den Bout and T. K. Miller III, "A travelling salesman objective function that works", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 299-303, 1988.

[104] D. E. Van den Bout and T. K. Miller III, "Improving the performance of the Hopfield-Tank neural network through normalization and annealing", *Biological Cybernetics*, vol. 62, pp. 129-139, 1989.

[105] D. E. Van den Bout and T. K. Miller III, "Graph partitioning using annealed neural networks", *IEEE Transactions on Neural Networks*, vol. 1, pp. 192-203, 1990.

[106] M. Verleysen and P. Jespers, "An analog VLSI implementation of Hopfield's neural network", *IEEE Micro*, pp. 46-55, December, 1989.

[107] L. Wang and K. A. Smith, "On chaotic simulated annealing", *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 716-718, 1998.

[108] L. Wang and F. Tian, "Noisy chaotic neural networks for solving combinatorial optimization problems", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 4, pp. 31-36, 2000.

[109] P. J. Werbos, "Extending chaos and complexity theory to address life, brain and quantum foundations", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 4, pp. 37-40, 2000.

[110] K. Wiesenfeld and F. Jaramillo, "Minireview of stochastic resonance", *Chaos*, vol. 8, no. 3, pp. 539-548, 1998.

[111] G. V. Wilson and G. S. Pawley, "On the stability of the TSP algorithm of Hopfield and Tank", *Biological Cybernetics*, vol. 58, pp. 63-70, 1988.

[112] N. Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood-Cliffs, NJ, 1976.

[113] A. M. Yaglom and I. M. Yaglom,. *Challenging Mathematical Problems with Elementary Solutions*, Holden-Day, San Francisco, CA, 1964.

[114] T. Yamada, K. Aihara and M. Kotani, "Chaotic neural networks and the travelling salesman problem", *Proceedings of the International Joint Conference on Neural Networks*, vol.2, pp. 1549-1552, 1993.

List of Publications

Refereed Journals

1. T. Kwok and K. A. Smith, “A Unified Framework for Chaotic Neural Network Approaches to Combinatorial Optimization”, *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 978-981, 1999.
2. T. Kwok and K. A. Smith, “Experimental Analysis of Chaotic Neural Network Models for Combinatorial Optimization under a Unifying Framework”, *Neural Networks*, vol. 13, no. 7, pp. 731-744, 2000.
3. T. Kwok and K. A. Smith, “A Noisy Self-Organizing Neural Network with Bifurcation Dynamics for Combinatorial Optimization”, submitted to *IEEE Transactions on Neural Networks*, 2001.

Refereed Book Chapters

1. T. Kwok, K. A. Smith and L. Wang, “Solving Combinatorial Optimization Problems by Chaotic Neural Networks”, in *Intelligent Engineering Systems through Artificial Neural Networks: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Rough Sets*, C. Dagli *et al* (eds.), vol. 8, pp. 317-322, ASME Press, NJ, 1998.

2. T. Kwok and K. A. Smith, "A Performance Comparison of Chaotic Simulated Annealing Models for Solving the *N*-queen Problem", in *Soft Computing in Industrial Applications*, Y. Suzuki *et al* (eds.), pp. 447-458, Springer-Verlag, London, 1999.
3. T, Kwok and K. A. Smith, "A Self-Organizing Neural Network with Attractor Nodes for Combinatorial Optimization", in *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems*, C. Dagli *et al* (eds.), vol. 10, pp. 209-216, ASME Press, NJ, 2000.

Refereed Conference Proceedings

1. T. Kwok, K. A. Smith and L. Wang, "Incorporating Chaos into the Hopfield Neural Network for Combinatorial Optimisation", *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Florida, vol. 1, pp. 659-665, 1998.
2. T. Kwok and K. A. Smith, "Improving the Optimisation Performance of a Self-Organising Neural Network with Weight Normalisation", *Proceedings of the ICSC Symposia on Intelligent Systems and Application (ISA2000)*, Wollongong, paper no. 1513-285, 2000.
3. T. Kwok and K. A. Smith, "Nonlinear System Dynamics in the Normalisation Process of a Self-Organising Neural Network for Combinatorial Optimisation", to appear in the *Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks*, Granada, 2001.